6. Motion Control

Wankyun Chung, Li-Chen Fu, Su-Hau Hsu[†]

This Chapter will focus on the motion control of robotic rigid manipulators. In other words, this Chapter does not treat the motion control of mobile robots, flexible manipulators, and manipulators with elastic joints. The main challenge in the motion control problem of rigid manipulators is the complexity of their dynamics and uncertainties. The former results from nonlinearity and coupling in the robot manipulators. The latter is twofold: structured and unstructured. Structured uncertainty means imprecise knowledge of the dynamic parameters and will be touched upon in this Chapter, whereas unstructured uncertainty results from joint and link flexibility, actuator dynamics, friction, sensor noise, and unknown environment dynamics, and will be treated in other Chapters.

In this Chapter, we begin with an introduction to motion control of robot manipulators from a fundamental viewpoint, followed by a survey and brief review of the relevant advanced materials. Specifically, the dynamic model and useful properties of robot manipulators are recalled in Sect. 6.1. The joint and operational space control approaches, two different viewpoints on control of robot manipulators, are compared in Sect. 6.2. Independent joint control and proportional–integral–derivative (PID) control, widely adopted in the field of industrial robots, are presented in Sections 6.3 and 6.4, respectively. Tracking control, based on feedback linearization, is introduced in Sect. 6.5. The computed-torque control and its variants are described in Sect. 6.6. Adaptive control is introduced in Sect. 6.7 to solve the problem of structural uncertainty, whereas the optimality and robustness issues are covered in Sect. 6.8. Since most controllers of robot manipulators are implemented by using microprocessors, the issues of digital implementation are discussed in Sect. 6.9. Finally, learning control, one popular approach to intelligent control, is illustrated in Sect. 6.10.

6.1 Introduction to Motion Control

The dynamical model of robot manipulators will be recalled in this section. Furthermore, important properties of this dynamical model, which is useful in controller design, will then be addressed. Finally, different control tasks of the robot manipulators will be defined.

6.1.1 Dynamical Model

For motion control, the dynamical model of rigid robot manipulators is conveniently described by Lagrange dynamics. Let the robot manipulator have n links and let the $(n \times 1)$-vector q of joint variables be $q = [q_1, \ldots, q_n]^\top$. The dynamic model of the robot manipulator is then described by Lagrange's equation [6.1–6]:

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + \tau_g(q) = \tau ,\qquad (6.1)$$

where $H(q)$ is the $(n \times n)$ inertia matrix, $C(q, \dot{q})\dot{q}$ is the $(n \times 1)$-vector of Coriolis and centrifugal forces, $\tau_g(q)$ is the $(n \times 1)$-vector of gravity force, and τ is the $(n \times 1)$-vector of joint control inputs to be designed. Friction and disturbance input have been neglected here.

Remark.

- Other contributions to the dynamic description of the robot manipulators may include the dynamics of the actuators, joint and link flexibility, friction, noise, and disturbances. Without loss of generality, the case of the rigid robot manipulators is stressed here.

The control schemes that we will introduce in this Chapter are based on some important properties of the dynamical model of robot manipulators. Before giving a detailed introduction to these different schemes, let us first give a list of those properties.

Property 6.1
The inertia matrix is a symmetric positive-definite matrix, which can be expressed

$$\lambda_h I_n \leq H(q) \leq \lambda_H I_n ,\qquad (6.2)$$

where λ_h and λ_H denote positive constants.

Property 6.2
The matrix $N(q, \dot{q}) = \dot{H}(q) - 2C(q, \dot{q})$ is skew-symmetric for a particular choice of $C(q, \dot{q})$ (which is always possible), i. e.,

$$z^\top N(q, \dot{q})z = 0 \qquad (6.3)$$

for any $(n \times 1)$-vector z.

Property 6.3
The $(n \times n)$-matrix $C(q, \dot{q})$ satisfies

$$\| C(q, \dot{q}) \| \leq c_o \| \dot{q} \| \qquad (6.4)$$

for some bounded constant c_o.

Property 6.4
The gravity force/torque vector satisfies

$$\| \tau_g(q) \| \leq g_o \qquad (6.5)$$

for some bounded constant g_o.

Property 6.5
The equation of motion is linear in the inertia parameters. In other words, there is a $(r \times 1)$ constant vector a and an $(n \times r)$ regressor matrix $Y(q, \dot{q}, \ddot{q})$ such that

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + \tau_g(q) = Y(q, \dot{q}, \ddot{q})a .\qquad (6.6)$$

The vector $\boldsymbol{a}$ is comprised of link masses, moments of inertia, and the link, in various combinations.

Property 6.6

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

$$\int_0^t \dot{\boldsymbol{q}}^\top(\beta)\boldsymbol{\tau}(\beta)\,\mathrm{d}\beta \geq -\alpha\,, \quad \forall t < \infty\,. \tag{6.7}$$

Remarks.

- Properties 6.3 and 6.4 are very useful since they allow us to establish upper bounds on the nonlinear terms in the dynamical model. As we will see further, several control schemes require knowledge of such upper bounds.
- In Property 6.5, the parameter vector $\boldsymbol{a}$ is comprised of several variables in various combinations. The dimensionality of the parameter space is not unique, and the search over the parameter space is an important problem.
- In this section, we assume that *the robot manipulator is fully actuated* and this indicates that there is an independent control input for each degree of freedom (DOF). In contrast, the robot manipulators with joint or link flexibility are no longer fully actuated and the control problems are more difficult in general.

6.1.2 Control Tasks

It is instructive for comparative purposes to classify control objectives into the following two classes:

- *Trajectory tracking* is aimed at following a time-varying joint reference trajectory specified within the manipulator workspace. In general, this desired trajectory is assumed to comply with the actuators' capacity. In other words, the joint velocity and acceleration associated with the desired trajectory should

not violate, respectively, the velocity and acceleration limit of the manipulator. In practice, the capacity of actuators is set by torque limits, which result in bounds on the acceleration that are complex and state dependent.

- *Regulation* sometimes is also called point-to-point control. A fixed configuration in the joint space is specified; the objective is to bring to and keep the joint variable at the desired position in spite of torque disturbances and independently of the initial conditions. The behavior of transients and overshooting, are in general, not guaranteed.

The selection of the controller may depend on the type of task to be performed. For example, tasks only requiring the manipulator to move from one position to another without requiring significant precision during the motion between these two points can be solved by regulators, whereas such as welding, painting, and so on, require tracking controllers.

Remarks.

- The regulation problem may be seen as a special case of the tracking problem (for which the desired joint velocity and acceleration are zero).
- The task specification above is given in the joint space and results in joint space control, which is the main content of this Chapter. Sometimes, the task specification of the robot manipulators in terms of the desired trajectory of the end-effector (e.g., control with eye-in-hand) is carried out in the task space and gives rise to the operational space control, which will be introduced in Sect. 6.2.

6.1.3 Summary

In this section, we introduced the dynamical model of the robot manipulators and important properties of this dynamical model. Finally, we defined different control tasks of the robot manipulators.

6.2 Joint Space Versus Operational Space Control

In a motion control problem, the manipulator moves to a position to pick up an object, transports that object to another location, and deposits it. Such a task is an integral part of any higher-level manipulation tasks such as painting or spot-welding.

Tasks are usually specified in the task space in terms of a desired trajectory of the end-effector, while control actions are performed in the joint space to achieve the desired goals. This fact naturally leads to two kinds of general control methods, namely *joint space con-*

trol and *operational space control (task space control)* schemes.

6.2.1 Joint Space Control

The main goal of the joint space control is to design a feedback controller such that the joint coordinates $q(t) \in R^n$ track the desired motion $q_d(t)$ as closely as possible. To this end, consider the equations of motion (6.1) of an n-DOF manipulator expressed in the joint space [6.2, 4]. In this case, the control of robot manipulators is naturally achieved in the joint space, since the control inputs are the joint torques. Nevertheless, the user specifies a motion in terms of end-effector coordinates, and thus it is necessary to understand the following strategy.

Figure 6.1 shows the basic outline of the joint space control methods. Firstly, the desired motion, which is described in terms of end-effector coordinates, is converted to a corresponding joint trajectory using the inverse kinematics of the manipulator. Then the feedback controller determines the joint torque necessary to move the manipulator along the desired trajectory specified in joint coordinates starting from measurements of the current joint states [6.1, 4, 7, 8].

Since it is always assumed that the desired task is given in terms of the time sequence of the joint motion, joint space control schemes are quite adequate in situations where manipulator tasks can be accurately preplanned and little or no online trajectory adjustments are necessary [6.1, 4, 7, 9]. Typically, inverse kinematics is performed for some intermediate task points, and the joint trajectory is interpolated using the intermediate joint solutions. Although the command trajectory consists of straight-line motions in end-effector coordinates between interpolation points, the resulting joint motion consists of curvilinear segments that match the desired end-effector trajectory at the interpolation points.

In fact, the joint space control includes simple PD control, PID control, inverse dynamic control, Lyapunov-based control, and passivity-based control, as explained in the following sections.

6.2.2 Operational Space Control

In more complicated and less certain environments, end-effector motion may be subject to online modifications in order to accommodate unexpected events or to respond to sensor inputs. There are a variety of tasks in manufacturing where these type of control problem arise. In particular, it is essential when controlling the interaction between the manipulator and environment is of concern.

Since the desired task is often specified in the operational space and requires precise control of the end-effector motion, joint space control schemes are not suitable in these situations. This motivated a different approach, which can develop control schemes directly based on the dynamics expressed in the operational space [6.10, 11].

Let us suppose that the Jacobian matrix, denoted by $J(q) \in R^{n \times n}$, transforms the joint velocity ($\dot{q} \in R^n$) to the task velocity ($\dot{x} \in R^n$) according to

$$\dot{x} = J(q)\dot{q} . \tag{6.8}$$

Furthermore, assume that it is invertible. Then, the operational space dynamics is expressed as follows:

$$f_c = \Lambda(q)\ddot{x} + \Gamma(q, \dot{q})\dot{x} + \eta(q) , \tag{6.9}$$

where $f_c \in R^n$ denotes the command forces in the operational space; the *pseudo-inertia matrix* is defined by

$$\Lambda(q) = J^{-\top}(q)H(q)J^{-1}(q) , \tag{6.10}$$

and $\Gamma(q, \dot{q})$ and $\eta(q)$ are given by

$$\Gamma(q, \dot{q}) = J^{-\top}(q)C(q, \dot{q})J^{-1}(q) - \Lambda(q)J(q)J^{-1}(q) ,$$
$$\eta(q) = J^{-\top}(q)\tau_g(q) .$$

The task space variables are usually reconstructed from the joint space variables, via the kinematic mappings. In fact, it is quite rare to have sensors to directly measure end-effector positions and velocities. Also, it is worth remarking that an analytical Jacobian is utilized since the control schemes operate directly on task space quantities, i. e., the end-effector position and orientation.

The main goal of the operational space control is to design a feedback controller that allows execution of an end-effector motion $x(t) \in R^n$ that tracks the de-

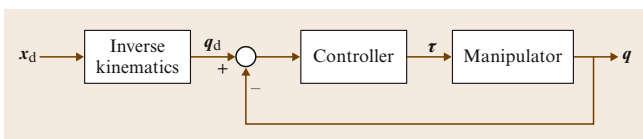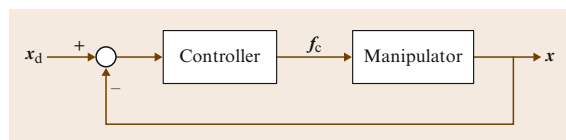**Fig. 6.1** Generic concept of joint space control

Fig. 6.2 Basic concept of operational space control

sired end-effector motion $x_\mathrm{d}(t)$ as closely as possible. To this end, consider the equations of motion (6.9) of the manipulator expressed in the operational space. For this case, Fig. 6.2 shows a schematic diagram of the operational space control methods. There are several advantages to such an approach because operational space controllers employ a feedback loop that directly minimizes task errors. Inverse kinematics need not be calculated explicitly, since the control algorithm embeds the velocity-level forward kinematics (6.8), as shown in the figure. Now, motion between points can be a straight-line segment in the task space.

6.3 Independent–Joint Control

By independent-joint control (i. e., decentralized control), we mean that the control inputs of each joint only depends on the measurement of the corresponding joint displacement and velocity. Due to its simple structure, this kind of control schemes offers many advantages. For example, by using independent-joint control, communication among different joints is saved. Moreover, since the computational load of controllers may be reduced, only low-cost hardware is required in actual implementations. Finally, independent-joint control has the feature of scalability, since the controllers on all the joints have the same formulation. In this section, two kinds of design of independent-joint control will be introduced: one focused on the dynamical model of each joint (i. e., based on the single-joint model) and the other based on the analysis of the overall dynamical model (i. e., the multijoint model) of robot manipulators.

6.3.1 Controller Design Based on the Single–Joint Model

The simplest independent-joint control strategy is to control each joint axis as a single-input single-output (SISO) system. Coupling effects among joints due to varying configuration during motion are treated as disturbance inputs. Without loss of generality, the actuator is taken as a rotary electric direct-current (DC) motor. Hence, the block diagram of the control scheme of joint i can be represented in the domain of the complex variables as shown in Fig. 6.3. In this scheme, θ is the angular variable of the motor, J is the effective inertia viewed from the motor side, R_a is the armature resistance (auto-inductance being neglected), and k_t and k_v are, respectively, the torque and motor constants. Furthermore, G_v denotes the voltage gain of the power amplifier so that the reference input is the input voltage V_c of the amplifier instead of the armature voltage V_a. It has also been assumed that $F_\mathrm{m} \ll k_\mathrm{v}k_\mathrm{t}/R_\mathrm{a}$, i. e., the mechanical (viscous) friction coefficient has been neglected with respect to the electrical coefficient. Now the input–output transfer function of the motor can be written as

$$M(s) = \frac{k_\mathrm{m}}{s(1+sT_\mathrm{m})}, \tag{6.11}$$

where $k_\mathrm{m} = G_\mathrm{v}/k_\mathrm{v}$ and $T_\mathrm{m} = R_\mathrm{a}J/k_\mathrm{v}k_\mathrm{t}$ are, respectively, the voltage-to-velocity gain and the time constant of the motor.

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

1. a large value of the amplifier before the point of intervention of the disturbance;
2. the presence of an integral action in the controller so as to cancel the effect of the gravitational component on the output at the steady state (i. e., constant θ).

In this case, as shown in Fig. 6.4, the types of control action with position and velocity feedback are characterized by [6.4]

$$G_\mathrm{p}(s) = K_\mathrm{P}, \quad G_\mathrm{v}(s) = K_\mathrm{V}\frac{1+sT_\mathrm{V}}{s}, \tag{6.12}$$

where $G_\mathrm{p}(s)$ and $G_\mathrm{v}(s)$ correspond to the position and velocity control actions, respectively. It is worth noting that the inner control action $G_\mathrm{v}(s)$ is in a form of propositional–integral (PI) control to yield zero error in the steady state under a constant disturbance d. Furthermore, k_TP and k_TV are both transducer constants, and the amplifier gain K_V has been embedded in the gain
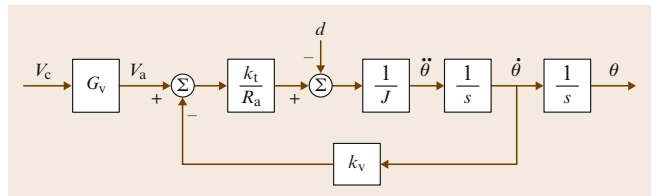

Fig. 6.3 Block scheme of a joint-driven system (after [6.4])
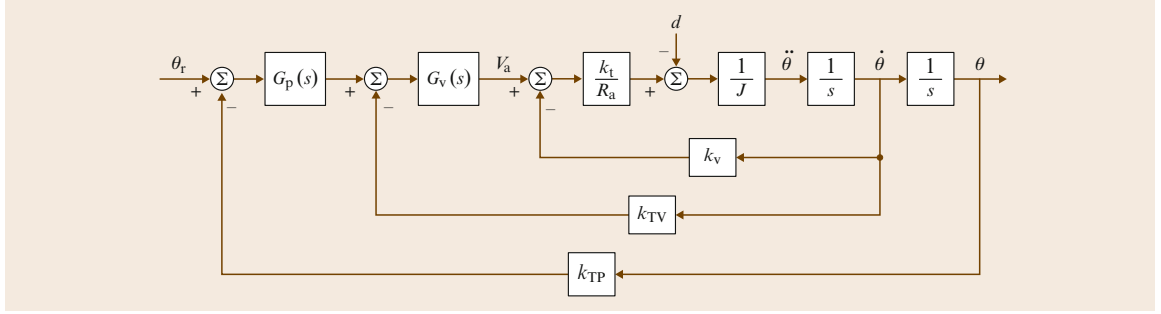
Fig. 6.4 Block scheme of position and velocity feedback (after [6.4])

of the inner controller. From the scheme of Fig. 6.4, the transfer function of the forward path is

$$P(s) = \frac{k_m K_P K_V (1 + s T_V)}{s(1 + s T_m)} \,, \tag{6.13}$$

while that of the return path is

$$H(s) = k_{TP} \left(1 + s \frac{k_{TV}}{K_P k_{TP}} \right) \,. \tag{6.14}$$

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, $k_m K_V k_{TV}$. By increasing the position feedback gain K_P, it is possible to confine the closed-loop poles to a region of the complex plane with large absolute real part. Then, the actual location can be established by a suitable choice of K_V.

The closed-loop input–output transfer function is

$$\frac{\Theta(s)}{\Theta_r(s)} = \frac{\frac{1}{k_{TP}}}{1 + \frac{s k_{TP}}{K_P k_{TP}} + \frac{s^2}{k_m K_P k_{TP} K_V}} \,, \tag{6.15}$$

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

$$W(s) = \frac{\frac{1}{k_{TP}}}{1 + \frac{2\zeta s}{\omega_n} + \frac{s^2}{\omega_n^2}} \,. \tag{6.16}$$

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 specifications, the following relations can be found

$$K_V k_{TV} = \frac{2\zeta \omega_n}{k_m} \quad \text{and} \quad K_P k_{TP} K_V = \frac{\omega_n^2}{k_m} \,. \tag{6.17}$$

For given transducer constants k_{TP} and k_{TV}, K_V and K_P will be chosen to satisfy the two equations above, respectively. On the other hand, the closed-loop disturbance/output function is

$$\frac{\Theta(s)}{D(s)} = \frac{\frac{s R_a}{k_t K_P K_{TP} K_V (1 + s T_m)}}{1 + \frac{s k_{TV}}{K_P k_{TP}} + \frac{s^2}{k_m K_P k_{TP} K_V}} \,, \tag{6.18}$$

which shows that the disturbance rejection factor is $X_R(s) = K_P k_{TP} K_V$ and is fixed, provided that K_V and K_P have been chosen via the approach above. Concerning the disturbance dynamics, the zero at the origin introduced by the PI, a real pole at $s = -1/T_m$, and the pair of complex poles with real part $-\zeta \omega_n$ should be kept in mind. In this case, an estimate T_R of the output recovery time needed by the control system to recover from the effect of a disturbance on the joint position can be evaluated by analyzing models of the transfer function above. Such an estimate can reasonably be expressed as $T_R = \max\{T_m, 1/\zeta \omega\}$.

6.3.2 Controller Design Based on the Multijoint Model

In recent years, independent-joint control schemes based on the complete dynamic model of the robot manipulators (i.e., a multijoint model) have been proposed. For example, following the approach of computed-torque-like control, [6.12] dealt with the regulation task for horizontal motions, and [6.13] and [6.14] handled the tracking task for arbitrary smooth trajectories. Since the overall dynamic model is considered, the effects of coupling among joints are handled. These schemes will be introduced in detail in Sect. 6.6.

6.3.3 Summary

In this section, we have presented two independent-joint control schemes: one is based on the single-joint model and the other is based on the multijoint model. The former focuses on the dynamics of a single joint and regards the interaction among joints as a disturbance. This control scheme is simple but may not be suitable for high-speed tracking. Hence, we introduce the latter, which considers the overall dynamical model of robot manipulators such that the interaction among joints can be handled.

6.4 PID Control

Traditionally, control design in robot manipulators can be understood as the simple fact of tuning of a PD or PID compensator at the level of each motor driving the manipulator joints [6.1]. Fundamentally, a PD controller is a position and velocity feedback that has good closed-loop properties when applied to a double integrator system.

The PID control has a long history since Ziegler and Nichols' PID tuning rules were published in 1942 [6.15]. Actually, the strong point of PID control lies in its *simplicity* and clear physical meaning. Simple control is preferable to complex control, at least in industry, if the performance enhancement obtained by using complex control is not significant enough. The physical meanings of PID control [6.16] are as follows:

P-control means the present effort making a present state into desired state;

I-control means the accumulated effort using the experience information of previous states;

D-control means the predictive effort reflecting the information about trends in future states.

6.4.1 PD Control for Regulation

A simple design method for manipulator control is to utilize a linear control scheme based on the linearization of the system about an operating point. An example of this method is a PD control with a gravity compensation scheme [6.17, 18]. Gravity compensation acts as a bias correction, compensating only for the amount of forces that create overshooting and an asymmetric transient behavior. Formally, it has the following form

$$\boldsymbol{\tau} = \boldsymbol{K}_\mathrm{P}(\boldsymbol{q}_\mathrm{d} - \boldsymbol{q}) - \boldsymbol{K}_\mathrm{V}\dot{\boldsymbol{q}} + \boldsymbol{\tau}_\mathrm{g}(\boldsymbol{q}) , \tag{6.19}$$

Further Reading

There are different types of feedback applied in the independent-joint control based on the single-joint model (such as pure position feedback or position, velocity, and acceleration feedback). A complete discussion is given in [6.4]. When the joint control servos are required to track reference trajectories with high speeds and accelerations, the tracking capabilities of the above schemes are inevitably degraded. A possible remedy is to adopt decentralized feedforward compensation to reduce tracking errors [6.4, 5].

where $\boldsymbol{K}_\mathrm{P}$ and $\boldsymbol{K}_\mathrm{V} \in R^{n \times n}$ are positive-definite gain matrices. This controller is very useful for set-point regulation, i. e., $\boldsymbol{q}_\mathrm{d} = \text{constant}$ [6.7, 18]. When this controller is applied to (6.1), the closed-loop equation becomes

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{K}_\mathrm{V}\dot{\boldsymbol{q}} - \boldsymbol{K}_\mathrm{P}\boldsymbol{e}_\mathrm{q} = \boldsymbol{0} , \tag{6.20}$$

where $\boldsymbol{e}_\mathrm{q} = \boldsymbol{q}_\mathrm{d} - \boldsymbol{q}$, and the equilibrium point is $\boldsymbol{y} = [\boldsymbol{e}_\mathrm{q}^\top, \dot{\boldsymbol{q}}^\top]^\top = \boldsymbol{0}$. Now, the stability achieved by PD control with gravity compensation can be analyzed according to the closed-loop dynamic (6.20). Consider the positive-definite function

$$V = \frac{1}{2}\dot{\boldsymbol{q}}^\top \boldsymbol{H}(\boldsymbol{q})\dot{\boldsymbol{q}} + \frac{1}{2}\boldsymbol{e}_\mathrm{q}^\top \boldsymbol{K}_\mathrm{V}\boldsymbol{e}_\mathrm{q} .$$

Then, the derivative of function becomes negative semidefinite for any value of $\dot{\boldsymbol{q}}$ by using Property 6.2 in Sect. 6.1, i. e.,

$$\dot{V} = -\dot{\boldsymbol{q}}^\top \boldsymbol{K}_\mathrm{V}\dot{\boldsymbol{q}} \leq -\lambda_{\min}(\boldsymbol{K}_\mathrm{V})\|\dot{\boldsymbol{q}}\|^2 , \tag{6.21}$$

where $\lambda_{\min}(\boldsymbol{K}_\mathrm{V})$ means the smallest eigenvalue of $\boldsymbol{K}_\mathrm{V}$. By invoking the Lyapunov stability theory and LaSalle's theorem [6.1], it can be shown that the regulation error will converge asymptotically to zero, while their high-order derivatives remain bounded. This controller requires knowledge of the gravity components (structure and parameters), though it is simple.

Now, consider simple PD control without gravity compensation

$$\boldsymbol{\tau} = \boldsymbol{K}_\mathrm{P}(\boldsymbol{q}_\mathrm{d} - \boldsymbol{q}) - \boldsymbol{K}_\mathrm{V}\dot{\boldsymbol{q}} , \tag{6.22}$$

then the closed-loop dynamic equation becomes

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{\tau}_\mathrm{g}(\boldsymbol{q}) + \boldsymbol{K}_\mathrm{V}\dot{\boldsymbol{q}} - \boldsymbol{K}_\mathrm{P}\boldsymbol{e}_\mathrm{q} = \boldsymbol{0} . \tag{6.23}$$

Consider the positive definite function

$$V = \frac{1}{2}\dot{q}^\top H(q)\dot{q} + \frac{1}{2}e_q^\top K_V e_q + U(q) + U_0 \,,$$

where $U(q)$ denotes the potential energy with the relation of $\partial U(q)/\partial q = \tau_g(q)$ and U_0 is a suitable constant. Taking time derivative of V along the closed-loop dynamics (6.23) gives the same result (6.21) with previous one using gravity compensation. In this case, the control system must be stable in the sense of Lyapunov, but it can not conclude that the regulation error will converge to zero by LaSalle's theorem [6.1]. Actually, the system precision (the size of th eregulation error vector) will depend on the size of the gain matrix K_P in the following form:

$$\|e_q\| \le \|K_P^{-1}\| g_0 \,, \tag{6.24}$$

where g_0 is that in Property 6.4 in Sect. 6.1. Hence, the regulation error can be arbitrarily reduced by increasing K_P; nevertheless, measurement noise and other unmodeled dynamics, such as actuator friction, will limit the use of high gains in practice.

6.4.2 PID Control for Regulation

An integral action may be added to the previous PD control in order to deal with gravity forces, which to some extent can be considered as a constant disturbance (from the local point of view). The PID regulation controller can be written in the following general form:

$$\tau = K_P(q_d - q) + K_I \int f(q_d - q)\,dt - K_V\dot{q} \,,$$

where $K_I \in R^{n\times n}$ is a positive-definite gain matrix, and

if $f(q_d - q) = q_d - q$, we have PID control,
if $K_I \int(-\dot{q})\,dt$ is added, we have PI^{2}D control,
if $f(\cdot) = \tanh(\cdot)$, we have PD + nonlinear integral control.

Global asymptotic stability (GAS) by PID control was proved in [6.12] for robotic motion control system including external disturbances, such as Coulomb friction. Also, *Tomei* proved the GAS of PD control in [6.19] by using an adaptation for gravity term. On the other hand, *Ortega* et al. showed in [6.20] that PI^{2}D control could yield semiglobal asymptotic stability (SGAS) in the presence of gravity and bounded external disturbances. Also, *Angeli* proved in [6.21] that PD control could achieve the input-output-to-state stability (IOSS) for robotic systems. Also, *Ramirez* et al. proved the SGAS (with some conditions) for PID gains in [6.22].

Also, *Kelly* proved in [6.23] that PD plus nonlinear integral control could achieve GAS under gravity.

Actually, a large integral action in PID control can cause instability of the motion control system. In order to avoid this, the integral gain should be bounded by an upper limit of the following form [6.1]:

$$\frac{k_P k_V}{\lambda_H^2} > k_I \,,$$

where λ_H is that in Property 6.1 in Sect. 6.1, $K_P = k_P I$, $K_I = k_I I$, and $K_V = k_V I$. This relation gives the guidelines for gain selection implicitly. Also, PID control has generated a great variety of PID control plus something, e.g., PID plus friction compensator, PID plus gravity compensator, PID plus disturbance observer.

6.4.3 PID Gain Tuning

The PID control can be utilized for trajectory tracking as well as set-point regulation. True tracking control will be treated after Sect. 6.5. In this section, the simple but useful PID gain tuning method will be introduced for practical use. The general PID controller can be written in the following general form:

$$\tau = K_V\dot{e}_q + K_P e_q + K_I \int e_q\,dt \,,$$

or, in another form,

$$\tau = \left(K + \frac{1}{\gamma^2}I\right)\left(\dot{e}_q + K_P e_q + K_I \int e_q\,dt\right) \,. \tag{6.25}$$

In a fundamental stability analysis of tracking control systems, *Qu* et al. proved in [6.24] that PD control could satisft uniform ultimate boundedness (UUB). Also, *Berghuis* et al. proposed output feedback PD control, which satisfies semiglobal uniform ultimate boundedness (SGUUB) in [6.25] under gravity and a bounded disturbance. Recently, *Choi* et al. suggested inverse optimal PID control [6.26], assuring extended disturbance input-to-state stability (ISS).

Actually, if a PID controller (6.25) is repeatedly applied to the same set point or desired trajectory, then the maximum error will be proportional to the gains in the following form:

$$\max_{0 \le t \le t_f} \|e_q(t)\| \propto \frac{\gamma^2}{\sqrt{2k\gamma^2 + 1}} \,, \tag{6.26}$$

where t_f denotes the final execution time of a given task and $K = kI$. This relation can be utilized to tune the gain of a PID controller and is referred to as the compound

tuning rule [6.16]. The compound tuning rule implicitly includes simple tuning rules as follows:

Square tuning: $\max \|e_q\| \propto \gamma^2$, for a small k,

Linear tuning: $\max \|e_q\| \propto \gamma$, for a large k.

For example, suppose we select positive constant diagonal matrices $K_P = k_P I$, $K_I = k_I I$, while satisfying $k_P^2 > 2k_I$. For small k, the maximum error will be reduced by $\frac{1}{4}$ according to the square tuning rule, if we reduce the value γ by $\frac{1}{2}$. For large k, the maximum error will be proportionally reduced as γ according to the linear tuning rule. This means that we can tune the PID controller using only one parameter γ when the other gain parameters are fixed [6.16]. Although these rules are very useful in tuning the control performance, they can be utilized only for repetitive experiments for the same set point or desired trajectory because the tuning rules consist of proportional relations.

Further Reading

PID-type controllers were designed to solve the regulation control problem. They have the advantage of requiring knowledge of neither the model structure nor the model parameters. Also, the stability achieved by PID-type controllers was presented in this section. A range of books and papers [6.1, 15, 16, 22, 27, 28] are available to the robotics audience, detailing the individual tuning methods used in PID control and their concrete proofs.

6.5 Tracking Control

While independent PID controls are adequate in most set-point regulation problems, there are many tasks that require effective trajectory tracking capabilities such as plasma-welding, laser-cutting or high-speed operations in the presence of obstacles. In this case, employing local schemes requires moving slowly through a number of intermediate set points, thus considerably delaying the completion of the task. Therefore, to improve the trajectory tracking performance, controllers should take account of the manipulator dynamic model via a computed-torque-like technique.

The tracking control problem in the joint or task space consists of following a given time-varying trajectory $q_d(t)$ or $x_d(t)$ and its successive derivatives $\dot{q}_d(t)$ or $\dot{x}_d(t)$ and $\ddot{q}_d(t)$ or $\ddot{x}_d(t)$, which describe the desired velocity and acceleration, respectively. To obtain successful performance, significant effort has been devoted to the development of model-based control strategies [6.1,2,7]. Among the control approaches reported in the literature, typical methods include inverse dynamic control, the feedback linearization technique, and the passivity-based control method.

6.5.1 Inverse Dynamics Control

Though the inverse dynamics control has a theoretical background, such as the theory of feedback linearization discussed later, its starting point is mechanical engineering intuition based on cancelling nonlinear terms and decoupling the dynamics of each link. Inverse dynamics control in joint space has the form

$$\tau = H(q)v + C(q, \dot{q})\dot{q} + \tau_g(q) \tag{6.27}$$

which, applied to (6.1), yields a set of n decoupled linear systems, e.g., $\ddot{q} = v$, where v is an auxiliary control input to be designed. Typical choices for v are

$$v = \ddot{q}_d + K_V(\dot{q}_d - \dot{q}) + K_P(q_d - q) \tag{6.28}$$

or with an integral component

$$v = \ddot{q}_d + K_V(\dot{q}_d - \dot{q}) + K_P(q_d - q) \\ + K_I \int (q_d - q)\, dt, \tag{6.29}$$

leading to the error dynamics equation

$$\ddot{e}_q + K_V \dot{e}_q + K_P e_q = 0$$

for an auxiliary control input (6.28), and

$$e_q^{(3)} + K_V \ddot{e}_q + K_P \dot{e}_q + K_I e_q = 0$$

if an auxiliary control input (6.29) is used. Both error dynamics are exponentially stable by a suitable choice of the gain matrices K_V and K_P (and K_I).

Alternatively, inverse dynamics control can be described in the operational space. Consider the operational space dynamics (6.9). If the following inverse dynamics control is used in the operational space,

$$f_c = \Lambda(q)(\ddot{x}_d + K_V \dot{e}_x + K_P e_x) + \Gamma(q, \dot{q})\dot{x} + \eta(q),$$

where $e_x = x_d - x$, the resulting error dynamics is

$$\ddot{e}_x + K_V \dot{e}_x + K_P e_x = 0, \tag{6.30}$$

and it is also exponentially stable. One apparent advantage of using this controller is that K_P and K_V can be selected with a clear physical meaning in operational space. However, as can be seen in (6.10), $\Lambda(q)$ becomes very large when the robot approaches singular configurations [6.8]. This means that large forces in some direction are needed to move the arm.

6.5.2 Feedback Linearization

This approach generalizes the concept of inverse dynamics of rigid manipulators. The basic idea of feedback linearization is to construct a transformation as a so-called *inner-loop control*, which exactly linearizes the nonlinear system after a suitable state space change of coordinates. One can then design a second stage or *outer-loop control* in the new coordinates to satisfy the traditional control design specifications such as tracking, disturbance rejection, etc. [6.5,29]. The full power of the feedback linearization scheme for manipulator control becomes apparent if one includes in the dynamic description of the manipulator the transmission dynamics, such as the elasticity resulting from shaft windup, gear elasticity, etc. [6.5].

In recent years, an impressive volume of literature has emerged in the area of differential-geometric methods for nonlinear systems. Most of the results in this area are intended to give abstract coordinate-free descriptions of various geometric properties of nonlinear systems and as such are difficult for the non-mathematician to follow. It is our intention here to give only the basic idea of the feedback linearization scheme and to introduce a simple version of this technique that finds an immediate application to the manipulator control problem. The reader is referred to [6.30] for a comprehensive treatment of the feedback linearization technique using differential-geometric methods.

Let us now develop a simple approach to the determination of linear state-space representations of the manipulator dynamics (6.1) by considering a general sort of output $\xi \in R^p$:

$$\xi = h(q) + r(t) , \tag{6.31}$$

where $h(q)$ is a general predetermined function of the joint coordinate $q \in R^n$ and $r(t)$ is a general predetermined time function. The control objective will be to select the joint torque inputs τ in order to make the output $\xi(t)$ go to zero.

The choice of $h(q)$ and $r(t)$ is based on the control purpose. For example, if $h(q) = -q$ and $r(t) = q_d(t)$, the desired joint space trajectory we would like the manipulator to follow, then $\xi(t) = q_d(t) - q(t) \equiv e_q(t)$ is the joint space tracking error. Forcing $\xi(t)$ to zero in this case would cause the joint variables $q(t)$ to track their desired values $q_d(t)$, resulting in a manipulator trajectory-following problem. As another example, $\xi(t)$ could represent the operational space tracking error $\xi(t) = x_d(t) - x(t) \equiv e_x(t)$. Then, controlling $\xi(t)$ to zero would result in trajectory following directly in operational space where the desired motion is usually specified.

To determine a linear state-variable model for manipulator controller design, let us simply differentiate the output $\xi(t)$ twice to obtain

$$\dot{\xi} = \frac{\partial h}{\partial q} \dot{q} + \dot{r} = T\dot{q} + \dot{r} , \tag{6.32}$$

$$\ddot{\xi} = T\ddot{q} + \dot{T}\dot{q} + \ddot{r} , \tag{6.33}$$

where we defined a $(p \times n)$ transformation matrix of the form

$$T(q) = \frac{\partial h(q)}{\partial q} = \left(\frac{\partial h}{\partial q_1} \quad \frac{\partial h}{\partial q_2} \quad \cdots \quad \frac{\partial h}{\partial q_n} \right) . \tag{6.34}$$

Given the output $h(q)$, it is straightforward to compute the transformation $T(q)$ associated with $h(q)$. In the special case where $\dot{\xi}$ represents the operational space velocity error, then $T(q)$ denotes the Jacobian matrix $J(q)$.

According to (6.1),

$$\ddot{q} = H^{-1}(q)[\tau - n(q, \dot{q})] \tag{6.35}$$

with the nonlinear terms represented by

$$n(q, \dot{q}) = C(q, \dot{q})\dot{q} + \tau_g(q) . \tag{6.36}$$

Then (6.33) yields

$$\ddot{\xi} = \ddot{r} + \dot{T}\dot{q} + T(q)H^{-1}(q)[\tau - n(q, \dot{q})] . \tag{6.37}$$

Define the *control input* function

$$u = \ddot{r} + \dot{T}\dot{q} + T(q)H^{-1}(q)[\tau - n(q, \dot{q})] . \tag{6.38}$$

Now we may define a state $y(t) \in R^{2p}$ by $y = (\xi \; \dot{\xi})$ and write the manipulator dynamics as

$$\dot{y} = \begin{pmatrix} 0 & I_p \\ 0 & 0 \end{pmatrix} y + \begin{pmatrix} 0 \\ I_p \end{pmatrix} u . \tag{6.39}$$

This is a linear state-space system of the form

$$\dot{y} = Ay + Bu , \tag{6.40}$$

driven by the control input u. Due to the special form of A and B, this system is called the *Brunovsky canonical form* and it is always controllable from $u(t)$.

Since (6.38) is said to be a *linearizing transformation* for the manipulator dynamic equation, one may invert this transformation to obtain the joint torque

$$\boldsymbol{\tau} = \boldsymbol{H}(\boldsymbol{q})\boldsymbol{T}^{+}(\boldsymbol{q})(\boldsymbol{u} - \ddot{\boldsymbol{r}} - \dot{\boldsymbol{T}}\dot{\boldsymbol{q}}) + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) , \qquad (6.41)$$

where $\boldsymbol{T}^{+}$ denotes the *Moore–Penrose inverse* of the transformation matrix $\boldsymbol{T}(\boldsymbol{q})$.

In the special case $\boldsymbol{\xi} = \boldsymbol{e}_{q}(t)$, and if we select $\boldsymbol{u}(t)$ so that (6.39) is stable by the PD feedback $\boldsymbol{u} = -\boldsymbol{K}_{P}\boldsymbol{\xi} - \boldsymbol{K}_{V}\dot{\boldsymbol{\xi}}$, then $\boldsymbol{T} = -\boldsymbol{I}_{n}$ and the control input torque $\boldsymbol{\tau}(t)$ defined by (6.41) makes the manipulator move in such a way that $\boldsymbol{y}(t)$ goes to zero. In this case, the feedback linearizing control and the inverse dynamics control become the same.

6.5.3 Passivity-Based Control

This method explicitly uses the passivity properties of the Lagrangian system [6.31, 32]. In comparison with the inverse dynamics method, passivity-based controllers are expected to have better robust properties because they do not rely on the exact cancellation of the manipulator nonlinearities. The passivity-based control input is given by

$$\dot{\boldsymbol{q}}_{r} = \dot{\boldsymbol{q}}_{d} + \alpha\boldsymbol{e}_{q} , \qquad \alpha > 0 ,$$
$$\boldsymbol{\tau} = \boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}}_{r} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_{r} + \boldsymbol{\tau}_{g}(\boldsymbol{q}) + \boldsymbol{K}_{V}\dot{\boldsymbol{e}}_{q} + \boldsymbol{K}_{P}\boldsymbol{e}_{q} . \tag{6.42}$$

With (6.42), we obtain the following closed-loop system

$$\boldsymbol{H}(\boldsymbol{q})\dot{\boldsymbol{s}}_{q} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{s}_{q} + \boldsymbol{K}_{V}\dot{\boldsymbol{e}}_{q} + \boldsymbol{K}_{P}\boldsymbol{e}_{q} = \boldsymbol{0} , \qquad (6.43)$$

where $\boldsymbol{s}_{q} = \dot{\boldsymbol{e}}_{q} + \alpha\boldsymbol{e}_{q}$. Let us choose a Lyapunov function $V(\boldsymbol{y}, t)$ as follows:

$$V = \frac{1}{2}\boldsymbol{y}^{\top}\begin{pmatrix} \alpha\boldsymbol{K}_{V} + \boldsymbol{K}_{P} + \alpha^{2}\boldsymbol{H} & \alpha\boldsymbol{H} \\ \alpha\boldsymbol{H} & \boldsymbol{H} \end{pmatrix}\boldsymbol{y}$$
$$= \frac{1}{2}\boldsymbol{y}^{\top}\boldsymbol{P}\boldsymbol{y} . \tag{6.44}$$

Since the above equation is positive definite, it has a unique equilibrium at the origin, i.e., $\boldsymbol{y} = (\boldsymbol{e}_{q}^{\top}, \ \dot{\boldsymbol{e}}_{q}^{\top})^{\top}$

$= \boldsymbol{0}$. Moreover, V can be bounded by

$$\sigma_{m}\|\boldsymbol{y}\|^{2} \le \boldsymbol{y}^{\top}\boldsymbol{P}\boldsymbol{y} \le \sigma_{M}\|\boldsymbol{y}\|^{2} , \quad \sigma_{M} \ge \sigma_{m} > 0. \ (6.45)$$

The time derivative of V gives

$$\dot{V} = -\dot{\boldsymbol{e}}_{q}^{\top}\boldsymbol{K}_{V}\dot{\boldsymbol{e}}_{q} - \alpha\boldsymbol{e}_{q}\boldsymbol{K}_{P}\boldsymbol{e}_{q} = -\boldsymbol{y}^{\top}\boldsymbol{Q}\boldsymbol{y} < 0 , \ (6.46)$$

where $\boldsymbol{Q} = \text{diag}[\alpha\boldsymbol{K}_{P}, \ \boldsymbol{K}_{V}]$. Since $\boldsymbol{Q}$ is positive definite and quadratic in $\boldsymbol{y}$, it can be also bounded by

$$\kappa_{m}\|\boldsymbol{y}\|^{2} \le \boldsymbol{y}^{\top}\boldsymbol{Q}\boldsymbol{y} \le \kappa_{M}\|\boldsymbol{y}\|^{2} , \quad \kappa_{M} \ge \kappa_{m} > 0. \tag{6.47}$$

Then, from the bound of the Lyapunov function V, we get

$$\dot{V} \le -\kappa_{m}\|\boldsymbol{y}\|^{2} = -2\eta V , \quad \eta = \frac{\kappa_{m}}{\sigma_{M}} \tag{6.48}$$

which finally yields

$$V(t) \le V(0)\mathrm{e}^{-2\eta t} . \tag{6.49}$$

It has been shown that the value of α affects the tracking result dramatically [6.33]. The manipulator tends to vibrate for small values of α. Larger values of α allow better tracking performance and protect $\boldsymbol{s}_{q}$ from being spoiled by the velocity measurement noise when the position error is small. In [6.34], it was suggested that

$$\boldsymbol{K}_{P} = \alpha\boldsymbol{K}_{V} \tag{6.50}$$

be used for quadratic optimization.

6.5.4 Summary

In this section, we have reviewed some of the model-based motion control methods proposed to date. Under some control approaches, the closed-loop system theoretically, gives either asymptotic stability or globally exponential stability. However, such ideal performance cannot be obtained in practical implementation because factors such as sampling rate, measurement noise, disturbances, and unmodeled dynamics will limit the achievable gain and the performance of the control algorithms [6.33, 35, 36].

6.6 Computed-Torque Control

Through the years many kinds of robot control schemes have been proposed. Most of these can be considered as special cases of the class of computed-torque control (Fig. 6.5) which is the technique of applying feedback

linearization to nonlinear systems in general [6.37, 38]. In the section, computed-torque control will be first introduced, and its variant, so-called computed-torque-like control, will be introduced later.
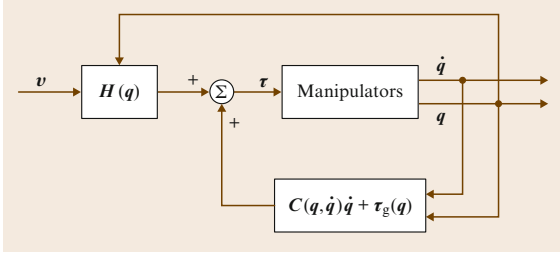
Fig. 6.5 Computed-torque control

6.6.1 Computed–Torque Control

Consider the control input (6.27)

$$\boldsymbol{\tau} = \boldsymbol{H}(\boldsymbol{q})\boldsymbol{v} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{\tau}_{\mathrm{g}}(\boldsymbol{q}) \,,$$

which is also known as computed-torque control; it consists of an inner nonlinear compensation loop and an outer loop with an exogenous control signal $\boldsymbol{v}$. Substituting this control law into the dynamical model of the robot manipulator, it follows that

$$\ddot{\boldsymbol{q}} = \boldsymbol{v} \,. \tag{6.51}$$

It is important to note that this control input converts a complicated nonlinear controller design problem into a simple design problem for a linear system consisting of n decoupled subsystems. One approach to the outer-loop control $\boldsymbol{v}$ is propositional–derivative (PD) feedback, as in (6.28):

$$\boldsymbol{v} = \ddot{\boldsymbol{q}}_{\mathrm{d}} + \boldsymbol{K}_{\mathrm{V}}\dot{\boldsymbol{e}}_{\mathrm{q}} + \boldsymbol{K}_{\mathrm{P}}\boldsymbol{e}_{\mathrm{q}} \,,$$

in which case the overall control input becomes

$$\boldsymbol{\tau} = \boldsymbol{H}(\boldsymbol{q})(\ddot{\boldsymbol{q}}_{\mathrm{d}} + \boldsymbol{K}_{\mathrm{V}}\dot{\boldsymbol{e}}_{\mathrm{q}} + \boldsymbol{K}_{\mathrm{P}}\boldsymbol{e}_{\mathrm{q}}) + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{\tau}_{\mathrm{g}}(\boldsymbol{q}) \,,$$

and the resulting linear error dynamics are

$$\ddot{\boldsymbol{e}}_{\mathrm{q}} + \boldsymbol{K}_{\mathrm{V}}\dot{\boldsymbol{e}}_{\mathrm{q}} + \boldsymbol{K}_{\mathrm{P}}\boldsymbol{e}_{\mathrm{q}} = \boldsymbol{0} \,. \tag{6.52}$$

According to linear system theory, convergence of the tracking error to zero is guaranteed [6.29, 39].

Remark.

- One usually lets $\boldsymbol{K}_{\mathrm{V}}$ and $\boldsymbol{K}_{\mathrm{P}}$ be $(n \times n)$ diagonal positive-definite gain matrices (i.e., $\boldsymbol{K}_{\mathrm{V}} = \mathrm{diag}(k_{\mathrm{V},1}, \cdots, k_{\mathrm{V},n}) > \boldsymbol{0}, \boldsymbol{K}_{\mathrm{P}} = \mathrm{diag}(k_{\mathrm{P},1}, \cdots, k_{\mathrm{P},n}) > \boldsymbol{0}$) to guarantee the stability of the error system. However, the format of the foregoing control never leads to *independent joint control* because the outer-loop multiplier $\boldsymbol{H}(\boldsymbol{q})$ and the full nonlinear compensation term $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{\tau}_{\mathrm{g}}(\boldsymbol{q})$ in the

inner loop scramble all joint signals among different control channels.

6.6.2 Computed–Torque–Like Control

It is worth noting that the implementation of computed-torque control requires that parameters of the dynamical model are accurately known and the control input is computed in real time. In order to avoid those problems, several variations of this control scheme have been proposed, for example, computed-torque-like control. An entire class of computed-torque-like controllers can be obtained by modifying the computed-torque control as

$$\boldsymbol{\tau} = \hat{\boldsymbol{H}}(\boldsymbol{q})\boldsymbol{v} + \hat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{\tau}}(\boldsymbol{q}) \,, \tag{6.53}$$

where $\hat{}$ represents the computed or nominal value and indicates that the theoretically exact feedback linearization cannot be achieved in practice due to the uncertainty in the systems. The overall control scheme is depicted in Fig. 6.6.

Computed–Torque–Like Control with Variable–Structure Compensation
Since there is parametric uncertainty, compensation is required in the outer-loop design to achieve trajectory tracking. The following shows a computed-torque-like control with variable-structure compensation

$$\boldsymbol{v} = \ddot{\boldsymbol{q}}_{\mathrm{d}} + \boldsymbol{K}_{\mathrm{V}}\dot{\boldsymbol{e}}_{\mathrm{q}} + \boldsymbol{K}_{\mathrm{P}}\boldsymbol{e}_{\mathrm{q}} + \Delta\boldsymbol{v} \,, \tag{6.54}$$

where the variable-structure compensation $\Delta\boldsymbol{v}$ is devised as

$$\Delta\boldsymbol{v} = \begin{cases} -\rho(\boldsymbol{x},t)\dfrac{\boldsymbol{B}^{\top}\boldsymbol{P}\boldsymbol{x}}{\|\boldsymbol{B}^{\top}\boldsymbol{P}\boldsymbol{x}\|} & \text{if} \quad \|\boldsymbol{B}^{\top}\boldsymbol{P}\boldsymbol{x}\| \neq 0 \\ 0 & \text{if} \quad \|\boldsymbol{B}^{\top}\boldsymbol{P}\boldsymbol{x}\| = 0 \,, \end{cases} \tag{6.55}$$

where $\boldsymbol{x} = (\boldsymbol{e}_{\mathrm{q}}^{\top}, \dot{\boldsymbol{e}}_{\mathrm{q}}^{\top})^{\top}$, $\boldsymbol{B} = (\boldsymbol{0}, \boldsymbol{I}_n)^{\top}$, $\boldsymbol{P}$ is a $(2n \times 2n)$ symmetric positive-definite matrix satisfying

$$\boldsymbol{P}\boldsymbol{A} + \boldsymbol{A}^{\top}\boldsymbol{P} = -\boldsymbol{Q} \tag{6.56}$$

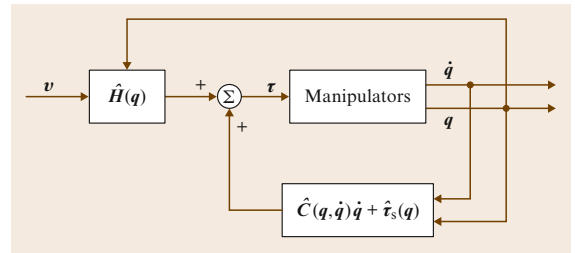**Fig. 6.6** Computed-torque-like control

with A being defined as

$$A = \begin{pmatrix} \mathbf{0} & I_n \\ -K_P & -K_V \end{pmatrix} , \qquad (6.57)$$

Q being any appropriate $(2n \times 2n)$ symmetric positive-definite matrix,

$$\rho(\mathbf{x}, t) = \frac{1}{1-\alpha}[\alpha\beta + \| K \| \| \mathbf{x} \| + \bar{H}\phi(\mathbf{x}, t)] , \qquad (6.58)$$

where α and β are positive constants such that $\|H^{-1}(q)\hat{H}(q) - I_n\| \le \alpha < 1$ for all $q \in R^n$ and $\sup_{t \in [0,\infty)} \| \ddot{q}_d(t) \| < \beta$, respectively, K is the $(n \times 2n)$ matrix defined as $K = [K_P \ K_V]$, $\bar{\lambda}_H$ being a positive constant such that $\| H^{-1}(q) \| \le \bar{\lambda}_H$ for all $q \in R^n$, and the function ϕ being defined as

$$\|[\hat{C}(q, \dot{q}) - C(q, \dot{q})]\dot{q} + [\hat{\tau}_g(q) - \tau_g(q)]\|$$
$$\le \phi(\mathbf{x}, t) . \qquad (6.59)$$

Convergence of the tracking error to zero can be shown using the Lyapunov function

$$V = \mathbf{x}^\top P \mathbf{x} , \qquad (6.60)$$

following the stability analysis in [6.5, 40].

Remarks.

- By Property 6.1 in Sect. 6.1, there exist positive constants $\bar{\lambda}_H$ and $\bar{\lambda}_h$ such that $\bar{\lambda}_h \le \| H^{-1}(q) \| \le \bar{\lambda}_H$ for all $q \in R^n$. If we choose

$$\hat{H} = \frac{1}{c}I_n , \quad \text{where} \quad c = \frac{\bar{\lambda}_H + \bar{\lambda}_h}{2} , \qquad (6.61)$$

it can be shown that

$$\| H^{-1}(q)\hat{H}(q) - I_n \| \le \frac{\bar{\lambda}_H - \bar{\lambda}_h}{\bar{\lambda}_H + \bar{\lambda}_h} \equiv \alpha < 1 , \qquad (6.62)$$

which indicates that there is always at least one choice of $\hat{H}$ for some $\alpha < 1$.

- Due to the discontinuity in $\Delta\mathbf{v}$, chattering phenomenon may occur when the control scheme is applied. It is worth noting that chattering is often undesirable since the high-frequency component in the control can excite unmodeled dynamic effect (such as joint flexibility) [6.6, 29, 38]. In order to avoid chattering, the variable-structure compensation can be modified to become smooth, i. e.,

$$\Delta\mathbf{v} = \begin{cases} -\rho(\mathbf{x}, t)\frac{B^\top P\mathbf{x}}{\| B^\top P\mathbf{x} \|} & \text{if} \quad \| B^\top P\mathbf{x} \| > \varepsilon , \\ \frac{-\rho(\mathbf{x}, t)}{\varepsilon}B^\top P\mathbf{x} & \text{if} \quad \| B^\top P\mathbf{x} \| \le \varepsilon , \end{cases}$$
$$(6.63)$$

where ε is a positive constant and is used as the boundary layer. Following this modification, convergence of tracking errors to a residual set can be ensured, and the size of this residual set can be made smaller by use of a smaller value of ε.

Computed–Torque–Like Control with Independent-Joint Compensation

Obviously, the previous compensation scheme is centralized, which implies that the online computation load is heavy and high-cost hardware is required for practical implementation. In order to solve this problem, a scheme with independent-joint compensation is introduced below. In this scheme, a computed-torque-like control is designed with estimates as

$$\hat{H}(q) = I , \quad \hat{C}(q, \dot{q}) = 0 , \quad \hat{\tau}(q) = \mathbf{0} . \qquad (6.64)$$

Then, we use the outer-loop $\mathbf{v}$ as

$$\mathbf{v} = K_V \dot{e}_q + K_P e_q + \Delta\mathbf{v} , \qquad (6.65)$$

where the positive constants K_P and K_V are selected to be sufficiently large, and the i-th component Δv_i of $\Delta\mathbf{v} = (\mathbf{v}_1, \dots, \mathbf{v}_n)^\top$ is

$$\Delta v_i = \begin{cases} -[\boldsymbol{\beta}^\top \mathbf{w}(q_d, \dot{q}_d)]^2 \frac{s_i}{\varepsilon_i} \\ \quad \text{if} \quad | s_i | \le \frac{\varepsilon_i}{\boldsymbol{\beta}^\top \mathbf{w}(q_d, \dot{q}_d)} , \\ -\boldsymbol{\beta}^\top \mathbf{w}(q_d, \dot{q}_d)\frac{s_i}{|s_i|} \\ \quad \text{if} \quad | s_i | > \frac{\varepsilon_i}{\boldsymbol{\beta}^\top \mathbf{w}(q_d, \dot{q}_d)} . \end{cases} \qquad (6.66)$$

In this compensation, $s_i = \dot{e}_{q,i} + \lambda_i e_{q,i}$, $i \in \{1, \dots, n\}$, and λ_i, $i \in \{1, \dots, n\}$ are positive constants. Furthermore, following the properties of robot manipulators, we have

$$\| H(q)\ddot{q}_d + C(q, \dot{q})\dot{q}_d + \tau_g(q) \|$$
$$\le \beta_1 + \beta_2 \| q \| + \| \dot{q} \| = \boldsymbol{\beta}^\top \mathbf{w}(q, \dot{q})$$

for some suitable positive constants β_1, β_2, and β_3, where $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3)^\top$ and

$$\mathbf{w}(q, \dot{q}) = [1, \| q \|, \| \dot{q} \|]^\top . \qquad (6.67)$$

Finally, ε_i, $i \in \{1, \dots, n\}$, is the variable length of the boundary layer, satisfying

$$\dot{\varepsilon}_i = -g_i \varepsilon_i , \quad \varepsilon(0) > 0 , \quad g_i > 0 . \qquad (6.68)$$

It is worth pointing out that the term $\boldsymbol{w}$ in this control scheme is devised as the desired compensation rather than feedback. Furthermore, this scheme is in a form of independent-joint control and hence has the advantages introduced before. The convergence of the tracking error to zero can be shown using the Lyapunov function

$$V = \frac{1}{2}\left(\boldsymbol{e}_{\mathrm{q}}^{\top}\, \dot{\boldsymbol{e}}_{\mathrm{q}}^{\top}\right)\begin{pmatrix} \lambda \boldsymbol{K}_{\mathrm{P}} & \boldsymbol{H} \\ \boldsymbol{H} & \lambda \boldsymbol{H} \end{pmatrix}\begin{pmatrix} \boldsymbol{e}_{\mathrm{q}} \\ \dot{\boldsymbol{e}}_{\mathrm{q}} \end{pmatrix} + \sum_{i=1}^{n} g_i^{-1} \varepsilon_i \, , \tag{6.69}$$

whose time derivative along the trajectory of the closed-loop systems follows

$$\dot{V} = -\alpha \left| \begin{pmatrix} \boldsymbol{e}_{\mathrm{q}} \\ \dot{\boldsymbol{e}}_{\mathrm{q}} \end{pmatrix} \right|^2 \tag{6.70}$$

with α being some positive constant, if sufficiently large $\boldsymbol{K}_{\mathrm{P}}$ and γ are used. The detailed analysis of stability, which requires Properties 6.3 and 6.4, can be found in [6.13].

Remark.

- Similar to computed-torque-like control with variable-structure compensation, we can consider the nonzero boundary layer as

$$\dot{\varepsilon}_i = -g_i \varepsilon_i \, , \quad \varepsilon(0) > 0 \, , \quad g_i \, , \, \alpha_i > 0 \, . \tag{6.71}$$

Following this modification, tracking errors converge to a residual set. The size of this residual set can be made smaller by the use of a smaller value of ε (i. e., smaller α_i).

For the task of point-to-point control, one PD controller with gravity compensation is designed with the estimates

$$\hat{\boldsymbol{H}}(\boldsymbol{q}) = \boldsymbol{I} \, , \quad \hat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{0} \, , \quad \hat{\boldsymbol{\tau}}_{\mathrm{g}}(\boldsymbol{q}) = \boldsymbol{\tau}_{\mathrm{g}}(\boldsymbol{q}) \tag{6.72}$$

with $\boldsymbol{\tau}_{\mathrm{g}}(\boldsymbol{q})$ being the *gravity term* of the dynamical model of the robot manipulators. Then, we use the outer-loop $\boldsymbol{v}$ as

$$\boldsymbol{v} = \boldsymbol{K}_{\mathrm{V}}\dot{\boldsymbol{e}}_{\mathrm{q}} + \boldsymbol{K}_{\mathrm{P}}\boldsymbol{e}_{\mathrm{q}} \, , \tag{6.73}$$

such that the control input becomes

$$\boldsymbol{\tau} = \boldsymbol{K}_{\mathrm{V}}\dot{\boldsymbol{e}}_{\mathrm{q}} + \boldsymbol{K}_{\mathrm{P}}\boldsymbol{e}_{\mathrm{q}} + \boldsymbol{\tau}_{\mathrm{g}}(\boldsymbol{q}) \, . \tag{6.74}$$

This scheme is much simpler to implement than the exact computed-torque controller. The convergence of the tracking error to zero can be shown using the Lyapunov function

$$V = \frac{1}{2}\dot{\boldsymbol{e}}_{\mathrm{q}}^{\top}\boldsymbol{H}(\boldsymbol{q})\dot{\boldsymbol{e}}_{\mathrm{q}} + \frac{1}{2}\boldsymbol{e}_{\mathrm{q}}^{\top}\boldsymbol{K}_{\mathrm{P}}\boldsymbol{e}_{\mathrm{q}} \, , \tag{6.75}$$

whose time derivative along the solution trajectories of the closed-loop system is

$$\dot{V} = -\dot{\boldsymbol{e}}_{\mathrm{q}}^{\top}\boldsymbol{K}_{\mathrm{V}}\dot{\boldsymbol{e}}_{\mathrm{q}} \, . \tag{6.76}$$

The detailed analysis of stability is given in [6.12]. It is necessary to note that this result is for the case of regulation, rather than for the case of tracking, since the former theoretical base, which relies on LaSalle's lemma, requires the system be autonomous (time invariant) [6.38, 41, 42].

Remark.

- If we neglect gravity in the dynamical model of the robot manipulators, then the gravity estimation can be omitted here, i. e., $\hat{\boldsymbol{\tau}}_{\mathrm{g}}(\boldsymbol{q}) = \boldsymbol{0}$, so that the control law becomes

$$\boldsymbol{\tau} = \boldsymbol{v} = \boldsymbol{K}_{\mathrm{V}}\dot{\boldsymbol{e}}_{\mathrm{q}} + \boldsymbol{K}_{\mathrm{P}}\boldsymbol{e}_{\mathrm{q}} \, , \tag{6.77}$$

which leads to pure PD control. The gain matrices, $\boldsymbol{K}_{\mathrm{P}}$ and $\boldsymbol{K}_{\mathrm{V}}$, can be selected to be diagonal so that this PD control is in a form of independent-joint control developed based on the multijoint dynamic model.

6.6.3 Summary

In this section, we have presented two control schemes: the computed-torque and computed-torque-like control. The former transforms a multi-input multi-output (MIMO) nonlinear robotic system into a very simple decoupled linear closed-loop system whose control design is a well-established problem. Since the practical implementation of the former control requires preknowledge of all the manipulator parameters and its payload, which may not be realistic, the latter was introduced to relax the foregoing requirement and still to achieve the objective of tracking subject to system's uncertainty.

Further Reading

The PD control with different feedforward compensation for tracking control is investigated in [6.43]. An adaptive control scheme based on PD control is presented in [6.19].

6.7 Adaptive Control

An adaptive controller differs from an ordinary controller in that the controller parameters are time varying, and there is a mechanism for adjusting these parameters online based on some signals in the closed-loop systems. By the use of such control scheme, the control goal can be achieved even if there is parametric uncertainty in the plant. In this section, we will introduce several adaptive control schemes to deal with the case of imperfect knowledge of dynamical parameters of the robot manipulators. The control performance of those adaptive control schemes, including adaptive computed-torque control, adaptive inertia-related control, adaptive control based on passivity, and adaptive control with desired compensation, are basically derived from Property 6.5. Finally, the condition of persistent excitation, which is important in parameter convergence, will be addressed.

6.7.1 Adaptive Computed-Torque Control

The computed-torque control scheme is appealing, since it allows the designer to transform a MIMO highly coupled nonlinear system into a very simple decoupled linear system, whose control design is a well-established problem. However, this method of feedback linearization relies on perfect knowledge of the system parameters, and failure to have this will cause erroneous parametric estimates, resulting in a mismatch term in the closed-loop model of the error system. That term can be interpreted as a nonlinear perturbation acting at the input of the closed-loop system. In order to solve the problem due to parametric uncertainty, we instead consider the inverse dynamics method with parameter estimates of

$$\boldsymbol{\tau} = \hat{\boldsymbol{H}}(\boldsymbol{q})(\ddot{\boldsymbol{q}}_{\mathrm{d}} + \boldsymbol{K}_{\mathrm{V}}\dot{\boldsymbol{e}}_{\mathrm{q}} + \boldsymbol{K}_{\mathrm{P}}\boldsymbol{e}_{\mathrm{q}}) + \hat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{\tau}}_{\mathrm{g}}(\boldsymbol{q}) , \tag{6.78}$$

where $\hat{\boldsymbol{H}}, \hat{\boldsymbol{C}}, \hat{\boldsymbol{\tau}}_{\mathrm{g}}$ have the same functional form as $\boldsymbol{H}, \boldsymbol{C}, \boldsymbol{\tau}_{\mathrm{g}}$. From Property 6.5 of the dynamics model, we have

$$\hat{\boldsymbol{H}}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \hat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \hat{\boldsymbol{\tau}}_{\mathrm{g}}(\boldsymbol{q}) = \boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\hat{\boldsymbol{a}} , \tag{6.79}$$

where $\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})$, called the regressor, is a known $(n \times r)$ function matrix and $\boldsymbol{a}$ is the $(r \times 1)$ vector that summarizes all the estimated parameters. Substituting this control input $\boldsymbol{\tau}$ into the manipulator dynamics gives the following closed-loop error model

$$\hat{\boldsymbol{H}}(\boldsymbol{q})(\ddot{\boldsymbol{e}}_{\mathrm{q}} + \boldsymbol{K}_{\mathrm{V}}\dot{\boldsymbol{e}}_{\mathrm{q}} + \boldsymbol{K}_{\mathrm{P}}\boldsymbol{e}_{\mathrm{q}}) = \boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\tilde{\boldsymbol{a}} , \tag{6.80}$$

where $\tilde{\boldsymbol{a}} = \hat{\boldsymbol{a}} - \boldsymbol{a}$. In order to acquire an appropriate adaptive law, we first assume that the acceleration term $\ddot{\boldsymbol{q}}$ is

measurable, and that the estimated inertia matrix $\hat{\boldsymbol{H}}(\boldsymbol{q})$ is never singular. Now, for convenience, the error equation is rewritten as

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\hat{\boldsymbol{H}}^{-1}(\boldsymbol{q})\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\tilde{\boldsymbol{a}} \tag{6.81}$$

with $\boldsymbol{x} = \left(\boldsymbol{e}_{\mathrm{q}}^{\top}, \dot{\boldsymbol{e}}_{\mathrm{q}}^{\top}\right)^{\top}$,

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{0}_n & \boldsymbol{I}_n \\ -\boldsymbol{K}_{\mathrm{P}} & -\boldsymbol{K}_{\mathrm{V}} \end{pmatrix} , \quad \boldsymbol{B} = \begin{pmatrix} \boldsymbol{0}_n \\ \boldsymbol{I}_n \end{pmatrix} . \tag{6.82}$$

The adaptive law is considered as

$$\dot{\hat{\boldsymbol{a}}} = -\boldsymbol{\Gamma}^{-1}\boldsymbol{Y}^{\top}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\hat{\boldsymbol{H}}^{-1}(\boldsymbol{q})\boldsymbol{B}^{\top}\boldsymbol{P}\boldsymbol{x} , \tag{6.83}$$

where $\boldsymbol{\Gamma}$ is an $(r \times r)$ positive-definite constant matrix, and $\boldsymbol{P}$ is a $(2n \times 2n)$ symmetric positive-definite constant matrix satisfying

$$\boldsymbol{P}\boldsymbol{A} + \boldsymbol{A}^{\top}\boldsymbol{P} = -\boldsymbol{Q} , \tag{6.84}$$

with $\boldsymbol{Q}$ being a symmetric positive-definite constant matrix with coherent dimension. In this adaptive law, we made two assumptions:

- the joint acceleration $\ddot{\boldsymbol{q}}$ is measurable, and
- the bounded range of the unknown parameter is available.

The first assumption is to ensure that the regressor $\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})$ is known a priori, whereas the second assumption is to allow one to keep the estimate $\hat{\boldsymbol{H}}(\boldsymbol{q})$ nonsingular by restricting the estimated parameter $\hat{\boldsymbol{a}}$ to lie within a range about the true parameter value.

Convergence of the tracking error and maintaining boundedness of all internal signals can actually be guaranteed by Lyapunov stability theory with the Lyapunov function

$$\dot{V} = -\boldsymbol{x}^{\top}\boldsymbol{Q}\boldsymbol{x} . \tag{6.85}$$

Detailed stability analysis is given in [6.2].

Remark.

- For practical and theoretical reasons, the first assumption above is hardly acceptable. In most cases, it is not easy to obtain an accurate measure of acceleration; the robustness of the above adaptive control scheme with respect to such a disturbance has to be established. Moreover, from a pure theoretical viewpoint, measuring $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}$ means that not only do we

need the whole system state vector, but we also need its derivative.

6.7.2 Adaptive Inertia–Related Control

Another adaptive control scheme is now introduced. This proposed scheme does not require the measurement of the manipulator's acceleration nor does it require inversion of the estimated inertia matrix. Hence, the drawbacks of the adaptive computed-torque control scheme are avoided. Let us consider the control input

$$\boldsymbol{\tau} = \hat{\boldsymbol{H}}(\boldsymbol{q})\dot{\boldsymbol{v}} + \hat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{v} + \hat{\boldsymbol{\tau}}_{\text{g}}(\boldsymbol{q}) + \boldsymbol{K}_{\text{D}}\boldsymbol{s} , \qquad (6.86)$$

where the auxiliary signals $\boldsymbol{v}$ and $\boldsymbol{s}$ are defined as $\boldsymbol{v} = \dot{\boldsymbol{q}}_{\text{d}} + \boldsymbol{\Lambda}\boldsymbol{e}_{\text{q}}$ and $\boldsymbol{s} = \boldsymbol{v} - \dot{\boldsymbol{q}} = \dot{\boldsymbol{e}}_{\text{q}} + \boldsymbol{\Lambda}\boldsymbol{e}_{\text{q}}$, with $\boldsymbol{\Lambda}$ being an $(n \times n)$ positive-definite matrix. Following Property 6.5 of the dynamic model, we have

$$\boldsymbol{H}(\boldsymbol{q})\dot{\boldsymbol{v}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{v} + \boldsymbol{\tau}_{\text{g}}(\boldsymbol{q}) = \bar{\boldsymbol{Y}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{v}, \dot{\boldsymbol{v}})\boldsymbol{a} , \quad (6.87)$$

where $\bar{\boldsymbol{Y}}(\cdot, \cdot, \cdot, \cdot)$ is an $(n \times r)$ matrix of known time functions. The formulation above is the same type of the parameter separation that was used in the formulation of the adaptive computed-torque control. Note that $\bar{\boldsymbol{Y}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{v}, \dot{\boldsymbol{v}})$ is independent of the joint acceleration. Similar to the formulation above, we also have

$$\hat{\boldsymbol{H}}(\boldsymbol{q})\dot{\boldsymbol{v}} + \hat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{v} + \hat{\boldsymbol{\tau}}_{\text{g}}(\boldsymbol{q}) = \bar{\boldsymbol{Y}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{v}, \dot{\boldsymbol{v}})\hat{\boldsymbol{a}} . \quad (6.88)$$

Substituting the control input into the equation of motion, it follows that

$$\begin{aligned} \boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} &+ \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{\tau}_{\text{g}}(\boldsymbol{p}) \\ &= \hat{\boldsymbol{H}}(\boldsymbol{q})\dot{\boldsymbol{v}} + \hat{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{v} + \hat{\boldsymbol{\tau}}_{\text{g}}(\boldsymbol{q}) + \boldsymbol{K}_{\text{D}}\boldsymbol{s} . \end{aligned}$$

Since $\ddot{\boldsymbol{q}} = \dot{\boldsymbol{v}} - \dot{\boldsymbol{s}}, \dot{\boldsymbol{q}} = \boldsymbol{v} - \boldsymbol{s}$, the previous result can be rewritten as

$$\boldsymbol{H}(\boldsymbol{q})\dot{\boldsymbol{s}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{s} + \boldsymbol{K}_{\text{D}}\boldsymbol{s} = \bar{\boldsymbol{Y}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{v}, \dot{\boldsymbol{v}})\tilde{\boldsymbol{a}} , \quad (6.89)$$

where $\tilde{\boldsymbol{a}} = \hat{\boldsymbol{a}} - \boldsymbol{a}$. The adaptive law is considered as

$$\dot{\hat{\boldsymbol{a}}} = -\boldsymbol{\Gamma}\bar{\boldsymbol{Y}}^{\top}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{v}, \dot{\boldsymbol{v}})\boldsymbol{s} . \qquad (6.90)$$

The convergence of the tracking error to zero with boundedness on all internal signals can be shown through Lyapunov stability theory using the following Lyapunov-like function

$$V = \frac{1}{2}\boldsymbol{s}^{\top}\boldsymbol{H}(\boldsymbol{q})\boldsymbol{s} + \frac{1}{2}\tilde{\boldsymbol{a}}^{\top}\boldsymbol{\Gamma}^{-1}\tilde{\boldsymbol{a}} , \qquad (6.91)$$

whose time derivative along the trajectories of the closed-loop systems can be found to be

$$\dot{V} = -\boldsymbol{s}^{\top}\boldsymbol{K}_{\text{D}}\boldsymbol{s} . \qquad (6.92)$$

The detailed stability analysis is given in [6.32].

Remark.

- The restrictions for adaptive computed-torque control formerly seen have been removed here.
- The term $\boldsymbol{K}_{\text{D}}\boldsymbol{s}$ introduces a PD-type linear stabilizing control action to the error system model.
- The estimated parameters converge to the true parameters provided the reference trajectory satisfies the condition of persistency of excitation,

$$\begin{aligned} \alpha_1 \boldsymbol{I}_r &\leq \int_{t_0}^{t_0+t} \boldsymbol{Y}^{\top}(\boldsymbol{q}_{\text{d}}, \dot{\boldsymbol{q}}_{\text{d}}, \boldsymbol{v}, \dot{\boldsymbol{v}})\boldsymbol{Y}(\boldsymbol{q}_{\text{d}}, \dot{\boldsymbol{q}}_{\text{d}}, \boldsymbol{v}, \dot{\boldsymbol{v}})\mathrm{d}t \\ &\leq \alpha_2 \boldsymbol{I}_r , \end{aligned}$$

for all t_0, where α_1, α_2, and t are all positive constants.

6.7.3 Adaptive Control Based on Passivity

Taking a physics viewpoint of control, we see that the concept of passivity has become popular for the development of adaptive control schemes. Here, we illustrate how the concept of passivity can be used to design a class of adaptive control laws for robot manipulators. First, we define an auxiliary filtered tracking error signal $\boldsymbol{r}$ as

$$\boldsymbol{r} = \boldsymbol{F}^{-1}(s)\boldsymbol{e}_{\text{q}} , \qquad (6.93)$$

where

$$\boldsymbol{F}^{-1}(s) = \left[s\boldsymbol{I}_n + \frac{1}{s}\boldsymbol{K}(s)\right] \qquad (6.94)$$

and s is the Laplace transform variable. The $(n \times n)$ matrix $\boldsymbol{K}(s)$ is chosen such that $\boldsymbol{F}(s)$ is a strictly proper, stable transfer function matrix. As in the preceding schemes, the adaptive control strategies has close ties to the ability to separate the known functions from the unknown constant parameters. We use the expression given above to define

$$\begin{aligned} \boldsymbol{Z}\boldsymbol{\varphi} &= \boldsymbol{H}(\boldsymbol{q})[\ddot{\boldsymbol{q}}_{\text{d}} + \boldsymbol{K}(s)\boldsymbol{e}_{\text{q}}] \\ &+ \boldsymbol{V}(\boldsymbol{q}, \dot{\boldsymbol{q}})\left[\dot{\boldsymbol{q}}_n + \frac{1}{s}\boldsymbol{K}(s)\boldsymbol{e}_{\text{q}}\right] + \boldsymbol{\tau}_{\text{g}}(\boldsymbol{q}) , \end{aligned}$$

where $\boldsymbol{Z}$ is a known $(n \times r)$ regression matrix and $\boldsymbol{\varphi}$ is a vector of unknown system parameters in the adaptive context. It is important to note that the above can be arranged such that $\boldsymbol{Z}$ and $\boldsymbol{r}$ do not depend on the measurement of the joint acceleration $\ddot{\boldsymbol{q}}$. The adaptive control scheme given here is called the passivity approach because the mapping of $-\boldsymbol{r} \rightarrow \boldsymbol{Z}\tilde{\boldsymbol{\varphi}}$ is constructed to be

a passive mapping. That is, we develop an adaptive law such that

$$\int_0^t -\boldsymbol{r}^\top(\sigma)\boldsymbol{Z}(\sigma)\tilde{\boldsymbol{\varphi}}(\sigma)\,\mathrm{d}\sigma \geq -\beta \tag{6.95}$$

is satisfied for all time and for some positive scalar constant β. For this class of adaptive controllers, the control input is given by

$$\boldsymbol{\tau} = \boldsymbol{Z}\hat{\boldsymbol{\varphi}} + \boldsymbol{K}_\mathrm{D}\boldsymbol{r} , \tag{6.96}$$

Detailed analysis of stability is given in [6.44].

Remarks.

- If $\boldsymbol{K}(s)$ is selected such that $\boldsymbol{H}(s)$ has a relative degree of one, $\boldsymbol{Z}$ and $\boldsymbol{r}$ will not depend on $\ddot{\boldsymbol{q}}$.
- Many types of control schemes can be generated from the adaptive passive control approach by selected different transfer function matrices $\boldsymbol{K}(s)$ in the definition of $\boldsymbol{r}$.
- Note that, by defining $\boldsymbol{K}(s) = s\boldsymbol{\Lambda}$ such that $\boldsymbol{F}(s) = (s\boldsymbol{I}_n + \boldsymbol{\Lambda})^{-1}$, we have the control input

  $$\boldsymbol{\tau} = \boldsymbol{Z}\hat{\boldsymbol{\varphi}} - \boldsymbol{K}_\mathrm{D}\boldsymbol{r} ,$$

  with

  $$\begin{aligned}\boldsymbol{Z}\hat{\boldsymbol{\varphi}} = &\hat{\boldsymbol{H}}(\boldsymbol{q})(\ddot{\boldsymbol{q}}_\mathrm{d} + \boldsymbol{\Lambda}\dot{\boldsymbol{e}}_\mathrm{q}) + \hat{\boldsymbol{C}}(\boldsymbol{q},\dot{\boldsymbol{q}})(\dot{\boldsymbol{q}}_\mathrm{d} + \boldsymbol{\Lambda}\boldsymbol{e}_\mathrm{q}) \\ &+ \hat{\boldsymbol{\tau}}_\mathrm{g}(\boldsymbol{q}) .\end{aligned}$$

  The adaptive law may be chosen as

  $$\dot{\hat{\boldsymbol{\varphi}}} = \boldsymbol{\Gamma}\boldsymbol{Z}^\top(\dot{\boldsymbol{e}}_\mathrm{q} + \boldsymbol{\Lambda}\boldsymbol{e}_\mathrm{q})$$

  to satisfy the condition of passive mapping. This indicates that adaptive inertia-related control can be viewed as a special case of adaptive passive control.

6.7.4 Adaptive Control with Desired Compensation

In order to implement the adaptive control scheme, one needs to calculate the elements of $\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})$ in real time. However, this procedure may be excessively time consuming since it involves computations with highly nonlinear functions of joint position and velocity. Consequently, the real-time implementation of such a scheme is rather difficult. To overcome this difficulty, the adaptive control with desired compensation was proposed and is discussed here. In other words, the variables $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, and $\ddot{\boldsymbol{q}}$ are replaced with the desired ones, namely, $\boldsymbol{q}_\mathrm{d}$,

$\dot{\boldsymbol{q}}_\mathrm{d}$, and $\ddot{\boldsymbol{q}}_\mathrm{d}$. Since the desired quantities are known in advance, all their corresponding calculations can be performed offline, which renders real-time implementation more plausible. Let us consider the control input

$$\boldsymbol{\tau} = \boldsymbol{Y}(\boldsymbol{q}_\mathrm{d},\dot{\boldsymbol{q}}_\mathrm{d},\ddot{\boldsymbol{q}}_\mathrm{d})\hat{\boldsymbol{a}} + k_\mathrm{a}\boldsymbol{s} + k_\mathrm{p}\boldsymbol{e}_\mathrm{q} + k_n\|\boldsymbol{e}_\mathrm{q}\|^2\boldsymbol{s} , \tag{6.97}$$

where the positive constants k_a, k_p, and k_n are sufficiently large, and the auxiliary signal $\boldsymbol{s}$ is defined as $\boldsymbol{s} = \dot{\boldsymbol{e}}_\mathrm{q} + \boldsymbol{e}_\mathrm{q}$. The adaptive law is considered as

$$\dot{\hat{\boldsymbol{a}}} = -\boldsymbol{\Gamma}\boldsymbol{Y}^\top(\boldsymbol{q}_\mathrm{d},\dot{\boldsymbol{q}}_\mathrm{d},\ddot{\boldsymbol{q}}_\mathrm{d})\boldsymbol{s} . \tag{6.98}$$

It is worth noting that the desired compensation is adopted in both the control and adaptive laws such that the computational load can be drastically reduced. For the sake of this analysis, we note that

$$\begin{aligned}&\|\boldsymbol{Y}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\boldsymbol{a} - \boldsymbol{Y}(\boldsymbol{q}_\mathrm{d},\dot{\boldsymbol{q}}_\mathrm{d},\ddot{\boldsymbol{q}}_\mathrm{d})\hat{\boldsymbol{a}}\| \\ &\quad \leq \zeta_1\|\boldsymbol{e}_\mathrm{q}\| + \zeta_2\|\boldsymbol{e}_\mathrm{q}\|^2 + \zeta_3\|\boldsymbol{s}\| + \zeta_4\|\boldsymbol{s}\|\|\boldsymbol{e}_\mathrm{q}\| ,\end{aligned}$$

where ζ_1, ζ_2, ζ_3, and ζ_4 are positive constants. In order to achieve trajectory tracking, it is required that

$$k_\mathrm{a} > \zeta_2 + \zeta_4 ,$$
$$k_\mathrm{p} > \frac{\zeta_1}{2} + \frac{\zeta_2}{4} ,$$
$$k_\mathrm{v} > \frac{\zeta_1}{2} + \zeta_3 + \frac{\zeta_2}{4} ,$$

(i.e., the gains k_a, k_p, and k_v must be sufficiently large). The convergence of the tracking error to zero with boundedness on all internal signals can be proved through application of Lyapunov stability theory with the following Lyapunov-like function

$$V = \frac{1}{2}\boldsymbol{s}^\top\boldsymbol{H}(\boldsymbol{q})\boldsymbol{s} + \frac{1}{2}k_\mathrm{p}\boldsymbol{e}_\mathrm{q}^\top\boldsymbol{e}_\mathrm{q} + \frac{1}{2}\tilde{\boldsymbol{a}}^\top\boldsymbol{\Gamma}^{-1}\tilde{\boldsymbol{a}} , \tag{6.99}$$

whose time derivative along the trajectories of the closed-loop system can be derived as

$$\dot{V} \leq -\boldsymbol{x}^\top\boldsymbol{Q}\boldsymbol{x} , \tag{6.100}$$

where

$$\boldsymbol{x} = \begin{pmatrix}\|\boldsymbol{e}_\mathrm{q}\| \\ \|\boldsymbol{s}\|\end{pmatrix} , \quad \boldsymbol{Q} = \begin{pmatrix}k_\mathrm{p} - \zeta_2/4 & -\zeta_1/2 \\ -\zeta_1/2 & k_\mathrm{v} - \zeta_3 - \zeta_4/4\end{pmatrix} .$$

Detailed stability analysis can be found in [6.45].

6.7.5 Summary

Since the computed-torque control suffers from parametric uncertainty, a variety of adaptive control schemes

have been proposed. Firstly, we have presented an adaptive control scheme based on computed-torque control. Then, in order to overcome the mentioned drawbacks such as the measurability of the joint acceleration and the invertibility of the estimated inertia matrix, we presented an alternative adaptive control scheme that is free of these drawbacks. Recently, to incorporate a physics viewpoint into control, adaptive passivity-based control has become popular, and hence is introduced and discussed here. Finally, to reduce the computational load of the adaptive schemes, we presented an adaptive control with desired compensation.

Further Reading

A computationally very fast scheme dealing with adaptive control of rigid manipulators was presented in [6.46]. The stability analysis was completed by assuming that the joint dynamics are decoupled, i. e., that each joint is considered as an independent second-order linear system. Other pioneering works in the field can be found, for example, in [6.47, 48]; although none of the fundamental dynamic model properties are used, the complete dynamics are taken into account, but the control input is discontinuous and may lead to chattering. Positive definiteness of the inertia matrix is explicitly used in [6.49], although it was assumed that some time-varying quantities remain constant during the adap-

tation. It is interesting to note that all of these schemes were based on the concept of model reference adaptive control (MRAC) developed in [6.50] for linear systems. Therefore, they are conceptually very different from the truly nonlinear schemes presented in this section.

A passive-based modified version of the least-squares estimation scheme has been proposed in [6.51] and [6.52], which guaranteed closed-loop stability of the scheme. Other schemes can be found in [6.53], where no use is made of the skew-symmetry property, and in [6.54], where the recursive Newton–Euler formulations is used instead of the Lagrange formulation to derive the manipulator dynamics, and thus computation is simplified to facilitate practical implementation.

Even though adaptive control provides a solution to the problem of parametric uncertainty, the robustness of adaptive controllers remains a topic of great interest in the field. Indeed, measurement noise or unmodeled dynamics (e.g., flexibility) may result in unbounded closed-loop signals. In particular, the estimated parameters may diverge; this is a well-known phenomenon in adaptive control and is called parameter drift. Solutions inspired from the adaptive control of linear systems have been studied [6.55, 56], where a modified estimation ensures boundedness of the estimates. In [6.57], the controller in [6.32] is modified to enhance robustness.

6.8 Optimal and Robust Control

Given a nonlinear system, such as robotic manipulators, one can develop many stabilizing controls [6.29, 41]. In other words, the stability of the control system cannot determine a unique controller. It is natural that one seeks an optimal controller among the many stable ones. However, the design of an optimal controller is possible provided that a rather exact information on the target system is available, such as an exact system model [6.34, 58]. In the presence of discrepancy between the real system and its mathematical model, a designed optimal controller is no longer optimal, and may even end up being instable in the actual system. Generally speaking, the optimal control design framework is not the best one to deal with system uncertainty. To handle system uncertainty from the control design stage, a robust control design framework is necessary [6.59]. One of main objectives of robust control is to keep the controlled system stabile even in presence of uncertainties in the mathematical model, unmodeled dynamics, and the like.

Let us consider an affine nonlinear system described by nonlinear time-varying differential equation in the state $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^\top \in R^n$

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}, t) + \boldsymbol{G}(\boldsymbol{x}, t)\boldsymbol{u} + \boldsymbol{P}(\boldsymbol{x}, t)\boldsymbol{w} \,, \qquad (6.101)$$

where $\boldsymbol{u} \in R^m$ is the control input, and $\boldsymbol{w} \in R^w$ is the disturbance. Without disturbances or unmodeled dynamics, the system simplifies to

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}, t) + \boldsymbol{G}(\boldsymbol{x}, t)\boldsymbol{u} \,. \qquad (6.102)$$

Actually, there are many kinds of methods describing the nonlinear system according to the objective of control [6.1, 16, 21, 23, 34, 53].

6.8.1 Quadratic Optimal Control

Every optimal controller is based on its own cost function [6.60, 61]. One can define a cost function as [6.62, 63]

$$\boldsymbol{z} = \boldsymbol{H}(\boldsymbol{x}, t)\boldsymbol{x} + \boldsymbol{K}(\boldsymbol{x}, t)\boldsymbol{u}$$

such that $H^\top(x, t)K(x, t) = 0$, $K^\top(x, t)K(x, t) = R(x, t)$ > 0, and $H^\top(x, t)H(x, t) = Q(x, t) > 0$. Then, we have

$$\frac{1}{2}z^\top z = \frac{1}{2}x^\top Q(x, t)x + \frac{1}{2}u^\top R(x, t)u .$$

The quadratic optimal control for the system (6.102) is found by solving, for a first order differentiable positive-definite function $V(x, t)$, the Hamilton–Jacobi–Bellman (HJB) equation [6.34, 58]

$$0 = \text{HJB}(x, t; V) = V_t(x, t) + V_x(x, t)f(x, t)$$
$$- \frac{1}{2}V_x(x, t)G(x, t)R^{-1}(x, t)G^\top(x, t)V_x^\top(x, t)$$
$$+ \frac{1}{2}Q(x, t) ,$$

where $V_t = \dfrac{\partial V}{\partial t}$ and $V_x = \dfrac{\partial V}{\partial x^\top}$. Then the quadratic optimal control is defined by

$$u = -R^{-1}(x, t)G^\top(x, t)V_x^\top(x, t) . \tag{6.103}$$

Note that the HJB equation is a nonlinear second-order partial differential equation in $V(x, t)$.

Unlike the aforementioned optimal control problem, the so-called inverse quadratic optimal control problem is to find a set of $Q(x, t)$ and $R(x, t)$ for which the HJB equation has a solution $V(x, t)$. Then the inverse quadratic optimal control is defined by (6.103).

6.8.2 Nonlinear $\mathcal{H}_\infty$ Control

When disturbances are not negligible, one can deal with their effect such that

$$\int_0^t z^\top(x, \tau)z(x, \tau)\,\mathrm{d}\tau \le \gamma^2 \int_0^t w^\top w\,\mathrm{d}\tau , \tag{6.104}$$

where $\gamma > 0$ specifies the L_2-gain of the closed-loop system from the disturbance input w to the cost variable z. This is called the L_2-gain attenuation requirement [6.62–64]. One systematic way to design an optimal and robust control is given by the nonlinear $\mathcal{H}_\infty$ optimal control. Let $\gamma > 0$ be given, by solving the following equation:

$$HJI_\gamma(x, t; V)$$
$$= V_t(x, t) + V_x(x, t)f(x, t)$$
$$- \frac{1}{2}V_x(x, t)\{G(x, t)R^{-1}(x, t)G^\top(x, t)$$
$$- \gamma^{-2}P(x, t)P^\top(x, t)\}V_x^\top(x, t)$$
$$+ \frac{1}{2}Q(x, t) \le 0 , \tag{6.105}$$

then the control is defined by

$$u = -R^{-1}(x, t)G^\top(x, t)V_x^\top(x, t) . \tag{6.106}$$

The partial differential inequality (6.105) is called the Hamilton–Jacobi–Isaac (HJI) inequality. Then one can define the inverse nonlinear $\mathcal{H}_\infty$ optimal control problem that finds a set of $Q(x, t)$ and $R(x, t)$ such that the L_2-gain requirement is achieved for a prescribed L_2-gain γ [6.65].

Two things deserve further comment. The first is that the L_2-gain requirement is only valid for disturbance signals w whose L_2-norm is bounded. The second is that $\mathcal{H}_\infty$ optimal control is not uniquely defined. Hence, one can choose a quadratic optimal among many $\mathcal{H}_\infty$ optimal controllers. Precisely speaking, the control (6.106) should be called $\mathcal{H}_\infty$ *suboptimal* control, since the desired L_2-gain is prescribed a priori. A true $\mathcal{H}_\infty$ optimal control is to find a minimal value of γ, such that the L_2-gain requirement is achieved.

6.8.3 Passivity–Based Design of Nonlinear $\mathcal{H}_\infty$ Control

There are many methodologies for the design of optimal and/or robust controls. Among these, passivity-based controls can take full advantage of the properties described above [6.31]. They consist of two parts: one coming from the reference motion compensation while preserving the passivity of the system, and the other to achieve stability, robustness, and/or optimality [6.65, 66].

Let us suppose that the dynamic parameters are identified as $\widehat{H}(q)$, $\widehat{C}(q, \dot{q})$, and $\widehat{\tau}_g(q)$, whose counterparts are $H(q)$, $C(q, \dot{q})$, and $\tau_g(q)$, respectively. Then, passivity-based control generates the following tracking control laws

$$\tau = \widehat{H}(q)\ddot{q}_{\text{ref}} + \widehat{C}(q, \dot{q})\dot{q}_{\text{ref}} + \widehat{\tau}_g(q) - u , \tag{6.107}$$

where $\ddot{q}_{\text{ref}}$ is the reference acceleration defined by

$$\ddot{q}_{\text{ref}} = \ddot{q}_d + K_V\dot{e}_q + K_P e_q , \tag{6.108}$$

where $K_V = \text{diag}\{k_{V,i}\} > 0$ and $K_P = \text{diag}\{k_{P,i}\} > 0$. Two parameters are involved in generating the reference acceleration. Sometimes the following alternative method can be adopted

$$\ddot{q}_{\text{ref}} = \ddot{q}_d + K_V\dot{e}_q .$$

This reduces the order of the closed-loop system because the state $x = (e_q^\top, \dot{e}_q^\top)^\top$ is sufficient for the system
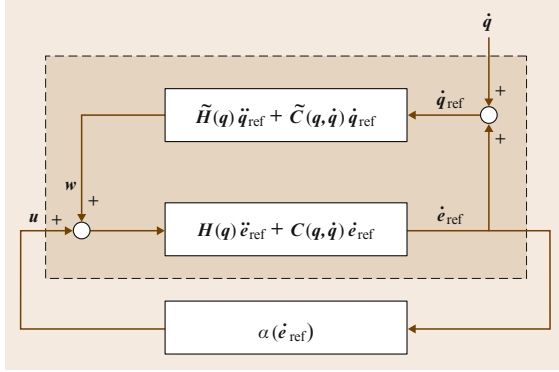
Fig. 6.7 The closed-loop system according to (6.109)

description, while the definition of (6.108) requires the state $\boldsymbol{x} = (\int \boldsymbol{e}_q^\top, \boldsymbol{e}_q^\top, \dot{\boldsymbol{e}}_q^\top)^\top$.

The closed-loop dynamics under the control is given by

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{e}}_{\text{ref}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{e}}_{\text{ref}} = \boldsymbol{u} + \boldsymbol{w}, \qquad (6.109)$$

where

$$\ddot{\boldsymbol{e}}_{\text{ref}} = \ddot{\boldsymbol{e}}_q + \boldsymbol{K}_{\text{V}}\dot{\boldsymbol{e}}_q + \boldsymbol{K}_{\text{P}}\boldsymbol{e}_q,$$

$$\dot{\boldsymbol{e}}_{\text{ref}} = \dot{\boldsymbol{e}}_q + \boldsymbol{K}_{\text{V}}\boldsymbol{e}_q + \boldsymbol{K}_{\text{P}}\int \boldsymbol{e}_q.$$

If $d(t) = 0$ and $\widehat{\boldsymbol{H}} = \boldsymbol{H}$, $\widehat{\boldsymbol{C}} = \boldsymbol{C}$, $\widehat{\boldsymbol{\tau}}_g = \boldsymbol{\tau}_g$, then $\boldsymbol{w} = 0$. Otherwise, the disturbance is defined as

$$\boldsymbol{w} = \widetilde{\boldsymbol{H}}(\boldsymbol{q})\ddot{\boldsymbol{q}}_{\text{ref}} + \widetilde{\boldsymbol{C}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}_{\text{ref}} + \widetilde{\boldsymbol{\tau}}_g(\boldsymbol{q}) + d(t), \qquad (6.110)$$

where $\widetilde{\boldsymbol{H}} = \boldsymbol{H} - \widehat{\boldsymbol{H}}$, $\widetilde{\boldsymbol{C}} = \boldsymbol{C} - \widehat{\boldsymbol{C}}$, and $\widetilde{\boldsymbol{\tau}}_g = \widehat{\boldsymbol{\tau}}_g - \boldsymbol{\tau}_g$. It is of particular interest that the system (6.109) defines a passive mapping between $\boldsymbol{u} + \boldsymbol{w}$ and $\dot{\boldsymbol{e}}_{\text{ref}}$.

According to the manner in which the auxiliary control input $\boldsymbol{u}$ is specified, passivity-based control can achieve stability, robustness, and/or optimality.

6.8.4 A Solution to Inverse Nonlinear $\mathcal{H}_\infty$ Control

Let us define the auxiliary control input by the reference-error feedback

$$\boldsymbol{u} = -\alpha \boldsymbol{R}^{-1}(\boldsymbol{x}, t)\dot{\boldsymbol{e}}_{\text{ref}}, \qquad (6.111)$$

where $\alpha > 1$ is arbitrary. Then, the control provides the inverse nonlinear $\mathcal{H}_\infty$ optimality.

Theorem 6.1 Inverse Nonlinear $\mathcal{H}_\infty$ Optimality [6.65]
Let the reference acceleration generation gain matrices $\boldsymbol{K}_{\text{V}}$ and $\boldsymbol{K}_{\text{P}}$ satisfy

$$\boldsymbol{K}_{\text{V}}^2 > 2\boldsymbol{K}_{\text{P}}. \qquad (6.112)$$

Then for a given $\gamma > 0$, the reference error feedback

$$\boldsymbol{u} = -\boldsymbol{K}\dot{\boldsymbol{e}}_{\text{ref}} = -\boldsymbol{K}\left(\dot{\boldsymbol{e}}_q + \boldsymbol{K}_{\text{V}}\boldsymbol{e}_q + \boldsymbol{K}_{\text{P}}\int \boldsymbol{e}_q\right) \qquad (6.113)$$

satisfies the L_2-gain attenuation requirement for

$$\boldsymbol{Q} = \begin{pmatrix} \boldsymbol{K}_{\text{P}}^2 \boldsymbol{K}_\gamma & 0 & 0 \\ 0 & (\boldsymbol{K}_{\text{V}}^2 - 2\boldsymbol{K}_{\text{P}})\boldsymbol{K}_\gamma & 0 \\ 0 & 0 & \boldsymbol{K}_\gamma \end{pmatrix}, \qquad (6.114)$$

$$\boldsymbol{R} = \boldsymbol{K}^{-1}, \qquad (6.115)$$

provided that

$$\boldsymbol{K}_\gamma = \boldsymbol{K} - \frac{1}{\gamma^2}\boldsymbol{I} > 0. \qquad (6.116)$$

Given γ, one can set $\boldsymbol{K} = \alpha \frac{1}{\gamma^2}\boldsymbol{I}$ for $\alpha > 1$. This yields $\boldsymbol{K}_\gamma = (\alpha - 1)\frac{1}{\gamma^2}\boldsymbol{I}$.

When the inertia matrix is identified as a diagonal constant matrix such as $\widehat{\boldsymbol{H}} = \text{diag}\{\widehat{m}_i\}$, one should set $\widehat{\boldsymbol{C}} = \boldsymbol{0}$. In addition, one can set $\widehat{\boldsymbol{\tau}}_g = \boldsymbol{0}$. Then this results in a decoupled PID control of the form

$$\tau_i = \widehat{m}_i\left(\ddot{q}_{\text{d},i} + k_{\text{V},i}\dot{e}_{\text{q},i} + k_{\text{P},i}e_{\text{q},i}\right)$$
$$+ \alpha\frac{1}{\gamma^2}\left(\dot{e}_{\text{q},i} + k_{\text{V},i}e_{\text{q},i} + k_{\text{P},i}\int e_{\text{q},i}\right)$$

for $\alpha > 1$, which can be rewritten as

$$\tau_i = \widehat{m}_i\ddot{q}_{\text{d},i} + \left(\widehat{m}_i k_{\text{V},i} + \alpha\frac{1}{\gamma^2}\right)\dot{e}_{\text{q},i}$$
$$+ \left(\widehat{m}_i k_{\text{P},i} + \alpha\frac{k_{\text{V},i}}{\gamma^2}\right)e_{\text{q},i} + \alpha\frac{k_{\text{P},i}}{\gamma^2}\int e_{\text{q},i}. \qquad (6.117)$$

This leads to a PID control with the desired acceleration feedforward [6.67] given by

$$\tau_i = \widehat{m}_i\ddot{q}_{\text{d},i} + k_{\text{V},i}^*\dot{e}_{\text{q},i} + k_{\text{P},i}^*e_{\text{q},i} + k_{\text{I},i}^*\int e_{\text{q},i}, \qquad (6.118a)$$

where

$$k_{\text{V},i}^* = \widehat{m}_i k_{\text{V},i} + \alpha\frac{1}{\gamma^2}, \qquad (6.118b)$$

$$k_{\text{P},i}^* = \widehat{m}_i k_{\text{P},i} + \alpha\frac{k_{\text{V},i}}{\gamma^2}, \qquad (6.118c)$$

$$k_{\text{I},i}^* = \alpha\frac{k_{\text{P},i}}{\gamma^2}. \qquad (6.118d)$$

6.9 Digital Implementation

Most controllers introduced in the previous sections are digitally implemented on microprocessors. In this section basic but essential practical issues related to their computer implementation are discussed. When the controller is implemented on a computer control system, the analog inputs are read and the outputs are set with a certain sampling period. This is a drawback compared to analog implementations, since sampling introduces time delays into the control loop. Figure 6.8 shows the overall block diagram of control system with a boxed digital implementation part. When a digital computer is used to implement a control law, it is convenient to divide coding sequence in the interrupt routine into four process routines, as shown in Fig. 6.9. Reading the input signal from sensors and writing the control signal to digital-to-analog (D/A) converters synchronized at the correct frequency is very important. Therefore, these processes are located in the first routine. After saving counter values and extracting D/A values, which are already calculated one step before, the next routine produces reference values. Control routines with filters follow and produce scalar or vector control outputs. Finally, the user interface for checking parameter values is made and will be used for tuning and debugging.

6.9.1 Reference Trajectory Generation

Reference trajectory generation is a starting point and is used as a target value of control. It is a set of values referred to the set point in every interrupt of the control processor and is therefore called the motion profile. Proper design of this profile is important because tracking and settling errors are very sensitive to it, since the target of the practical servo control is not regulating but tracking. A general trapezoidal shape profile is simple to calculate and constant current is required during acceleration and deceleration periods. However, large jerks occur at both ends of the motion, resulting in vibration. Large jerks (rapid change in acceleration, and therefore rapid changes in force) could cause substantial damage to the dynamic systems and induce unwanted vibrations, therefore a smoother profile is required, although this needs more calculation time. A simple motion profile using a third-order polynomial is as follows.

$$y(t) = a + bt + ct^2 + dt^3 \ ,$$
$$\dot{y}(t) = b + 2ct + 3dt^2 \ ,$$
$$\ddot{y}(t) = 2c + 6dt \ .$$

In practice, the motion profile equations are normalized and programmed as functions. With the constraints

$$y(0) = 0 \ , \quad y(1) = 1 \ ,$$
$$\dot{y}(0) = 0 \ , \quad \dot{y}(1) = 0 \ ,$$

the program code takes the form shown in (6.119) when moving time and distance are normalized.

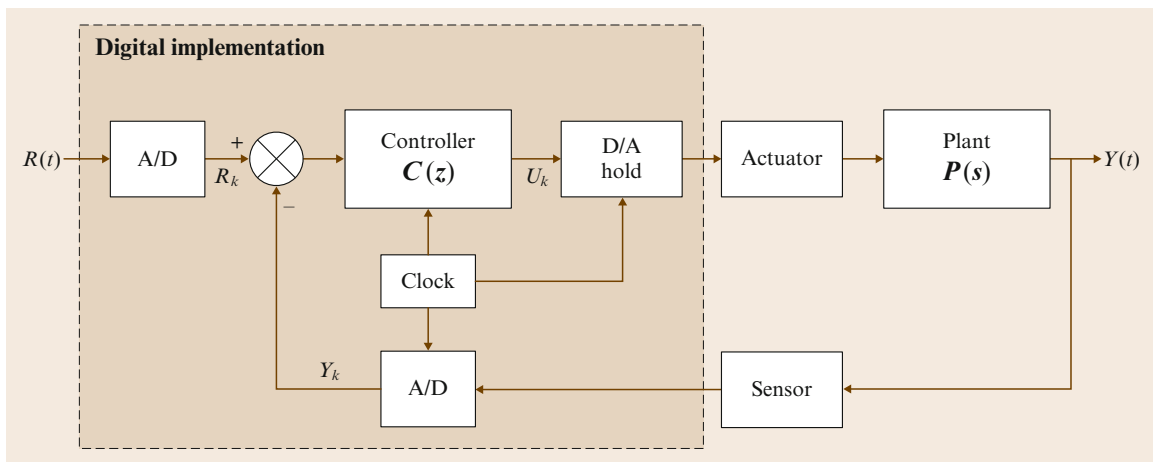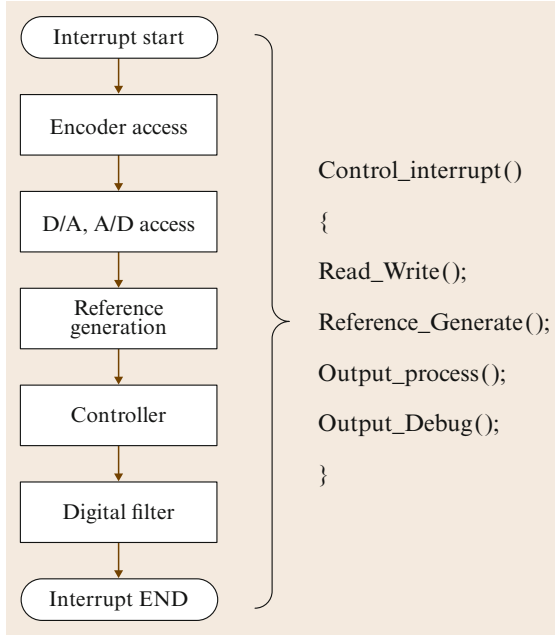**Fig. 6.8** Digital implementation of system control

Fig. 6.9 The sequence in the interrupt routine for digital control

$$y\left(\frac{t}{T}\right) = \left[3\left(\frac{t}{T}\right)^2 - 2\left(\frac{t}{T}\right)^3\right]S,$$

$$\dot{y}\left(\frac{t}{T}\right) = \left[\frac{6}{T}\left(\frac{t}{T}\right) - \frac{6}{T}\left(\frac{t}{T}\right)^2\right]S,$$

$$\ddot{y}\left(\frac{t}{T}\right) = \left[\frac{6}{T^2} - \frac{12}{T^2}\left(\frac{t}{T}\right)\right]S,$$

$$\dddot{y}\left(\frac{t}{T}\right) = -\frac{12}{T^3}S,$$

$$\tag{6.119}$$

where S is the distance and T is the time for motion.

For high-speed servo systems, the jerk profile is constant during runtime with large changes at start and stop. This means that high mechanical stiffness is required. Even though the calculation load increases, either fifth- or seventh-order polynomial profiles can be designed in order to satisfy acceleration or jerk constraints.

The motion profile is often saved in read-only memory (ROM) in the form of a table or located in the interrupt routine *Reference_Generate()* in the form of a function, as shown in Fig. 6.9. In most tracking servo control problem, a position profile is generated. However, velocity and acceleration are also calculated in the interrupt routine in real time to be used as inputs for the feedforward control. These are able to improve the

tracking performance, and the algorithm is as follows:

$$Pos_{\text{ref}} = PosTable(\text{pointer}) \times Distance\,;$$

$$Vel_{\text{ref}} = Pos_{\text{ref}} - Pos_{\text{ref,OLD}}$$

$$Acc_{\text{ref}} = Vel_{\text{ref}} - Vel_{\text{ref,OLD}}$$

$$\vdots \tag{6.120}$$

where Vel_{ref} and Acc_{ref} do not mean the real reference velocity and acceleration, but the reference velocity scaled by the interrupt time interval and the acceleration scaled by the square of the interrupt time interval, respectively, for practical use. Also, the limited size of the registers of a digital signal processor (DSP) or processor can cause errors due to the loss of significant digits when the velocity profile is calculated from the position profile using the simple difference formula in (6.120). This means that the velocity profile includes a periodic noise signal. This kind of numerical error creates real noise in the motion and reduces the performance of the control system when profiles of velocity, acceleration, and jerk are used as inputs for the feedforward control system for high-speed tracking.

Instead of the simple difference in (6.120), numerical differential equations can be used. Equation (6.121) shows the first-order backward-difference equations, which are the simplest form,

$$\dot{y}_k = \frac{y_k - y_{k-1}}{\Delta t}\,,$$

$$\ddot{y}_k = \frac{y_k - 2y_{k-1} + y_{k-2}}{\Delta t^2}\,,$$

$$\dddot{y}_k = \frac{y_k - 3y_{k-1} + 3y_{k-2} - y_{k-3}}{\Delta t^3}\,. \tag{6.121}$$

Saving each profiles in the form of a table can be another good method if the memory is large enough.

The important thing is that the maximum specification for the movement of controlled system, such as speed or acceleration or even jerk, should be considered in the profile design step. If the profile forces the system to breach the maximum specification, the possibility for the system to be driven recklessly is greatly increased due to missing of encoder pulses.

6.9.2 Z–Transform for Coding

Continuous-time systems are transformed into discrete-time systems by using the Z-transform. A discrete-time system is used to obtain a mathematical model that gives the behavior of a physical process at the sampling points, although the physical process is still a continuous-time

system. A Laplace transform is used for the analysis of control system in the s-domain. In most cases, the design of controllers and filters are done using tools in the s-domain. In order to realize those results in program code, understanding the Z-transform is essential. All controllers and filters designed in the s-domain can be easily translated to a program code through a Z-transform because it has the form of digitized difference sequences.

A coding PID controller is used as an example. In transfer-function form, this controller has the basic structure

$$\frac{Y(s)}{E(s)} = K_P + \frac{K_I}{s} + sK_V \, . \tag{6.122}$$

There are several methods for transformation from the frequency to the discrete domain. For stability conservation, backward Euler and Tustin algorithms are often used. Though the Tustin algorithm is known as the more exact one, the backward Euler algorithm is utilized in the following procedure.

After substituting the backward Euler equation into (6.122),

$$s \cong \frac{z-1}{zT}$$

the following discrete form is produced

$$\frac{Y(z)}{E(z)} = \frac{\alpha + \beta z^{-1} + \gamma z^{-2}}{T(1-z^{-1})} \, , \tag{6.123}$$

where

$$\alpha = K_I T^2 + K_P T + K_V \, ,$$
$$\beta = -K_P T - 2K_V \, ,$$
$$\gamma = K_V \, .$$

Now, this is rearranged using the difference equation:

$$T(y_k - y_{k-1}) = \alpha e_k + \beta e_{k-1} + \gamma e_{k-2} \, ,$$
$$y_k = y_{k-1} + \frac{1}{T}(\alpha e_k + \beta e_{k-1} + \gamma e_{k-2}) \, ,$$
$$y_k - y_{k-1} = +\frac{\alpha}{T}e_k + \frac{\beta}{T}e_{k-1} + \frac{\gamma}{T}e_{k-2} \, . \tag{6.124}$$

This can be directly coded in the program as follows:

$$e_k = p_{k,\text{ref}} - p_k \, ,$$
$$e_k^v = v_{k,\text{ref}} - v_k$$
$$= (p_{k,\text{ref}} - p_{k-1,\text{ref}}) - (p_k - p_{k-1})$$
$$= e_k - e_{k-1} \, ,$$
$$\text{sum}_k = \text{sum}_{k-1} + e_k = \sum_{j=0}^{k} e_j \, ,$$
$$y_k = K_{P,c}e_k + K_{V,c}(e_k - e_{k-1}) + K_{I,c}\text{sum}_k \, ,$$
$$y_k - y_{k-1}$$
$$= (K_{P,c}e_k - K_{P,c}e_{k-1})$$
$$+ [K_{V,c}(e_k - e_{k-1})] - [K_{V,c}(e_{k-1} - e_{k-2})]$$
$$+ (K_{I,c}\text{sum}_k - K_{I,c}\text{sum}_{k-1})$$
$$= (K_{P,c} + K_{V,c} + K_{I,c})e_k$$
$$- (K_{P,c} + 2K_{V,c})e_{k-1} + K_{V,c}e_{k-2} \, , \tag{6.125}$$

where p_k is the present position, v_k is the present velocity, ref means reference, and c means code.

Comparing the parameters in (6.124) and (6.125), one obtains

$$\frac{\alpha}{T} = K_P + \frac{K_V}{T} + K_I T = (K_{P,c} + K_{V,c} + K_{I,c}) \, ,$$
$$\frac{\beta}{T} = -K_P - \frac{2K_V}{T} = -(K_{P,c} + 2K_{V,c}) \, ,$$
$$\frac{\gamma}{T} = \frac{K_V}{T} = K_{V,c} \, ,$$

which shows that there is a relation between the designed and coded forms of the gains:

$$K_{P,c} = K_P \, ,$$
$$K_{V,c} = \frac{K_V}{T} \, ,$$
$$K_{I,c} = K_I T \, . \tag{6.126}$$

As the sampling frequency is increased in the same system, the coded K_V gain should be increased and the coded K_I gain should be decreased. Using this method, the designed controller can be coded in a DSP or microprocessor. However, sufficient analysis and simulation for control algorithms should be performed beforehand to obtain successful control system performance.

6.10 Learning Control

Since many robotic applications, such as pick-and-place operations, paintings, and circuit-board assembly, involve repetitive motions, it is natural to consider the use of data gathered in previous cycles to try to improve the performance of the manipulator in subsequent cycles. This is the basic idea of repetitive control or learn-

ing control. Consider the robot model given in Sect. 6.1 and suppose that one is given a desired joint trajectory $q_d(t)$ on a finite time interval $0 \leq t \leq T$. The reference trajectory q_d is used in repeated trails of the manipulator, assuming either that the trajectory is periodic, $q_d(T) = q_d(0)$, (repetitive control) or that the robot is reinitialized to lie on the desired trajectory at the beginning of each trail (learning control). Hereafter, we use the term learning control to mean either repetitive or learning control.

6.10.1 Pure P–type Learning Control

Let τ_k be the input torque during the k-th cycle, which produces an output $q_k(t)$, $0 \leq t \leq T_{bnd}$. Now, let us consider the following set of assumptions:

- Assumption 1: Every trial ends at a fixed time of duration $T_{bnd} > 0$.
- Assumption 2: Repetition of the initial setting is satisfied.
- Assumption 3: Invariance of the system dynamics is ensured throughout these repeated trails.
- Assumption 4: Every output q_k can be measured and thus the error signal $\Delta q_k = q_k - q_d$ can be utilized in the construction of the next input τ_{k+1}.
- Assumption 5: The dynamics of the robot manipulators is invertible.

The learning control problem is to determine a recursive learning law L

$$\tau_{k+1} = L[\tau_k(t), \Delta q_k(t)], \quad 0 \leq t \leq T_{bnd}, \quad (6.127)$$

where $\Delta q_k(t) = q_k(t) - q_d(t)$, such that $\|\Delta q_k\| \to 0$ as $k \to \infty$ in some suitably defined function norm, $\|\cdot\|$. The initial control input can be any control input that produces a stable output, such as PD control. Such learning control schemes are attractive because accurate models of the dynamics need not be known a priori.

Several approaches have been used to generate a suitable learning law L and to prove convergence of the output error. A pure P-type learning law is one of the form

$$\tau_{k+1}(t) = \tau_k(t) - \boldsymbol{\Phi} \Delta q_k(t), \quad (6.128)$$

and is given this name because the correction term for the input torque at each iteration is proportional to the error Δq_k. Now let τ_d be defined by the computed-torque control, i. e.,

$$\tau_d(t) = \boldsymbol{H}[q_d(t)]\ddot{q}_d(t) + \boldsymbol{C}[q_d(t), \dot{q}_d(t)]\dot{q}_d(t)$$
$$+ \tau_g[q_d(t)]. \quad (6.129)$$

One should recall that the function τ_k actually does not need to be computed; it is sufficient to know that it exists. Considering the P-type learning control law, we have

$$\Delta \tau_{k+1}(t) = \Delta \tau_k(t) - \boldsymbol{\Phi} \Delta q_k(t), \quad (6.130)$$

where $\Delta \tau_k(t) = \tau_k(t) - \tau_d(t)$, so that

$$\|\Delta \tau_{k+1}(t)\|^2 \leq \|\Delta \tau_k(t)\|^2 - \beta\|\boldsymbol{\Phi} \Delta q_k(t)\|^2 \quad (6.131)$$

provided there exist positive constant λ and β such that

$$\int_0^{T_{bnd}} e^{-\lambda t} \Delta q_k^\top \Delta \tau_k(t) \, dt \geq \frac{1+\beta}{2} \|\boldsymbol{\Phi} \Delta q_k(t)\|^2$$

$$(6.132)$$

for all k. It then follows from the inequality above that $\Delta q_k \to 0$ in the norm sense as $k \to \infty$. Detailed stability analysis of this control scheme is given in [6.68, 69].

6.10.2 P–Type Learning Control with a Forgetting Factor

Although pure P-type learning control achieves the desired goal, several strict assumptions may be not valid in actual implementations, for example, there may be an initial setting error. Furthermore, there may be small but nonrepeatable fluctuations of dynamics. Finally, there may exit a (bounded) measurement noise $\boldsymbol{\xi}_k$ such that

$$\Delta q_k(t) + \boldsymbol{\xi}_k(t) = [q_k(t) + \boldsymbol{\xi}_k(t)] - q_d(t). \quad (6.133)$$

Thus the learning control scheme may fail. In order to enhance the robustness of P-type learning control, a forgetting factor is introduced in the form

$$\tau_{k+1}(t) = (1 - \alpha)\tau_k(t) + \alpha\tau_0(t)$$
$$- \boldsymbol{\Phi}[\Delta q_k(t) + \boldsymbol{\xi}_k(t)]. \quad (6.134)$$

The original idea of using a forgetting factor in learning control originated with [6.70].

It has been rigorously proven that P-type learning control with a forgetting factor guarantees convergence to a neighborhood of the desired one of size $O(\alpha)$. Moreover, if the content of a long-term memory is refreshed after every k trials, where k is of $O(1/\alpha)$, then the trajectories converge to an ε-neighborhood of the desired control goal. The size of ε is dependent on the magnitude of the initial setting error, the nonrepeatable fluctuations of the dynamics, and the measurement noise. For a detailed stability investigation, please refer to [6.71, 72].

6.10.3 Summary

By applying learning control, the performance of repetitive tasks (such as painting or pick-and-place operation) is improved by utilizing data gathered in the previous cycles. In this section, two learning control schemes were introduced. First, pure P-type learning control and its robustness problem were described. Then P-type learning control with a forgetting factor was presented, enhancing the robustness of learning control.

Further Reading

Rigorous and refined exploration of learning control is first discussed independently in [6.2, 12].

References

6.1 C. Canudas de Wit, B. Siciliano, G. Bastin: *Theory of Robot Control* (Springer, London 1996)

6.2 J.J. Craig: Adaptive Control of Mechanical Manipulators. Ph.D. Thesis (UMI Dissertation Information Service, Ann Arbor 1986)

6.3 R.J. Schilling: *Fundametals of Robotics: Analysis and Control* (Prentice-Hall, Upper Saddle River 1989)

6.4 L. Sciavicco, B. Siciliano: *Modeling and Control of Robot Manipulator* (McGraw-Hill, New York 1996)

6.5 M.W. Spong, M. Vidyasagar: *Robot Dynamics and Control* (Wiley, New York 1989)

6.6 M.W. Spong, F.L. Lewis, C.T. Abdallah (Eds.): *Robot Control* (IEEE, New York 1989)

6.7 C.H. An, C.G. Atkeson, J.M. Hollerbach: *Model-Based Control of a Robot Manipulator* (MIT Press, Cambridge, 1988)

6.8 R.M. Murray, Z. Xi, S.S. Sastry: *A Mathematical Introduction to Robotic Manipulation* (CRC Press, Boca Raton 1994)

6.9 T. Yoshikawa: *Foundations of Robotics* (MIT Press, Cambridge 1990)

6.10 O. Khatib: A unified approach for motion and force control of robot manipulators: The operational space formulation, IEEE J. Robot. Autom. **3**(1), 43–53 (1987)

6.11 J.Y.S. Luh, M.W. Walker, R.P.C. Paul: Resolved-acceleration control of mechanical manipulator, IEEE Trans. Autom. Contr. **25**(3), 468–474 (1980)

6.12 S. Arimoto, F. Miyazaki: Stability and robustness of PID feedback control for robot manipulators of sensory capability. In: *Robotics Research*, ed. by M. Brady, R. Paul (MIT Press, Cambridge 1984) pp. 783–799

6.13 L.C. Fu: Robust adaptive decentralized control of robot manipulators, IEEE Trans. Autom. Contr. **37**(1), 106–110 (1992)

6.14 H. Seraji: Decentralized adaptive control of manipulators: Theory, simulation, and experimentation, IEEE Trans. Robot. Autom. **5**(2), 183–201 (1989)

6.15 J.G. Ziegler, N.B. Nichols: Optimum settings for automatic controllers, ASME Trans. **64**, 759–768 (1942)

6.16 Y. Choi, W.K. Chung: *PID Trajectory Tracking Control for Mechanical Systems*, Lecture Notes in Control and Information Sciences, Vol. 289 (Springer, New York 2004)

6.17 R. Kelly: PD control with desired gravity compensation of robot manipulators: A review, Int. J. Robot. Res. **16**(5), 660–672 (1997)

6.18 M. Takegaki, S. Arimoto: A new feedback method for dynamic control of manipulators, Trans. ASME J. Dyn. Syst. Meas. Contr. **102**, 119–125 (1981)

6.19 P. Tomei: Adaptive PD controller for robot manipulators, IEEE Trans. Robot. Autom. **7**(4), 565–570 (1991)

6.20 R. Ortega, A. Loria, R. Kelly: A semi-globally stable output feedback PI^2 D regulator for robot manipulators, IEEE Trans. Autom. Contr. **40**(8), 1432–1436 (1995)

6.21 D. Angeli: Input-to-State stability of PD-controlled robotic systems, Automatica **35**, 1285–1290 (1999)

6.22 J.A. Ramirez, I. Cervantes, R. Kelly: PID regulation of robot manipulators: stability and performance, Sys. Contr. Lett. **41**, 73–83 (2000)

6.23 R. Kelly: Global positioning of robot manipulators via PD control plus a class of nonlinear integral actions, IEEE Trans. Autom. Contr. **43**(7), 934–937 (1998)

6.24 Z. Qu, J. Dorsey: Robust tracking control of robots by a linear feedback law, IEEE Trans. Autom. Contr. **36**(9), 1081–1084 (1991)

6.25 H. Berghuis, H. Nijmeijer: Robust control of robots via linear estimated state feedback, IEEE Trans. Autom. Contr. **39**(10), 2159–2162 (1994)

6.26 Y. Choi, W.K. Chung, I.H. Suh: Performance and $\mathcal{H}_\infty$ optimality of PID trajectory tracking controller for Lagrangian systems, IEEE Trans. Robot. Autom. **17**(6), 857–869 (2001)

6.27 K. Aström, T. Hagglund: *PID Controllers: Theory, Design, and Tuning* (Instrument Society of America, Research Triangle Park 1995)

6.28 C.C. Yu: *Autotuning of PID Controllers: Relay Feedback Approach* (Springer, London 1999)

6.29 F.L. Lewis, C.T. Abdallah, D.M. Dawson: *Control of Robot Manipulators* (Macmillan, New York 1993)

6.30 A. Isidori: *Nonlinear Control Systems: An Introduction*, Lecture Notes in Control and Information Sciences, Vol. 72 (Springer, New York 1985)

6.31 H. Berghuis, H. Nijmeijer: A passivity approach to controller-observer design for robots, IEEE Trans. Robot. Autom. **9**, 740–754 (1993)

Part A | 6

6.32   J.J. Slotine, W. Li: On the adaptive control of robot manipulators, Int. J. Robot. Res. **6**(3), 49–59 (1987)

6.33   G. Liu, A.A. Goldenberg: Comparative study of robust saturation–based control of robot manipulators: analysis and experiments, Int. J. Robot. Res. **15**(5), 473–491 (1996)

6.34   D.M. Dawson, M. Grabbe, F.L. Lewis: Optimal control of a modified computed–torque controller for a robot manipulator, Int. J. Robot. Autom. **6**(3), 161–165 (1991)

6.35   D.M. Dawson, Z. Qu, J. Duffie: Robust tracking control for robot manipulators: theory, simulation and implementation, Robotica **11**, 201–208 (1993)

6.36   A. Jaritz, M.W. Spong: An experimental comparison of robust control algorithms on a direct drive manipulator, IEEE Trans. Contr. Syst. Technol. **4**(6), 627–640 (1996)

6.37   A. Isidori: *Nonlinear Control Systems*, 3rd edn. (Springer, New York 1995)

6.38   J.J. Slotine, W. Li: *Applied Nonlinear Control* (Prentice–Hall, Englewood Cliffs 1991)

6.39   W.J. Rugh: *Linear System Theory*, 2nd edn. (Prentice-Hall, Upper Saddle River 1996)

6.40   M.W. Spong, M. Vidyasagar: Robust microprocessor control of robot manipulators, Automatica **23**(3), 373–379 (1987)

6.41   H.K. Khalil: *Nonlinear Systems*, 3rd edn. (Prentice-Hall, Upper Saddle River 2002)

6.42   M. Vidysagar: *Nonlinear Systems Analysis*, 2nd edn. (Prentice–Hall, Englewood Ciffs 1993)

6.43   J.T. Wen: A unified perspective on robot control: The energy Lyapunov function approach, Int. J. Adapt. Contr. Signal Proc. **4**, 487–500 (1990)

6.44   R. Ortega, M.W. Spong: Adaptive motion control of rigid robots: A tutorial, Automatica **25**(6), 877–888 (1989)

6.45   N. Sadegh, R. Horowitz: Stability and robustness analysis of a class of adaptive contollers for robotic manipulators, Int. J. Robot. Res. **9**(3), 74–92 (1990)

6.46   S. Dubowsky, D.T. DesForges: The application of model–reference adaptive control to robotic manipulators, ASME J. Dyn. Syst. Meas. Control. **37**(1), 106–110 (1992)

6.47   A. Balestrino, G. de Maria, L. Sciavicco: An adaptive model following control for robotic manipulators, ASME J. Dyn. Syst. Meas. Control. **105**, 143–151 (1983)

6.48   S. Nicosia, P. Tomei: Model reference adaptive control algorithms for industrial robots, Automatica **20**, 635–644 (1984)

6.49   R. Horowitz, M. Tomizuka: An adaptive control scheme for mechanical manipulators-Compensation of nonlinearity and decoupling control, ASME J. Dyn. Syst. Meas. Control. **108**, 127–135 (1986)

6.50   I.D. Laudau: *Adaptive Control: The Model Reference Approach* (Dekker, New York 1979)

6.51   R. Lozano, C. Canudas de Wit: Passivity based adaptive control for mechanical manipulators using LS type estimation, IEEE Trans. Autom. Contr. **35**(12), 1363–1365 (1990)

6.52   B. Brogliato, I.D. Laudau, R. Lozano: Passive least squares type estimation algorithm for direct adaptive control, Int. J. Adapt. Contr. Signal Process. **6**, 35–44 (1992)

6.53   R. Johansson: Adaptive control of robot manipulator motion, IEEE Trans. Robot. Autom. **6**(4), 483–490 (1990)

6.54   M.W. Walker: Adaptive control of manipulators containing closed kinematic loops, IEEE Trans. Robot. Autom. **6**(1), 10–19 (1990)

6.55   J.S. Reed, P.A. Ioannou: Instability analysis and robust adaptive control of robotic manipulators, IEEE Trans. Autom. Contr. **5**(3), 74–92 (1989)

6.56   G. Tao: On robust adaptive control of robot manipulators, Automatica **28**(4), 803–807 (1992)

6.57   H. Berghuis, R. Ogata, H. Nijmeijer: A robust adaptive controller for robot manipulators, Proc. IEEE Int. Conf. Robot. Autom. (1992) pp. 1876–1881

6.58   R. Johansson: Quadratic optimization of motion coordination and control, IEEE Trans. Autom. Contr. **35**(11), 1197–1208 (1990)

6.59   Z. Qu, D.M. Dawson: *Robust Tracking Control of Robot Manipulators* (IEEE, Piscataway 1996)

6.60   P. Dorato, C. Abdallah, V. Cerone: *Linear-Quadratic Control* (Prentice-Hall, Upper Saddle River 1995)

6.61   A. Locatelli: *Optimal Control: An Introduction* (Birkhäuser, Basel 2001)

6.62   A. Isidori: Feedback control of nonlinear systems, Int. J. Robust Nonlin. Contr. **2**, 291–311 (1992)

6.63   A.J. Van Der Schaft: Nonlinear state space $\mathcal{H}_\infty$ control theory. In: *Essays on Control: Perspective in Theory and its Applications*, ed. by H.L. Trentelman, J.C. Willems (Birkhäuser, Basel 1993) pp. 153–190

6.64   A.J. Van Der Schaft: $L_2 9$-gain analysis of nonlinear systems and nonlinear state feedback $\mathcal{H}_\infty$ control, IEEE Trans. Autom. Contr. **37**(6), 770–784 (1992)

6.65   J. Park, W.K. Chung, Y. Youm: Analytic nonlinear $\mathcal{H}_\infty$ inverse-optimal control for Euler-Lagrange system, IEEE Trans. Robot. Autom. **16**(6), 847–854 (2000)

6.66   B.S. Chen, T.S. Lee, J.H. Feng: A nonlinear $\mathcal{H}_\infty$ control design in robotics systems under parametric perturbation and external disturbance, Int. J. Contr. **59**(12), 439–461 (1994)

6.67   J. Park, W.K. Chung: Design of a robust $\mathcal{H}_\infty$ PID control for industrial manipulators, ASME J. Dyn. Syst. Meas. Contr. **122**(4), 803–812 (2000)

6.68   S. Arimoto: Mathematical theory or learning with application to robot control. In: *Adaptive and Learning Control*, ed. by K.S. Narendra (Plenum, New York 1986) pp. 379–388

6.69   S. Kawamura, F. Miyazaki, S. Arimoto: Realization of robot motion based on a learning method, IEEE Trans. Syst. Man. Cybern. **18**(1), 126–134 (1988)

6.70   G. Heinzinger, D. Frewick, B. Paden, F. Miyazaki: Robust learning control, Proc. IEEE Int. Conf. Dec. Contr. (1989)

6.71 S. Arimoto: Robustness of learning control for robot manipulators, Proc. IEEE Int. Conf. Dec. Contr. (Bellingham 1990) pp. 1523–1528

6.72 S. Arimoto, T. Naiwa, H. Suzuki: Selective learning with a forgetting factor for robotic motion control, Proc. IEEE Int. Conf. Dec. Contr. (Bellingham 1991) pp. 728–733