22

Control of Robots

Miomir Vukobratović
Mihajlo Pupin Institute

Dragan Stokić
ATB Institute

22.1 Introduction

This chapter is dedicated to the synthesis of basic control of manipulation robots. Because the successful application of robots in industry and other domains often depends to a great extent upon the efficiency, reliability, and capabilities of a control system, it is obvious that the synthesis of adequate control systems is of the highest importance for further application of robotics in industrial practice.

Control systems of robots can be realized in different ways. Historically speaking, different open-loop control systems were applied to control the first manipulation robots. However, current robots include digital (microprocessor)-based control systems that enable flexible specification of the tasks, adaptation to environment changes, etc. A robot's joints are controlled by servo-systems (or servos) based on the feedback loops providing information on positions, velocity, and accelerations of the joints.

In this chapter we mainly focus upon the synthesis of servos for robots. In order to enable efficient specification of the tasks to be fulfilled by the robot, modern control systems include options to specify directly desired the position of the gripper (hand). To accomplish various tasks, the hand of the robot (or the payload, or the tool) has to be placed in the desired locations at the workplace and take the desired orientation (and, sometimes, to produce certain desired forces upon the other objects in the workspace). If an operator, when specifying the task for the robot to accomplish, intends to place the hand of the robot in a desired position by specifying the positions of the joints, he or she would have to determine the corresponding positions of the joints in an iterative way. For some robot structures this may be easy task (e.g., a robot with three linear joints, or a robot with a cylindrical structure, etc.). However, for the majority of robot structures, this can be a very tedious and time-consuming job. Therefore, it is necessary to enable the user to directly specify the desired positions of the robot hand, either by programming the robot, or by a teaching-box, or by some other means. In this case, the operator of the robot has to specify the desired position and orientation of the hand, and the control system has to compute automatically the corresponding positions of the joints. This means that the control system has to compute internal (joint) coordinates of the robot based on the desired position/orientation of the robot hand, i.e., based on the specified so-called external (or Cartesian) coordinates. This calculation can be performed by the control computer in various ways. Most modern robots are equipped with control systems that enable direct specification of the Cartesian coordinates.

Modern industry and other application domains are assigning more and more complex tasks to robots. Apart from the simplest task (such as pick-and-place, which can be reduced to a free motion of the robot and payload from one position to another), modern robots have to ensure movements along prespecified paths in the workspace (for example, arc-welding by robots, gluing by robots, moving a robot in a workspace with many obstacles, etc.). In these tasks, the operator has to specify the desired path of the robot's hand and the control system has to calculate the corresponding trajectories of the robot's joints and ensure their execution (i.e., the robot's joints are tracking these trajectories which, in turn, should ensure that the hand is following the desired path in the workspace).

Often robot tasks can be complex and the operator may need a very long time to specify the positions through which the robot has to move, or the paths along which the robot hand has to move to perform the desired tasks. For example, if the robot has to move very close to various machines and equipment in its workspace (i.e., if the robot has to move close to various obstacles), the operator has to plan all the intermediate positions through which the robot has to pass, or to plan paths along which the robot's hand has to move to avoid collisions of the hand or any of the robot's links with the obstacles. Obviously, such trajectory planning task can be very difficult, which is why it is desirable to have a control system capable of solving such problems automatically, and by this the operator is no longer responsible for path planning tasks. A number of modern robots include such control systems with automatic path planning. The user has to specify the task in relatively high abstract form (e.g., replace an object from one position to another), and the control system then automatically plans all movements of the robot (approaching the object, orientation of the hand, grasping the object, lifting the object, replacement of the object to another location with obstacle avoidance, putting the object into another location etc.). This automatic planning of the robot's paths and tasks represents the main prerequisites to introducing robots to flexible manufacturing systems. Obviously, it is also a prerequisite for further spreading of robots in various nonindustrial applications (e.g., service domains, space applications, etc.). Therefore, control systems of the current and future generations of robots required such capabilities.

22.2 Hierarchical Control of Robots

Control systems which can accommodate the requirements explained above, are obviously very complex. To simplify the synthesis and implementation of the control system, it has to be carefully structured. The usual approach to structure control system is to apply hierarchical architecture in which the robot's control system is organized in several levels, with each control level solving its specific task. One such (simple) hierarchical structure is presented in Figure 22.1. In this structure the control system includes three levels:[1]

1. The strategic control level has to plan the robot's paths. This level receives its tasks from the operator who is communicating with the control system by a programming language (normally each control system has special language enabling easy programming of the robot task). The strategic control level has to plan each motion of the robot. The operator specifies the tasks to be accomplished by the robot, and the strategic control level defines those paths of the robot's hand which have to be realized. If the workspace of the robot is predefined (i.e., if all obstacles are prespecified), the strategic control level can plan the paths in the space without additional information from external systems (e.g., sensors). However, if

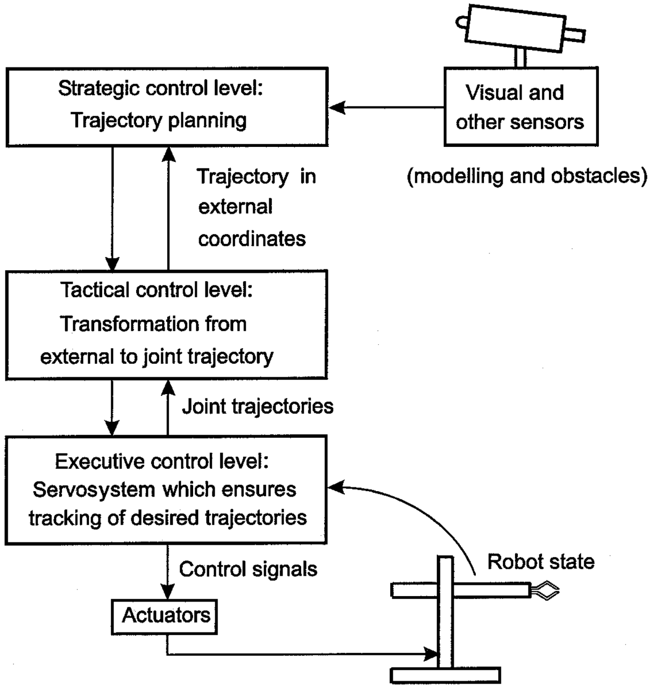

FIGURE 22.1 Simple hierarchical structure of a robot's control system.

locations of different obstacles are not (accurately) defined in advance or may change during the operation of the robot (e.g., movements of the parts not defined in advance), path planning must be performed based upon the sensor information (e.g., cameras, proximity sensor, etc. that provide information on the actual, current positions and shapes of different obstacles). In this case, the strategic level often must solve path-planning problems in real time, i.e., during process execution, which is a much more complex problem than if it can be done off-line (before task execution). In both cases, the strategic control level generates the trajectories of the robot's hand, i.e., it defines trajectories of the external coordinates of the robot.

2. The tactical control level has to map the trajectories from the external into internal (joint) coordinates of the robot. That is, the strategic control level provides the trajectories of the robot's hand coordinates and the tactical control level has to compute the corresponding trajectories of the robot's joints which have to be realized to execute the imposed hand trajectories. This problem is solved using the so-called "inverse kinematic model of the robot." Output of the tactical control level is joint trajectories. This control level can operate in either an off-line or on-line mode, depending on the conditions imposed in the specific tasks.
3. The executive control level has to realize the trajectories (or positions) of the robot's joints which are imposed by the higher, tactical control level. This control level must ensure realization of the imposed trajectories on the basis of information on the actual robot state (positions, speeds, and accelerations of the joints). By ensuring the tracking of the imposed joint trajectories, the trajectories of the robot hand are also accomplished, and the task imposed by the operator is accomplished.

It should be noted that some control systems do not include all three control levels; however, all control levels must include an executive control level to realize desired positions or trajectories of the robot's joints. As explained above, modern robots incorporate specifications of hand coordinates, which

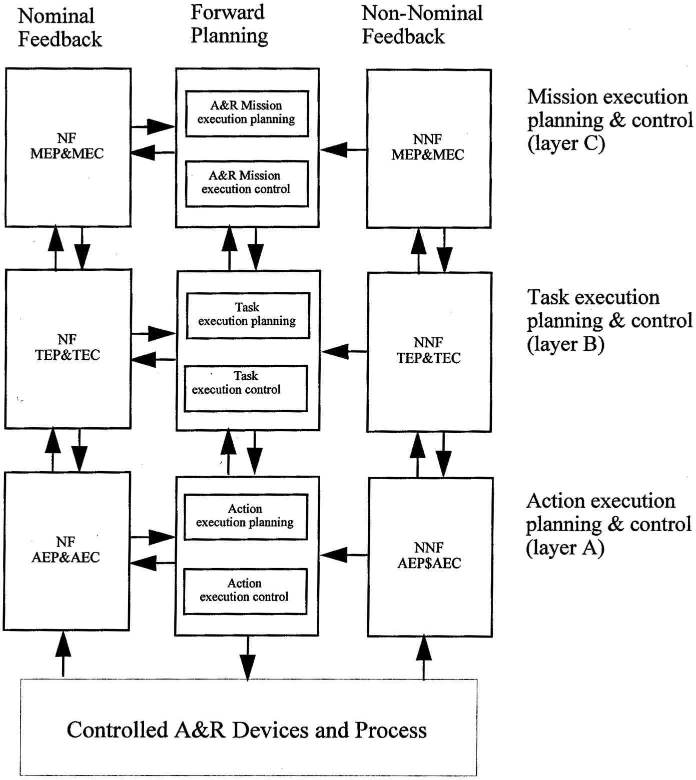

FIGURE 22.2 A&R (automation and robotics) FRM structure. (From Dornier GmbH, 1992.[4] With permission.)

means that they include a tactical control level. However, a number of modern robots still do not include a strategic control level, which means they are not capable of automatically planning hand trajectories. For such robots the operator has to impose the desired trajectories (or positions) of the robot hand, and plan the paths using the robot's programming languages and teaching boxes, etc. Some robots have the strategic control level in a very rudiment form. Given the tasks demanded of modern robots, robots in the near future must include more sophisticated and complex strategic control levels.

The presented control structure is relatively simple. In order to cover the various complex tasks requested by different applications, the control systems must have a much more complex architecture. Different control architectures have been developed for industrial, space, and service applications. Attempts were made to defined general standard structures.[2-4] For example, with the introduction of automation and robot (A&R) technology in space applications, the European Space Agency ESA has identified the need for generic approaches in the development of such systems and has defined the so-called functional reference model (FRM) which provides a unified representation of essential robot control functions.[4] This reference model offers an essential functional and information architecture (a logical model) of general robot control systems that is independent of particular applications, operational scenarios, and implementations.

FRM is presented in Figure 22.2. It includes three main levels (or layers) and three main paths: the forward path where control actions are planned and executed, the nominal feedback path (NF) which establishes the feedback loop from the sensors to correct planned actions based on the current state of the robot and its environment under nominal conditions, and the non-nominal feedback path (NNF) which ensures an appropriate reaction of the robot in non-nominal situations, i.e., when some exceptional, accidental, and unforeseen situations appear (e.g., an actuator or a sensor failure, etc.)

22.2.1 Mission Layer

This layer covers the overall system planning functions. Its functions perform time-lining (planning), scheduling, and dispatching tasks to A&R devices.

The inputs for this layer are the missions from an external system. Missions are process-oriented instructions which do not explicitly specify the way the mission should be accomplished by the robot (e.g., "assemble the parts" etc.).

The output is a set of tasks which specifies how the input mission will be executed by an A&R device such as a manipulation robot or a mobile vehicle.

22.2.2 Task Layer

The basic function of the task layer is to transform the process-oriented input tasks into device-oriented actions of robot. To do this, the most important function for a controller is the planning of trajectories to reach the given locations. The task layer performs path-planning activities (and adjusts these on the basis of sensor readings) to produce executable path segments specified by mathematical curves (straight lines, polynomials, clothoids, etc.).

Typically, this layer contains the following modules: path planning, path control, object recognition, subactivity planning (scheduling), and subactivity control (dispatching).

Input is a set of tasks that describes locations to reach and activities to perform in each location. Output is the path to be followed by the robot and activities to be performed in different parts of the path.

22.2.3 Action Layer

The action layer serves to transform the device-oriented action instructions from the task layer into control commands to the actuator and sensor hardware. The transformation requires the transition from Cartesian space in which the input is specified, into the configuration space of the actuators and sensors. Controlling the robot at this level also includes reactions to obstacles and avoidance of collisions. At this layer this is performed locally, which means with respect to the configuration space of actuators and sensors.

Typical modules for this layer are trajectory interpolation, trajectory control, actuator path interpolation, actuator control (servo control), local position estimation, obstacle detection and avoidance (local), detection of failure to reach goal, elementary activity planning, and elementary activity control.

Input is the path to be followed by the robot's hand, and the actions to be performed in different parts of the path. Outputs are control output signals to the actuator and sensor hardware.

In this chapter we focus on the problems related to the synthesis of the executive control level (Figure 22.1), i.e., the actuator control module in FRM. This means that we consider control of the actuators that drive the joints of the robot to maintain positions and trajectories imposed either by a higher (tactical) control level, or directly by the operator. In doing this we observe both problems: if the robot moves point-to-point (from one position to another), and if it has to move along desired continuous trajectories. It should be mentioned that the synthesis of the executive control level considered is relevant for all generations of robots and for both remote and manual robot control. We present some of the simplest approaches for robot control synthesis, those most often applied in practice. More sophisticated methods may be found in the corresponding literature.

22.3 Control of a Single Joint of the Robot

First we consider a simple case when a single joint of the robot is moving while all other joints are fixed. Let us assume the i-th joint of the robot has to be moved. The joints of the robot are driven by the actuators, and therefore, we consider the synthesis of control of an actuator driving the i-th joint while all other joints are fixed.

22.3.1 Model of Actuator and Joint Dynamics

The actuators driving the joints may be D.C. or A.C. electromotors, hydraulic or pneumatic actuators. Because a large number of robots are driven by D.C. electromotors, we consider synthesis of the control for such actuators. However, these considerations can be easily extended to other types of actuators.[5]

The model of the dynamics of a D.C. electromotor, with a permanent magnet driving the i-th joint can be written in the following form.[5,6] The equation of moments equilibrium around the motor axis can be written as:

$$N_M^i N_V^i J_M^i \ddot{q}_i + P_i = N_M^i C_M^i i_r^i - B_C^i \dot{q}_i \quad (22.1)$$

where J_M^i is the moment of inertia of the rotor of the motor, q_i is the angle of rotation of the motor (joint) P_i is the load acting around the motor axis, C_M^i is the mechanical constant of the motor (the coefficient of proportionality between the moments developed by the motor and the current of the rotor coil), i_M^i is the current in the rotor coil, B_C^i is the coefficient of the viscous friction of the motor N_M^i is the moment reduction ratio at the motor axis (the ratio between the moment behind and in front of the gear), N_V^i is the speed reduction ratio of the gear (the ratio between the speed of the input and output shafts of the gear). The equation describing the equilibrium in the electric circuit of the rotor coil can be written in the form (assuming that the inductivity of the coil can be ignored):

$$R_r^i i_r^i + C_e^i N_V^i \dot{q}_i = u_i \quad (22.2)$$

where u_i is the input voltage on the rotor circuit, C_e^i is the coefficient of proportionality between the contra-electromotor force of the rotor and the rotation speed of the motor (this force is the voltage developing due to rotation of the rotor coil in the magnetic field), and R_r^i is the rotor coil resistance. Based on Equations (22.1) and (22.2) we can write:

$$J_M^i N_M^i N_V^i \ddot{q}_i + P_i = C_M^{i'} u_i - (B_C^i + C_E^{i'}) \dot{q}_i \quad (22.3)$$

where the following notations are introduced:

$$C_M^{i'} = N_M^i C_M^i / R_r^i \text{ and } C_E^{i'} = N_V^i N_M^i C_M^i C_e^i / R_r^i$$

In order to write the model of actuators in the state space, let us introduce the state vector in the form:

$$x_i = (q_i, \dot{q}_i)^T \quad (22.4)$$

Now, instead of Equation (22.3) we may write:

$$\dot{x}_i = A_i x_i + b_i u_i + f_i P_i \quad (22.5)$$

where A_i is the matrix of dimensions 2×2, and f_i are vectors of dimensions 2×1 given by:

$$A_i = \begin{bmatrix} 0 & 1 \\ 0 & -(B_C^i + C_E^{i'})/(J_M^i N_V^i N_M^i) \end{bmatrix}, \quad b_i = \begin{bmatrix} 0 \\ C_M^{i'}/(J_M^i N_V^i N_M^i) \end{bmatrix}, \quad f_i = \begin{bmatrix} 0 \\ -1/(J_M^i N_V^i N_M^i) \end{bmatrix} \quad (22.6)$$

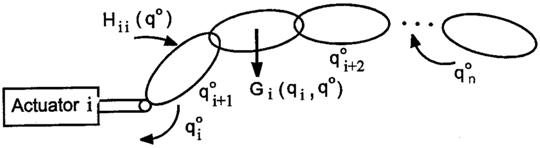

FIGURE 22.3 Actuator in the i-th joint of the robot (the remaining joints are fixed).

The actuator is driving the i-th joint while all other joints are in some fixed positions $q_j = q_j^0$, j = 1, 2, …, n, j ≠ i. The i-th actuator is driving the mechanical part of the robot (kinematic chain) around the i-th joint. In the given fixed positions of the joints q_j^0, j > i, the mechanical part of the robot has a constant moment of inertia around the i-th joint $H_{ii}(q_j^0)$ (see Figure 22.3). The actuator practically drives the set of links which together has a moment of inertia around the i-th joint $H_{ii}(q_j^0)$. These links also produce gravitational moment around the axis of the i-th joint.

This moment depends on the positions in which the joints are fixed and the current (variable) angle (linear displacement) of the i-th joint, i.e., $G_i(q_j^0, q_i)$. Thus, the moment produced by the mechanism around the i-th axis (i.e., around the shaft of the i-th motor) might be written as:

$$P_i = H_{ii}(q_j^0)\ddot{q}_i + G_i(q_j^0, q_i) \tag{22.7}$$

If we introduce the dynamic model of the rotation of the mechanism around the i-th axis in the model of the actuator (22.5), we obtain the model of the actuator's dynamics and the mechanism driven by the actuator in the following form (for simplicity, we shall write $H_{ii} = H_{ii}(q_j^0)$ and $G_i = G_i(q_j^0, q_i)$):

$$\dot{x}_i = \bar{A}_i x_i + \bar{b}_i u_i + \bar{f}_i P_i \tag{22.8}$$

where

$$\bar{A}_i = \begin{bmatrix} 0 & 1 \\ 0 & -(B_C^i + C_E^{i'})/(J_M^i N_V^i N_M^i + H_{ii}) \end{bmatrix}, \quad \bar{b}_i = \begin{bmatrix} 0 \\ C_M^{i'}/(J_M^i N_V^i N_M^i + H_{ii}) \end{bmatrix},$$

$$\bar{f}_i = \begin{bmatrix} 0 \\ -1/(J_M^i N_V^i N_M^i + H_{ii}) \end{bmatrix}$$

The model (22.8) represents the object of control (the actuator and the mechanism, the mechanical part of the robot which has to be controlled).

22.3.2 Synthesis of Servosystem

The task is to synthesize such a control law of the actuator and the joint (robot mechanical part) which should ensure that once the position of the joint is set at desired value q_i^0 the joint will be driven to this position in an adequate way. The control law accomplishing this task is usually a servosystem (servo), the scheme of which is presented in Figure 22.4.

A servo for control of the i-th actuator and joint consists of the following (basic) elements: a position sensor which provides information on the current (actual) position of the i-th joint and of the shaft of the actuator q_i (usually a potentiometer, or opto-encoder, etc.); a rotational (or displacement) velocity sensor of the joint and the motor $\dot{q}_i$ (usually tachogenerators are used, or numerical differentiation of the position/angle is applied); a differentiator which provides the difference

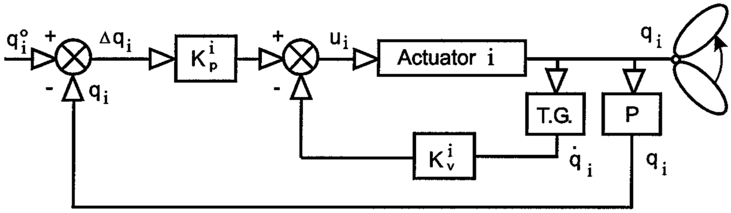

FIGURE 22.4 Positional servosystem (servo).

between the set (desired) position of the i-th joint, q_i^0 and actual position, q_i, obtained from the position sensor; an amplifier of the position error which amplifies the position error signal $\Delta q_i = q_i^0 - q_i$ by k_p^i times, where k_p^i represents the position gain; the velocity signal amplifier (i.e., information on current velocity) which amplifies the signal from the velocity sensor of the joint $\dot{q}_i$ by k_v^i times, where k_v^i represents the velocity feedback gain (in the following, we call it velocity gain).

The way a servosystem operates is obvious from Figure 22.4. The information on the actual joint position is returned as feedback and the difference between the desired and actual position is amplified by k_p^i times. This represents the input signal for the actuator. If $q_i^0 > q_i$ this produces a positive signal which drives the motor so increase q_i until it reaches q_i^0; if $q_i^0 < q_i$, negative signal appears which drives the actuator toward decrement of the angle q_i until it reaches q_i^0; when q_i reaches q_i^0 the error Δq_i reduces to zero, and the signal at the actuator input also falls to zero, which in turn means that the actuator is stopped. However, due to rotor of the motor's inertia and the inertia of the mechanism $J_M^i N_V^i N_M^i + H_{ii}(q_j^0)$, the motor cannot be stopped instantly, and it could incur over-shooting, i.e., the real position may overshoot the desired position, q_i^0, before the motor stops. To ensure an adequate positioning of the joint (without overshoot) we have introduced a velocity feedback loop: the information (signal) from the velocity sensor is amplified k_v^i times and brought to the actuator input to dampen too sharp changes in the actuator motion that may be caused by the position feedback loop.

Therefore, the servosystem generates the following signal at the actuator input:

$$u_i = -k_p^i (q_i - q_i^0) - k_v^i \dot{q}_i = k_p^i \Delta q_i - k_v^i \dot{q}_i \qquad (22.9)$$

The synthesis of a servosystem means selection of the position and velocity gains to achieve a satisfactory positioning of the joint in the desired position q_i^0. This means that if, e.g., a signal of a step type which corresponds to a desired position of the joint (Figure 22.5) is fed at the servosystem input, the servosystem response, i.e., the resulting movement of the joint, depends upon the selection of the feedback gains. It can be shown, by solving the differential Equations (22.8) with the input defined by (22.9), that the response of the servo can appear in three various forms depending on the selection of k_p^i and k_v^i (see Figure 22.5):

1. The servo can be underdamped, in which case the joint rapidly moves from its initial position and reaches the desired position q_i^0 but then overshoots it, i.e., q_i gets values that are higher than q_i^0 and, then, it oscillates around the desired position before settling at the final desired value.
2. The servo can be critically damped. In this case, the joint reaches the desired position relatively quickly, but there are no overshoots or oscillations, and the joint quickly settles at the given q_i^0.
3. The servo can be overdamped, in which case as the joint slowly approaches the desired position, there are no overshoots or oscillations, but the settling period is considerably longer than in the case of a critically damped servo.[7]

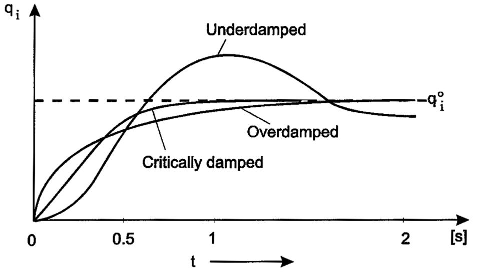

FIGURE 22.5 Responses of servo to step input.

These three types of servo responses can be described by the following functions (as the solutions of the differential Equations (22.8) and (22.9), depending on k_p^i and k_v^i):

1. The under-damped servo:

$$q_i - q_i^0 \approx C_1 e^{-\xi_i \omega_i t} \sin(\omega_i \sqrt{1-\xi_i^2}\, t) + C_2 e^{-\xi_i \omega_i t} \cos(\omega_i \sqrt{1-\xi_i^2}\, t) \qquad (22.10)$$

2. The critically damped servo:

$$q_i - q_i^0 \approx C_3 e^{-\omega_i t} \qquad (22.11)$$

3. The overdamped servo:

$$q_i - q_i^0 \approx C_4 e^{-(\xi_i \omega_i t + \omega_i \sqrt{1-\xi_i^2}\, t)} + C_5 e^{-(\xi_i \omega_i t - \omega_i \sqrt{1-\xi_i^2}\, t)} \qquad (22.12)$$

In the functions (22.10)–(22.12) C_i represent constants (which depend on q_i^0) while ξ_i represents the damping factor and ω_i is the so-called characteristic frequency of the servo. The damping factor ξ_i defines whether the servo is critically damped, overdamped, or underdamped. If:

$$\begin{aligned} &\xi_i < 1, \text{ then the servo is underdamped} \\ &\xi_i = 1, \text{ then the servo is critically damped} \\ &\xi_i > 1, \text{ then the servo is overdamped.} \end{aligned} \qquad (22.13)$$

The damping factor ξ_i and the characteristic frequency ω_i of the system are the features of the servo that are direct functions of the selection of the feedback gains k_p^i and k_v^i, as well of the parameters of the actuator and the mechanical part of the robot. It can be shown that[8]

$$\xi_i = (C_M^{i'} k_v^i + C_E^{i'} + B_C^i) / [2\sqrt{C_M^{i'} k_p^i (J_M^i N_V^i N_M^i + H_{ii})}] \qquad (22.14)$$

$$\omega_i = [\sqrt{C_M^{i'} k_p^i} / \sqrt{J_M^i N_V^i N_M^i + H_{ii}}] \qquad (22.15)$$

It is obvious by the selection of the gains k_p^i and k_v^i that it is possible to directly influence ξ_i and ω_i, and by this it is possible to directly change the character of the servo's response, and to directly influence the way in which the joint is driven to the imposed (desired) position.

In the selection of k_p^i and k_v^i several requirements have to be satisfied:

1. The servo controlling the joint of a robot must not be underdamped under any circumstances. If a servo is underdamped, an overshoot of a desired joint position would occur and oscillations would appear. This is not acceptable with robots, because if the desired position of the link is close to some obstacle in a workspace, and an overshoot occurs, the robot could hit or collide with the obstacle. The servo, therefore, has to be overdamped ($\xi_i > 1$) or critically ($\xi_i = 1$) damped. As the servo's response is significantly slower if it is overdamped, to achieve a response as fast as possible (but without an overshoot and oscillations), it is most suitable that the servo is critically damped.
2. Up to now we have ignored the influence of the gravitational moment about the joint and actuator axis G_i. All the above considerations are valid assuming that the external moments are not acting upon the actuator (except the inertia moment, $H_{ii}\ddot{q}_i$). Let us consider the influence of the gravitational moment. When the joint comes close to the desired position q_i^0, the gravitational moment of the mechanism $G_i(q_j^0, q_i^0)$ is acting about the axis of the joint and the actuator. Because the error between the desired and actual position Δq_i would drop to zero and as the actuator is stopped the velocity, $\dot{q}_i$ also would fall to zero, and the signal at the actuator input would also have to drop to zero in accordance to Equation (22.9). This means that the driving torque produced by the actuator would also fall to zero. However, the actuator should produce the torque to compensate for the gravitational moment G_i (if not, the gravitational moment causes movement of the joint). To produce the actuator torque which would compensate for the external load $G_{i,}$ some signal u_i must be generated at the actuator input. Looking at Equation (22.9) it is obvious that such a signal can be generated only if some error occurs between the actual and the desired positions, once the joint motion is terminated. The error in the positioning of the joint which appears in a steady state due to external load G_i is called the steady-state error. From Equations (22.8) and (22.9) it is easy to calculate this error as:

$$\Delta q(\infty) = [G_i(q_j^0, q_i^0) / C_M^{i'} k_p^i) \tag{22.16}$$

   i.e., the steady-state error is inversely proportional to the position gain. Because our aim is to reduce the error in robot positioning to the minimum, it is obviously necessary to increase the position gain as much as possible.
3. The structure of the robot itself has its own frequency at which the resonant oscillations of the entire robot structure appear. This frequency is called the structural frequency ω_o. According to requirement (1), the gains have to be selected in a way to ensure that the servos are always critically damped. However, because the damping factor ξ_i depends upon the different parameters of the actuators and the mechanism, it is possible that the oscillations of the servos with the frequency ω_i yet may appear. If the characteristic frequency of the servo ω_i is close (equal) to the structural frequency ω_o, the resonant oscillations of the whole structure may appear. Because these oscillations must not be allowed under any circumstances, the characteristic frequency of the servo must be sufficiently below the range of any possible structural frequency; that is, the characteristic frequency must satisfy:[7]

$$\omega_i \leq 0.5\omega_O \tag{22.17}$$

   If condition (22.17) is met, the characteristic frequency is sufficiently low so that the structural frequency cannot be excited and the undesired oscillations cannot appear. The problem lies in the fact that the structural frequency is often hard to determine theoretically and usually is identified experimentally. Because according to Equation (22.15) the characteristic frequency of the servo is directly proportional to the position gain, condition (22.17) means that the position gain has to be limited, it must not be too high to prevent the servo's characteristic frequency from becoming too high and reaching the range of the structural frequency of the robot mechanism.

4. The electrical signals in the servos in Figure 22.4 are never ideally "clean," but always include a certain "noise" superimposed upon the useful information. For example, apart from the useful information, signals from sensors (potentiometers, tachogenerators, etc.) may include noise which originates from various sources (voltage sources are never accurate, certain oscillatory modes always appear, etc.). The noise is usually an order-of-magnitude lower signal than the useful signal. These signals are amplified by the amplifiers k_p^i and k_v^i. If these gains are too high, they amplify not only the useful signals but also the noises; thus, the influence of these noises upon the servo's performance may become significant, which is why limited values of the gains have to be selected.

Based upon the above listed requirements, the gains k_p^i and k_v^i have to be selected. Requirements (3) and (4) are essentially the same, and both demand that the gains to be limited (i.e., the gains must not be too high). Usually if requirement (3) is satisfied, requirement (4) is also met. However, requirement (2) is opposite to these two, as it demands that the position gain should be as high as possible (to keep the steady-state error minimal). Because of this, the following procedure for selecting the gains is usually applied:

1. The maximum allowed position gain is selected to satisfy requirement (3). Based upon Equation (22.15) and (22.17) we get:

$$k_p^i = \frac{\omega_O^2}{4C_M^{i'}}(J_M^i N_V^i N_M^i + H_{ii}) \tag{22.18}$$

2. It is necessary to check whether or not the gain k_p^i calculated by Equation (22.18) also satisfies requirement (4). Because we have selected the maximum allowed k_p^i, we have also satisfied requirement (2) to the highest possible degree.
3. Because the servo has to be critically damped, $\xi_i = 1$, the velocity gain is defined by:

$$k_v^i = [2\sqrt{C_M^{i'} k_p^i}\sqrt{(J_M^i N_V^i N_M^i + H_{ii})} - C_E^{i'} - B_C^i] / C_M^{i'} \tag{22.19}$$

In this way we obtain the gains which satisfy all requirements to the maximum possible degree.

It should be noted that, because the linear servos are applied not only in robotics, but for the control of a number of other systems as well, it is possible to synthesize the feedback gains by applying various other methods developed in automatic control theory. These methods, such as methods in frequency domain, pole-placement methods, linear optimal regulator, etc. can be easily found in the relevant references.[8,9]

Example: For the first joint of the manipulator presented in Figure 22.6, a synthesis of the servo gains should be carried out. The joint is driven by a D.C. electromotor of the type IG2315-P20, the parameters of which are presented in Table 22.1. The data on masses, moments of inertia, lengths, and positions of the centers of masses of the robot links are provided in Table 22.2. It is rather easy to show that the moment of inertia of the mechanical part of the robot around the axis of the first joint is given by

$$H_{11}(q_j^0) = J_{z1} + J_{z2} + J_{z3} + m_3(l_3 + q_3)^2 \tag{22.20}$$

If the third link is fixed in the position $q_3^0 = 0$, the moment of inertia of the mechanism around the axis of the first joint is $H_{11} = 0.403$ kgm^2 ($l_3 = 0.035$ m). Using the values of the actuator

TABLE 22.1 Data on Actuators for the Robot Presented in Figure 22.6

Actuator	1	2	3
C_e^i (V/rad/s)	0.0459	0.0459	0.0459
C_M^i (M/A)	0.0480	0.0480	0.0480
J_M^i (kgm^2)	0.00003	0.00003	0.00003
N_V^i (–)	31.17	2616.0	1570.0
N_M^i (–)	31.17	2616.0	1570.0
R_r^i (Ω)	1.6	1.6	1.6
B_C^i (Nm/rad/s)	0.0058	0.0154	0.00092
			3.0

TABLE 22.2 Data on Robot Presented in Figure 22.6

Link	1	2	3
Mass (kg)	10.0	7.0	4.15
Length (m)	0.213	0.026	0.035
J_x (kg m^2)	—	—	—
J_y (kg m^2)	—	—	—
J_z (kg m^2)	0.0294	0.055	0.318

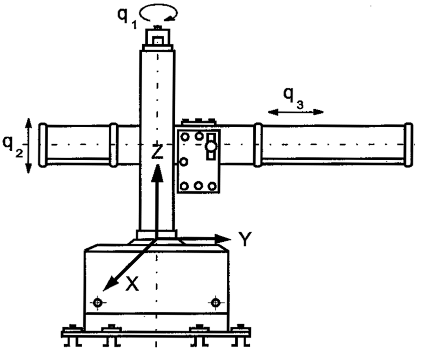

FIGURE 22.6 Robot with three joints.

parameters as given in Table 22.1, we can get the model of the actuator and the joint dynamics in the form (22.8) where the matrices are given by:

$$\bar{A}_1 = \begin{bmatrix} 0 & 1 \\ 0 & -3.117 \end{bmatrix}, \bar{b}_1 = \begin{bmatrix} 0 \\ 2.17 \end{bmatrix}, \bar{f}_1 = \begin{bmatrix} 0 \\ -2.13 \end{bmatrix} \tag{22.21}$$

The structure of the servo to be synthesized is given in Figure 22.4. The gains of the servo are selected according to the above presented approach. Let us assume that the structural frequency is identified (experimentally) to be ω_o = 24 Hz. Based on Equation (22.18) we obtain the position feedback gain as:

$$k_p^i = 62.2 \text{ [V/rad]}$$

Assuming that the noises in the sensor that measures the position of the joint do not exceed 1% of the useful signal and assuming that the total angle of a rotation of this joint is ±180°, we can determine the signal at the amplifier output due to noises to be 0.3 V, which may be considered as negligible. The velocity feedback gain is obtained based on Expression (22.19):

$$k_v^i = 9.62 \text{ [V/rad/s]}$$

This gain is also relatively low so it will not cause significant influence of the noise.

22.3.3 Influence of Variable Moments of Inertia

The described synthesis of a servo is in essence the standard synthesis of a servosystem for mechanical systems. However, robotic systems have some essential differences to other mechanical systems. For example, robots have variable moments of inertia of the mechanisms about the joint axes. We have assumed that only i-th joint can move while all the other joints are fixed in the given positions q_j^0. The moment of inertia of the mechanism about the axis of the i-th joint $H_{ii}(q_j^0)$ depends on the angles (positions) q_j^0 at which the joints behind the i-th joint in the kinematic chain are fixed. If the position (angle) of any joint behind the i-th joint is changed, the moment of inertia of the mechanism about the axis of the i-th joint will change as well.

Let us briefly consider how the variations of the moment of inertia of the mechanism influence the performance of the servo in the i-th joint. Let us assume the gains k_p^i and k_v^i are calculated for such a position of the joints of the robot q_j^0 for which the moment of inertia around the axis of the i-th joint has the value of $\bar{H}_{ii}$. In this case the gains are given by:

$$\begin{aligned} k_p^i &= \frac{1}{4C_M^{i'}} \omega_o^2(\bar{H}_{ii})(J_M^i N_V^i N_M^i + \bar{H}_{ii}) \\ k_v^i &= [2\sqrt{C_M^i k_p^i}\sqrt{(J_M^i N_V^i N_M^i + \bar{H}_{ii})} - C_E^{i'} - B_C^i] / C_M^{i'} \end{aligned} \tag{22.22}$$

where by $\omega_o(\bar{H}_{ii})$ we have denoted the structural frequency of the robot for the moment of inertia $\bar{H}_{ii}$. It has been shown[7] that the structural frequency is inversely proportional to the square root of the moment of inertia of the mechanism, i.e.,

$$\omega_o(H_{ii}) = \frac{k}{\sqrt{(J_M^i N_V^i N_M^i + H_{ii})}} = \frac{\omega_o(\bar{H}_{ii})\sqrt{(J_M^i N_V^i N_M^i + \bar{H}_{ii})}}{\sqrt{(J_M^i N_V^i N_M^i + H_{ii})}} \tag{22.23}$$

where k is the proportionality factor.

If any of the joints in the kinematic chain of the robot (behind the i-th joint) change its position $q_j \neq q_j^0$, then the moment of inertia about the i-th joint axis will also change and become $H_{ii}(q_j) \neq \bar{H}_{ii}$. In this case the characteristic frequency of the i-th joint servo can be obtained in the following form (if we introduce the expression (22.22) for the position gain in Equation (22.15)):

$$\omega_i(H_{ii}) = \frac{\omega_o(\bar{H}_{ii})\sqrt{(J_M^i N_V^i N_M^i + \bar{H}_{ii})}}{2\sqrt{(J_M^i N_V^i N_M^i + H_{ii})}} \tag{22.24}$$

Obviously, the characteristic frequency of the i-th servo has to satisfy the following inequality:

$$\omega_i(H_{ii}) = \frac{\omega_o(\bar{H}_{ii})\sqrt{(J_M^i N_V^i N_M^i + \bar{H}_{ii})}}{2\sqrt{(J_M^i N_V^i N_M^i + H_{ii})}} \le \frac{1}{2}\omega_o(H_{ii}) \tag{22.25}$$

By introducing the expression (22.23) for the structural frequency into (22.25), it can be easily checked that this inequality is always satisfied. This means that regardless of the moment of inertia of the mechanism, requirement (3) (given by Equation (22.17)), which stipulates that the characteristic frequency of the servo has to be sufficiently beyond the structural frequency, is always satisfied (if the position gain is selected according to Relation (22.22)).

However, the damping factor of the servo in the i-th joint varies with the moment of inertia of the mechanism according to the following Equation (based on Equations (22.14) and (22.22)):

$$\xi_i = \frac{C_M^{i'} k_v^i + C_E^{i'} + B_C^i}{2\sqrt{C_M^{i'} k_p^i}\sqrt{J_M^i N_V^i N_M^i + H_{ii}}} = \frac{\sqrt{J_M^i N_V^i N_M^i + \bar{H}_{ii}}}{\sqrt{J_M^i N_V^i N_M^i + H_{ii}}} \tag{22.26}$$

If the j-th joint changes its position to the one in which the mechanism's moment of inertia around the i-th joint H_{ii} is less than $\bar{H}_{ii}$ for which the servo gains were computed, i.e., if $\bar{H}_{ii} > H_{ii}$, the servo is obviously overdamped in the new position of the mechanism, i.e., $\xi_i > 1$. However, if the mechanism comes into the position in which the robot's moment of inertia mechanism around the i-th joint is greater than the moment of inertia $\bar{H}_{ii}$ for which the gains were computed, i.e., if $\bar{H}_{ii} < H_{ii}$ it is obviously $\xi_i < 1$. This means the servo would be underdamped. As we have explained above (requirement 1), the servo for robots must not be underdamped under any circumstances. To ensure that the servo is always over-critically damped ($\xi_i > 1$), we must not allow the case $\bar{H}_{ii} < H_{ii}$. This leads to the following conclusion: to ensure that the servo is always over-critically damped, the gains have to be selected for the mechanism's position for which the moment of inertia of the mechanism around the i-th joint is maximal. As can be seen from Equation (22.26), the damping factor does not depend upon the selection of the position gain (if the velocity gain is selected according to Equation (22.22)). Thus, we have to select the velocity gain for the mechanism's position for which the mechanism's moment of inertia around the axis of the i-th joint $\bar{H}_{ii}$ is at the maximum possible.

The procedure is as follows. All possible positions of the mechanism should be examined (by varying the joints angles q_j) and the maximum moment of inertia of the mechanism $\bar{H}_{ii} = \max H_{ii}(q_j)$ should be determined. For the defined moment of inertia we have to compute the velocity gain k_v^i according to Equation (22.22). In all positions of the mechanism for which $H_{ii}(q_j) \ne \bar{H}_{ii}$ the servo must be overdamped (according to Equation (22.26) because $\bar{H}_{ii} > H_{ii}$). However, if the moment of inertia varies so much that in some positions of the mechanism $\bar{H}_{ii} >> H_{ii}$, the damping factor can become too high $\xi_i >> 1$, which in turn means that the servo is very over-critically damped, the positioning is very slow, and the performance of the servo then may become nonuniform depending on the mechanism position, which is unacceptable for any robot application. To ensure that robot performance is nearly uniform in all positions of the mechanism, we have to ensure that the damping factor is approximately constant. To achieve this we must introduce the variable velocity gain k_v^i (because the damping factor does not depend upon the selection of the position gain). For each position of the mechanism we have to compute the moment of inertia $H_{ii}(q_j)$ and determine the gains k_v^i so as to achieve $\xi_i = 1$. The implementation of a variable gain is significantly more complex than the implementation of fixed gains. Another way to compensate for the influence of the variable moment of inertia of the mechanism is by an introduction of global gain (see 22.4.2.).

However, if the variation of the mechanism's moment of inertia is not too high, quite satisfactory performance of the servo can be obtained even with constant velocity gains (computed for max $H_{ii}(q_j)$). If we consider Equation (22.26) for the damping factor, it is obvious that the moment of inertia of the motor rotor and the reduction ratio of the gears have an effect upon the variation of

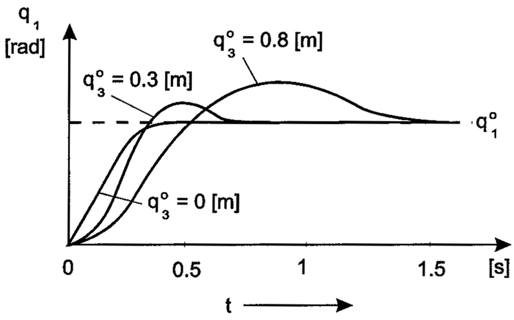

FIGURE 22.7 Responses of the servo in the first joint of the robot presented in Figure 22.6 for various positions of the third joint.

the damping factor with the variation of $H_{ii}(q_j)$. If $J^i_M N^i_V N^i_M >> (\overline{H}_{ii} - H_{ii}(q_j))$, it is obvious that the damping factor will not change significantly regardless of the moment of inertia's variation of the mechanism. In other words, if the equivalent moment of inertia of the motor's rotor is large with regard to the variation of the mechanism's moment of inertia, we may expect that the performance of the servo will be uniform (and approximately critically damped) for all positions of the mechanism, even if we keep the velocity gain fixed. Thus, by selecting a large (powerful) motor and gears we may eliminate the influence of the variable mechanism's moment of inertia. This approach is often applied in the design of robots. However, it is obvious that such a solution has certain drawbacks from the point of view of power consumption, unnecessary loading of joints, as well as the use of unnecessarily powerful actuators and large (heavy) gears.

The bigger gears may be especially inconvenient due to a large backlash and high dry friction coefficients which they may introduce in the system. The introduction of direct-drive actuators (i.e., motors without gears) effectively solves the problems regarding the backlash and friction, but on the other hand, the variation of the mechanism's moment of inertia may affect the servo's performance with such actuators and, therefore, a more complex control law (e.g., with variable velocity gain) has to be applied.

Example: For the servo in the first joint of the robot presented in Figure 22.6, in the previous example, we have computed the gains when the third joint is in the position $q^0 = 0$. Considering Equation (22.20) for the moment of inertia of the mechanism around the axis of the first joint, it is obvious that if the third joint is set in the position $q^0 > 0$ the moment of inertia of the mechanism H_{ii} will be higher and the damping factor will be less than 1. Using Equation (22.26), the damping factor for the position of the third joint, $q_3^0 = 0.3$ m, can be calculated as:

$$\xi_1 = \frac{\sqrt{0.435}}{\sqrt{0.895}} < 1$$

Thus, the gains selected in the previous example will not be satisfactory for all positions of the mechanism. In Figure 22.7 the servo's responses for the various positions of the third joint are presented. This is why the gains must be selected for the mechanism's position for which $\overline{H}_{ii} = \max H_{ii}(q_j)$. In this case, H_{ii} is at maximum if q_3^0 is at maximum, i.e., for $q_3^0 = 0.8$ m. We may calculate that $\overline{H}_{ii}$ ($q_3^0 = 0.8$ m) = 3.323 kg m^2, and the gains are obtained as:

$$k_p^i = 62.2 \text{ [V/rad]}, \; k_v^i = 27.5 \text{ [V/rad/s]}$$

If we compute the gains in this way, the servo will be overdamped for all positions of the mechanism. According to Equation (22.26) the damping factor changes with the variation of q_3^0 as

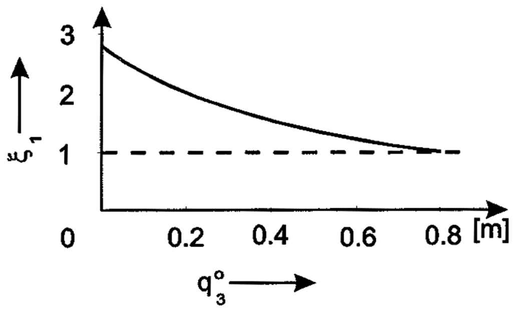

FIGURE 22.8 The variation of the damping factor of the servo in the first joint of the robot presented in Figure 22.6 for various positions of the third joint.

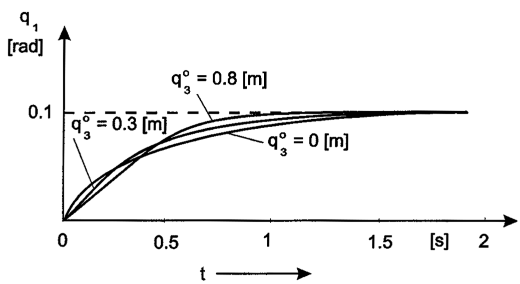

FIGURE 22.9 Responses of the servo in the first joint of the robot presented in Figure 22.6 for various positions of the third joint.

presented in Figure 22.8. It can be seen that for $q_3^0 = 0$, the servo is strongly overdamped, which causes slow positioning of the first joint. In Figure 22.9, the response of the first joint for various positions of the third joint is presented. To achieve a more uniform positioning of the joint it is necessary to introduce (a) variable gains, (b) global control loop, or (c) to apply a larger actuator and gears with a higher equivalent moment of inertia of the rotor.

22.3.4 Influence of Gravity Moment and Friction

We have already explained that the gravity moment of the mechanism causes a steady-state error in robot positioning. Because our aim is to minimize the errors in robot positioning, we have to consider various possibilities to compensate for the influence of gravity moments:

1. We have shown above that a steady-state error is directly proportional to the gravity moment and inversely proportional to the position gain and the moment coefficient of the actuator. We have shown as well that if we select higher position gain the steady-state error will be reduced. However, the position gain is limited by the resonant structural frequency and noises, so the steady-state error cannot be eliminated beyond a certain limit by purely increasing the position gain. Obviously, by the selection of a more powerful actuator (with a higher moment coefficient) and larger gears (with a higher moment reduction ratio), one may decrease the steady-state error, but this solution has some drawbacks, as already pointed out above.
2. Gravity moments can be compensated for by introducing an additional signal at the actuator input; this signal is proportional to the gravity moment (see Figure 22.10). In this case the control system has to compute the gravity moment of the i-th joint $G_i(q_j^0, q_i)$ as a function of the coordinates (positions) of the robot's joints, and generate at the actuator input an

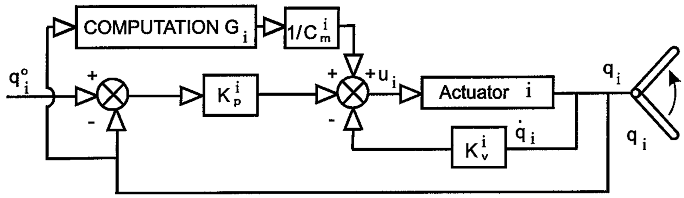

FIGURE 22.10 Positional servo with gravity moment compensation.

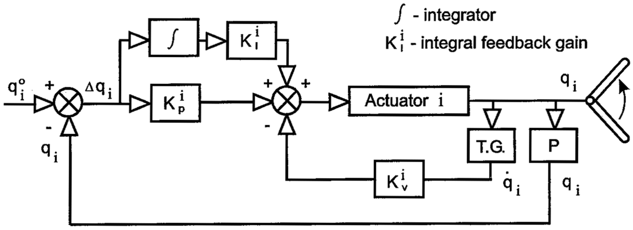

FIGURE 22.11 PID regulator in the i-th joint of the robot.

additional signal which will produce a compensating torque. Thus, the input signal for the actuator is defined by:

$$u_i = -k_p^i(q_i - q_i^0) - k_v^i \dot{q}_i + \frac{G_i}{C_M^{i'}} \tag{22.27}$$

In this way we can eliminate the steady-state error due to the gravity moment. However, a drawback of this solution lies in the fact that it requires the control system to compute gravity moments which, in turn, ask for an accurate identification of the parameters of the mechanism (masses, centers of masses, lengths of links).

3. A steady-state error can be eliminated by introducing an integral feedback loop, i.e., a feedback loop with respect to the integral of the position error (see Figure 22.11). Thus, the PID regulator is obtained (P, proportional; I, integral action; D, differential) which is often applied in practice to a number of systems. In this case, the signal for the actuator input is generated as:

$$u_i = -k_p^i(q_i - q_i^0) - k_v^i \dot{q}_i + k_I^i \int_0^t (q_i^0(t) - q_i)dt \tag{22.28}$$

where k_I^i is the integral feedback gain. The integral feedback has the role of producing a signal proportional to the integral of the position error when the servo approaches the desired position. This signal obviously compensates for the external load and eliminates the steady-state error. There are obvious advantages of this solution over the previous one: the PID solution does not require knowledge of the robot parameters, and the PID regulator compensates other (time constant) perturbing moments' action about the joint axis (these perturbing moments need not be identified, but the PID regulator may compensate them). However, the synthesis of the gains for the PID regulator (which will not be considered here, see, e.g., Paul,[13] is not simple because with the PID regulator it is not possible to satisfy all the above defined requirements upon the servo (e.g., it is not possible to eliminate overshoots, etc.).

4. Finally, a steady-state error due to gravity moments can be reduced by introducing brakes in the joints, which should hold a joint in the desired position once the servo reaches it. This solution is rather simple regarding the control, but often it cannot technically be applied and is inconvenient for an elimination of errors due to the gravity moments if trajectory tracking has to be realized.

Besides the gravity moments, friction forces may also affect the performance of a servo. These forces about the joint axis also cause errors in servo positioning and operation. In this, special problems arise due to static friction forces that appear when the joint starts to move from the still position which differs from the dynamic friction forces during the motion. Compensation of these forces can be realized by one of the above listed methods for the compensation of gravity moments. However, the model and parameters of these forces are often very difficult to identify, and therefore, computation and introduction of additional compensation signals (analogous to the solution in Figure 22.10) cannot be easily implemented. The reliability of such a solution may not be appropriate. The compensation signal can be identified experimentally.

Backlash in the gears, elastic modes, and other nonlinear effects, the models of which are not simple to identify, also may affect performance of the servo. One must carefully consider these effects during synthesis and implementation of servos for robots.

Additionally, it should be mentioned that the amplitude of the input signal to the actuators is constrained, which limits the speed of the servo's positioning if the given (desired) position is far from the initial position of the joint.

22.3.5 Synthesis of the Servosystem for Trajectory Tracking

Up to now we have considered the problem of positioning of the joint in the set (desired) position q_i^0. At the input of the servo a desired position q_i^0 is fed and the joint is positioned following the above described process. However, as we have already underlined, modern industry and other applications of robots require robots which have to be not only precisely positioned in various positions in the workspace, but can also track continual trajectories. For example, with arc welding, the robot hand should move along a prescribed trajectory in the workspace with an accurately defined velocity. Often, a definition of the desired trajectory can be achieved by imposing a set of discrete points (positions) in the space through which the robot hand has to pass (the point-to-point motion). However, with the above-mentioned example of arc welding it is necessary to implement a motion of the hand (tool) along a continuous path in the workspace. In this case, all joints of the robot have to realize their desired trajectories as continual functions of time $q_i^0(t)$. This is why it is necessary to consider how the servo can ensure tracking of the continual trajectory of the joint coordinates (assuming the rest of the joints are fixed).

Let us assume that at the servo input (Figure 22.4) a signal $q_i^0(t)$ introduced which is a continuous function of time. This signal corresponds to the desired nominal trajectory of the i-th joint, i.e., to the desired variation of the joint angle along the time. This means that the joint angle has to track the trajectory $q_i^0(t)$. The servo must ensure that the actual joint position is as close as possible to $q_i^0(t)$ at each time instant. Even more important, it should ensure that the rotational speed of the joint $\dot{q}_i(t)$ is as close as possible to the desired trajectory of the speed $\dot{q}_i^0(t)$ at each time instant. However, if we just feed the desired trajectory $q_i^0(t)$ at the input of the servo (Figure 22.4), the servo output–joint angle will undoubtedly have a delay with respect to the given (desired) trajectory $q_i^0(t)$. This delay is due to dynamic characteristics of the actuator and the mechanism driven by the actuator (i.e., the inertia of the actuator rotor and of the mechanism, friction, and contra-electromotive force which is generated in the motor). Here, we will not analyze mathematically this phenomenon, but it is clear that it is necessary to compensate for this delay in order for the joint to implement accurately the desired trajectory. To compensate for this delay caused by servo dynamics, we may introduce the feedforward signal (precompensation signal).

The feedforward term has to compensate for a delay of the servo along the given nominal trajectory and can be synthesized in various ways. Here, we briefly present one simple procedure for the synthesis of the feedforward term for a robot servo. The model of the actuator of the i-th joint is given by Equation (22.8) in the state space. The nominal trajectory of the joint $q_i^0(t)$ has to be realized. Because the trajectory of the joint is given, by differentiating it we can obtain the desired variation (a trajectory) of the joint velocity $\dot{q}_i^0(t)$. The state vector of the servo and the actuator is given by (22.4). This means that the desired nominal trajectory of the state vector of the servo and the actuator is given as well, $x_i^0(t)=(q_i^0(t),\ \dot{q}_i^0(t))^T$. At each time instant t the difference between the actual state vector x_i (t) and the nominal trajectory $x_i^0(t)$ should be as small as possible. The feedforward term represents the signal at the actuator input $\bar{u}_i^0(t)$ which satisfies the following equation:[10]

$$\dot{x}_i^0(t)=\bar{A}_i\, x_i^0(t)+\bar{b}_i\,\bar{u}_i^0(t)+\bar{f}_i\, G_i(q_j^0,q_i^0) \tag{22.29}$$

i.e., the signal $\bar{u}_i^0(t)$ satisfies the model of the actuator and joint (Equation (22.8)) along the specified trajectory $x_i^0(t)$. The signal $\bar{u}_i^0(t)$ represents the programmed signal as a function of time and is called local nominal, programmed control. The name "local" originates from the fact that this signal is computed for one local actuator and one joint ignoring the other joints (i.e., they are assumed to be fixed). The name "programmed" originates from the fact that this control is a function exclusively of time, and not of the actual (temporary) state of the joint and the actuator (i.e., it is not dependent on the actual position and speed of the joint), and therefore, it represents the programmed input for the actuator corresponding to the programmed trajectory $x_i^0(t)$. Taking into account the form of the matrix and vectors in Equation (22.8) it can be easily shown that the signal $\bar{u}_i^0(t)$ satisfying Equation (22.29) can be computed according to the following equation:

$$\bar{u}_i^0(t)=[(J_M^i N_V^i N_M^i+\bar{H}_{ii})\ddot{q}_i^0(t)+(B_C^i+C_E^{i'})\dot{q}_i^0(t)+G_i(q_j^0,q_i^0(t))]\,/\,C_M^{i'} \tag{22.30}$$

where $\ddot{q}_i^0(t)$ represents the desired variation of joint acceleration along the specified trajectory $q_i^0(t)$, and it is obtained by the differentiation of the nominal trajectory of the velocity $\dot{q}_i^0(t)$. Based on (22.30) we obtain the local nominal control using the specified nominal trajectory of the joint. If the local nominal control is fed into the input of the actuator (as a programmed signal), and if no perturbation is acting upon the joint, the actuator and joint would move along the specified trajectory $q_i^0(t)$. However, it is obvious certain perturbations always act upon the system, and the model and parameters used for the computation of (22.30) are not ideally accurate. In addition, in the initial moment t = 0, the joint angle $q_i(0)$ need not correspond to the nominal angle $q_i^0(0)$. Because of this, the motion of the joint always deviates from the nominal trajectory when we feed the actuator with the programmed nominal control $\bar{u}_i^0(t)$ only. The behaviour of the actuator and the joint in this case is described by:

$$\dot{x}_i(t)=\bar{A}_i\, x_i(t)+\bar{b}_i\,\bar{u}_i^0(t)+\bar{f}_i\, G_i(q_j^0,q_i) \tag{22.31}$$

Obviously, if $x_i(0)\neq x_i^0(0)$, the actual state vector $x_i(t)$ will not coincide with the nominal trajectory $x_i^0(t)$. Due to this, an additional signal Δu_i must be fed into the input of the actuator to ensure that the state vector $x_i(t)$ is as close as possible to $x_i^0(t)$ when the perturbations are acting upon the system and when $x_i(0)\neq x_i^0(0)$. Let us introduce a vector of deviation of the system state from the nominal trajectory as a difference between the actual state and the nominal state $\Delta x_i(t)=x_i(t)-x_i^0(t)$. The model (22.31) can then be written in the following form:

$$\Delta\dot{x}_i(t)=\bar{A}_i\,\Delta x_i(t)+\bar{b}_i\,\Delta u_i(t)+\bar{f}_i\,[G_i(q_j^0,q_i)-G_i(q_j^0,q_i^0)] \tag{22.32}$$

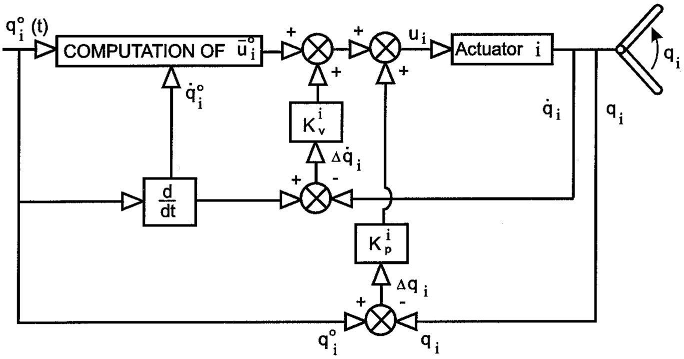

FIGURE 22.12 Servosystem with local nominal control.

Equation (22.32) is called the model of deviation of the system state from the nominal trajectory. We have to ensure that the deviation of the state $\Delta x_i(t) = x_i(t) - x_i^0(t)$ approaches zero, i.e., we have to synthesize the additional control signal Δu_i in such a way that it ensures that the deviation vector is close to zero. The model (22.32) is analogous to the basic model of the actuator and the joint (22.8). It is, therefore, obvious that the additional control signal Δu_i for the deviation model (22.32) can be generated analogously as for the actuator and joint positioning. The problem of a reduction of the deviation state vector of the deviation model (22.32) to zero is analogous to the problem of the positioning the actuator and joint (22.8) into the position $x_i = (0,0)^T$. Thus, the additional control signal can be generated as:

$$\Delta u_i = k_p^i \Delta q_i + k_v^i \Delta\dot{q}_i = -k_i^T \Delta x_i \tag{22.33}$$

where by $k_i = (k_p^i, k_v^i)^T$ is denoted a vector of feedback gains. The total signal which has to be fed to the actuator input is

$$u_i(t) = \bar{u}_i^0(t) + \Delta u_i(t) = \bar{u}_i^0(t) - k_i^T \Delta x_i \tag{22.34}$$

Figure 22.12 presents the control scheme which ensures tracking of the trajectory. The servo for trajectory tracking has a similar structure as the servo for positioning (Figure 22.4). The only difference is in the feedforward term which represents the computation of the local nominal control according to Equation (22.30), and in the fact that instead of feedback by the velocity we introduce the difference between the actual velocity and the nominal velocity $\Delta\dot{q}_i$. This difference (velocity error) is amplified by k_v^i. Because the deviation model (22.32) is analogous to the model (22.8), the synthesis of the gains k_p^i and k_v^i for the servo with the feedforward term is analogous to the synthesis of the gains for positioning of the servo.

It should be mentioned that for the computation of the local nominal control according to Equation (22.30), for the moment of inertia of the mechanism $H_{ii}(q_j)$ the least possible value this moment of inertia may have depending on the position of the rest of the joints q_j should be taken. The reason for this is to avoid overshoots. Therefore, the procedure for selecting the moment of inertia of the mechanism for the calculation of local nominal control is analogous to the one for the computation of the velocity feedback gain (but the minimum value is searched for).

Example: Let us assume that at the input of the servo for the first joint of the robot presented in Figure 22.6 (the servo was synthesized in Section 22.3.2), instead of the position, the signal corresponding to the joint trajectory $q_1^0(t)$ is fed. This trajectory is presented in Figure 22.13 and can be described by

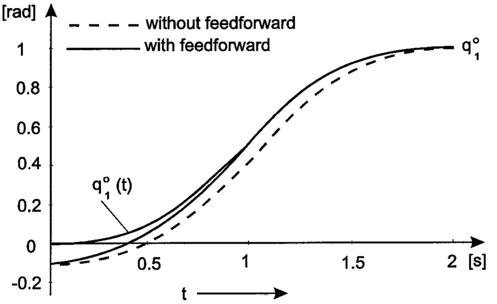

FIGURE 22.13 Trajectory tracking with and without a feedforward term (the first joint of the robot presented in Figure 22.6).

$$q_1^0(t) = a_1 t^2 / 2, \quad 0 < t \leq \tau / 2$$

$$q_1^0(t) = a_1 (\tau t - t^2 / 2 - \tau^2 / 4), \quad \tau / 2 < t < \tau$$

where $a_1 = 1$ rad/sec^2 is the acceleration, and $\tau = 2$ s is the time duration of the movement. If the feedforward is not introduced, but we directly feed the desired trajectory at the input of the servo, the actual trajectory will be delayed to the nominal one, as can be seen from Figure 22.13 (for an initial error $\Delta q_1(0) = q_1^0(0) - q_1(0) = 0.1$ rad). This is why it is necessary to introduce a feedforward term to compensate for this delay.

22.4 Control of Simultaneous Motion of Several Robot Joints

Up to now we have analyzed the control of one single joint of the robot, assuming that all the remaining joints are fixed. However, to execute different control tasks by the robot, the hand of the robot has to be positioned in the workspace. To do this, it is necessary to drive all joints of the robot into certain positions (angles) which correspond to the desired position of the hand. Generally speaking, it would be possible to drive the joints to the certain positions successively, one by one, so that while each joint is moving the remaining joints are fixed. In this case, the control we have observed up to now would be satisfactory. However, it is obvious that such a positioning of the robot, joint by joint, is not efficient from the point of view of the time required to accomplish the task. Obviously, the time required for the positioning of the hand into the desired position if all joints are moving simultaneously is considerably less than if the joints are moving successively. Because one of the main goals in robot design is to achieve a quick as possible working speed, it is clear that for the modern robots simultaneous positioning of all joints must be ensured. Even more if the tracking of a given path of the hand is required, it is obvious that all joints of the robot must move simultaneously (i.e., they have to track their corresponding trajectories simultaneously). Thus, the simultaneous movement of all robot's joints is a must for modern robots.

If several joints of the robot are moving simultaneously, dynamic coupling between the joints must appear. This dynamic coupling must affect the performance of the servo in the robot's joints. We consider the influence of the dynamic forces, and the synthesis of the control which ensures a satisfactory performance of the robot when its joints are moving simultaneously.

22.4.1 Analysis of the Influence of Dynamic Forces

The model of robot dynamics was presented in previous chapters. The moments about the joint axes can be described as functions of the joint angles, velocities, and accelerations:

$$P = H(q)\ddot{q} + h(q,\dot{q}) \tag{22.35}$$

where $H(q)$ is the inertia matrix n × n, $h(q,\dot{q})$ is the vector of the centrifugal, Coriolis, and gravity moments n × 1. If several joints are moving simultaneously, the dynamic moment P_i is acting around the i-th joint. This means that an external load is acting upon the i-th actuator and this load P_i is the function of the angles, velocities, and accelerations of all joints. If only the i-th joint is moving (and all the remaining joints are fixed) upon the actuator is acting the load given by (22.7) which we took into account within the synthesis of the servo in the i-th joint. The dynamic moment caused by simultaneous motion of several joints loads the i-th servo and affects servo performance. In the text to follow we consider qualitatively how certain dynamic forces (moments) in simultaneous motion of several joints affect the performance of the servos. We consider both the positioning problem (by simultaneous positioning of all robot's joints) and the problem of tracking the robot hand trajectory (by simultaneous tracking all the joint trajectories). Let us assume that the desired positions of the joints q_i^0 or the trajectories $q_i^0(t)$ are fed simultaneously at the inputs of all robot servos.

1. **Variable moment of inertia.** If several joints are moving simultaneously, the moment of inertia of the mechanism around the i-th joint varies during the motion, because H_{ii} depends on the positions of all the joints in the kinematic chain behind the i-th joint. In 22.3.3 we considered the influence of the variation of the moment of inertia of the mechanism upon the servo performance. We have shown that it is necessary to compute the feedback gains for the maximum possible value of the moment of inertia of the mechanism to prevent the system from being underdamped. However, we have also seen that if the moment of inertia is significantly varied, the performance of the robot (servo) can be uneven. This is especially inconvenient with simultaneous motion of several joints, as the moment of inertia about a joint varies during the motion, which may cause oscillatory tracking of trajectories or positioning. However, this problem can be solved in one of the ways previously mentioned.
2. **Cross-inertia members.** Accelerations of the j-th joint cause the moment in the i-th joint through cross-inertia moments, which themselves represent the elements of the inertia matrix H off the main diagonal $H_{ij}\,(i \neq j)$. Thus, due to acceleration in the j-th joint $\ddot{q}_j$, an external load $H_{ij}(q)\ddot{q}_j$ acts upon the i-th servo. As we explained above, the external load upon the shaft of the actuator causes errors in the positioning of the joint, or in the tracking of trajectories (because the servo must overcome this external load through a position error which will create a corresponding compensating signal). However, this moment is significant only if accelerations are relatively high. When the robot stops in the desired position, the accelerations drop to zero and, therefore, they do not influence the positioning of the joints, i.e., they do not cause steady-state errors. These moments can cause errors in tracking the trajectories if they are with high accelerations $\ddot{q}_j$. If in a certain application task it is not essential to ensure accurate tracking of fast trajectories, the effects of these moments can be ignored. However, if accurate tracking of the fast trajectories is essential (which means that in each moment the difference between the actual position of the joint and the nominal trajectory must be minimal), then the moments due to cross-inertia members must be compensated for.
3. **Gravity moments.** The effects of gravity moments have already been considered in 22.3.4. In the simultaneous motion of several joints the gravity moments vary during the movement, causing errors both in positioning and tracking of trajectories. The compensations for these moments can be reached through one of the previously described manners, but it should be kept in mind that the gravity moments vary during the tracking of trajectories and, therefore, they cannot be completely compensated for by PID regulators.
4. **Centrifugal and Coriolis moments.** These moments are produced by the velocities $\dot{q}_j$ in the robot's joints. They also act as external loads upon the servos. However, because the centrifugal

and Coriolis forces are directly proportional to rotational (or linear) velocities of the joints $\dot{q}_j$, these forces are significant only if the joints are moving at relatively high speeds. When the robot starts to move or it stops, these forces are negligible, which means they do not affect the positioning of the robot in any desired positions and do not cause steady-state errors. These forces only cause errors in the tracking of fast trajectories. Similarly, as in the case of cross-inertia moments, here we can also conclude that if an accurate tracking of fast trajectories is not required, the effects of these forces can be ignored. However, if the tracking of fast trajectories is essential for the robot application, we must take into account the centrifugal and Coriolis forces within the synthesis of control.

As can be seen from these considerations, when several joints are moving simultaneously, dynamic coupling between the joints appears which affects the positioning and tracking of trajectories. The influence of these dynamic forces upon the positioning and tracking of slow trajectories is not significant and, therefore, the servos synthesized for isolated joints of the robot can easily overcome them. In the previous paragraphs, we presented how at the level of local servos we can compensate for the effects of variable moment of inertia and external loads. These compensations are often quite sufficient to ensure positioning and tracking of slow trajectories even if several joints are moving simultaneously. For such tasks it is quite acceptable to control the robot by local servos synthesized for isolated joints. However, if tracking of fast trajectories is required, the effects of dynamic forces cannot be ignored. Because these forces act as external loads, if the servo's feedback gains are high, the errors caused by these forces may be very negligible, so that even in the case of relatively fast trajectories we may accept the servo synthesized for the isolated joints. However, because the gains are normally limited, as explained before, if we have to ensure accurate tracking of fast trajectories we cannot apply only local servos, the dynamic forces must be compensated for.

22.4.2 Dynamic Control of Robots

A number of robots available in the market are not capable of ensuring accurate tracking of fast trajectories, because their applications were sufficient for local servos without introducing any compensation for dynamic forces. However, because demands upon the robots regarding the high speed and quality of operation (accuracy of tracking of the desired paths) in modern industry and other application domains are increasing, the control systems of the newest generations of robots have to take into account the dynamics of robots. The control law which takes into account all (or some of) dynamic forces in the robotic systems is called the dynamic control of robots.

The basic problem when applying a dynamic control lies in the fact that the dynamic forces acting within the robotic mechanism are generally very complex functions comprising coordinates, velocities, and accelerations of the joints. Thus, if we want the control system to compensate for the effects of these forces, this may lead to relatively complex control laws.

Various approaches have been developed for the synthesis of the dynamic control of robots. A survey of these approaches can be found in Vukobratović et al.[11] Here, we very briefly consider just two approaches: nominal programmed control and global control.[10–12]

In Section 22.3.5 we showed that by applying the nominal local programmed control delays in servos along the trajectory, we can be compensated for the tracking process. If several joints are moving simultaneously, local nominal control cannot compensate for dynamic moments which act about the i-th joint, which is why instead of the local nominal control we may apply the nominal programmed control computed on the basis of the complete dynamic model of the system. This control can be computed in the following way. Let the nominal trajectories of the robot $q_j^0(t)$, i = 1, 2, …, n be given. By differentiating, we get the nominal trajectories of the velocities and accelerations. Now, based upon the model of the mechanism dynamics (22.35), we can compute the nominal driving torques in the robot's joints:

$$P^0(t) = H(q^0(t))\ddot{q}^0(t) + h(q^0(t), \dot{q}^0(t)) \tag{22.36}$$

where by $P^0(t)$ we denote the vector of nominal driving torques, while by $q^0(t)$ we denote the vector of nominal trajectories in all n joints. The nominal driving torques represent the moments which have to be implemented about the joints of the robot to ensure that it moves along the desired nominal trajectories $q_j^0(t)$.

The models of the actuators in the joints are given by Equation (22.5) where P_i represents the external load by which the mechanism acts upon the actuators, i.e., P_i is the driving torque realized by the actuator. The nominal control signal at the actuator input which has to ensure the nominal driving torque must satisfy the equations:

$$\dot{x}_i^0(t) = A_i x_i^0(t) + b_i u_i^0(t) + f_i P_i^0(t), \ i = 1, 2, \ldots, n \tag{22.37}$$

The nominal programmed control $u_i^0(t)$ which satisfies Equation (22.37) can be computed by:

$$u_i^0(t) = [(J_M^i N_V^i N_M^i + \overline{H}_{ii})\ddot{q}_i^0(t) + (B_C^i + C_E^{i'})\dot{q}_i^0(t) + P_i(t))] / C_M^{i'} \tag{22.38}$$

Obviously, the nominal control $u_i^0(t)$ differs from the local nominal control $\overline{u}_i^0(t)$ computed by (22.30), since the former includes the total nominal driving torque which represents the dynamic moment due to the movements of all joints of the robot. Thus, the nominal programmed control compensates not only for the dynamics of the actuator and a single joint, but it also compensates for the dynamics of the complete mechanism (but, only along the nominal trajectory). If we feed the nominal programmed control $u_i^0(t)$ at the inputs of all actuators, and if the models of the actuators and of the mechanism were exact, and the robot in the initial moment is in such a position that $q_i(0) = q_i^0(0)$ for all joints, and if no perturbation is acting upon the robot, the joints of the robot would move along the imposed trajectories. However, all these assumptions do not hold often (or nearly never hold). This is the reason why deviations of the joint coordinates from the nominal trajectories appear whenever we apply the nominal programmed control only. The motion of the joints can be described by the model

$$\dot{x}_i(t) = A_i x_i(t) + b_i u_i(t) + f_i(P_i^0(t) + \Delta P_i), \ i = 1, 2, \ldots, n \tag{22.39}$$

where ΔP_i represents a deviation of the actual load in the joint from the nominal value $P^0(t)$. If $x_i(0) \neq x_i^0(0)$, then $x_i(t)$ will not coincide with $x_i^0(t)$ so we have to introduce an additional control signal at the actuator input. This additional control has to ensure that the actual state $x_i(t)$ is as close as possible to $x_i^0(t)$. The model (22.39) can be written as (based upon Equation (22.37)):

$$\Delta\dot{x}_i(t) = A_i \Delta x_i(t) + b_i \Delta u_i(t) + f_i \Delta P_i, \ i = 1, 2, \ldots, n \tag{22.40}$$

where $\Delta x_i(t) = x_i(t) - x_i^0(t)$. The model (22.40) represents the model of deviation of the actuator state from its nominal trajectory if the motions of the joints of the robot deviate from the nominal trajectory. This model is similar to the basic model of the actuator (22.5) save for the fact that as external moment instead the total moment P_i, here acts as a deviation of the dynamic moment from the nominal driving torque. The nominal programmed control compensates for the nominal driving torque (i.e., nominal moment $P^0(t)$) and by this reduces the effects of the dynamic forces upon the servo performance. If we apply the servo synthesized in Section 22.3.5, the reduced ΔP_i dynamic moment acts upon it. Thus, the servos can overcome this external load more efficiently if we apply the nominal programmed control (see Figure 22.14).

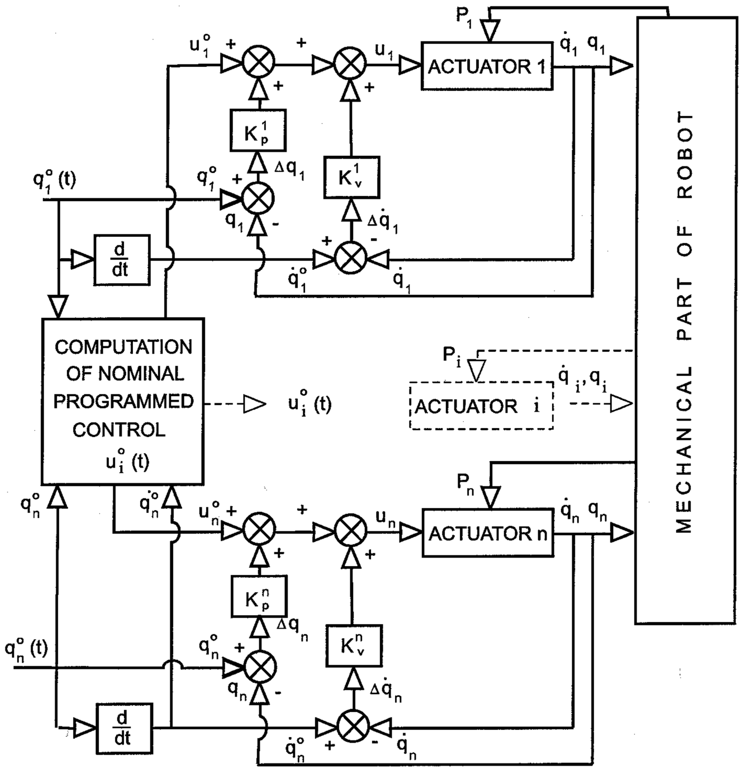

FIGURE 22.14 Control scheme of the robot including the nominal programmed control.

However, the application of the nominal programmed control involves numerous difficulties. To compute the nominal programmed control on the basis of Equation (22.38) we have to compute on-line the complete dynamic model of the robot (22.35) which may be rather complex and requires powerful computer. However, this can be easily achieved by applying standard microcomputers; the main problem is to accurately identify parameters of the actuators and of the mechanisms as well as model different dynamic effects not included in the assumed models (e.g., elastic effects, etc.).

The second possibility to compensate for the dynamic moments during the trajectory tracking is application of the global control. If we apply local nominal programmed control (22.3.5), the external load comes from the moments of the mechanism dynamics. To compensate for the effects of these dynamic moments we may introduce an additional compensating control signal at the actuator input. This additional signal has to be proportional to the dynamic moment P_i acting upon the i-th joint. This additional signal can be calculated in the following form:

$$\Delta u_i^G = -k_i^G \tilde{P}_i \tag{22.41}$$

where k_i^G represents the global gain, and $\tilde{P}_i$ represents the value which is proportional (or equal) to the dynamic moment P_i in the i-th joint. This additional control signal is called the global control because it represents the feedback between the joints. The local servo includes the feedback by local coordinates and velocities of a joint which is controlled by the servo, so it has just local information, compared with the global control which represents an exchange of information between the servos (cross-feedback global loops, see Figure 22.15).

Global control has to compensate for the effects of the dynamic moments by generating the signal at the actuator input which is proportional to this moment. Therefore, the basic problem

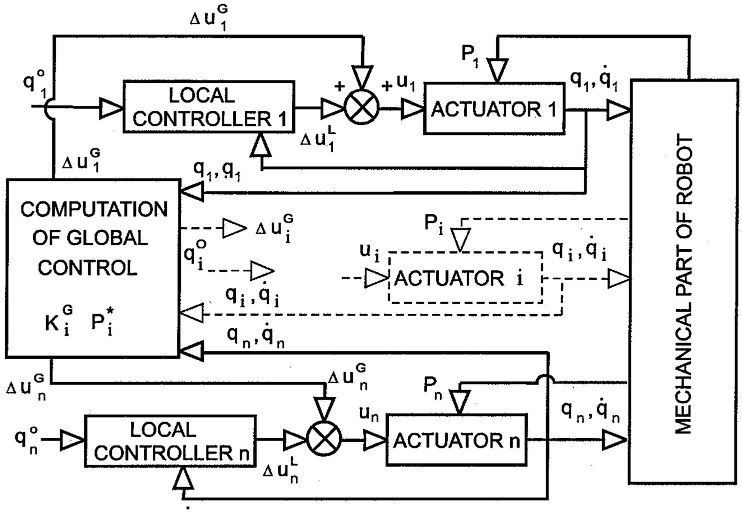

FIGURE 22.15 Control scheme of the robot including global control.

regarding global control is how to obtain information on the actual dynamic moment acting upon the servo P_i.

Two options for implementation of the global control are proposed:[10,11]

1. The dynamic force (moment) which acts upon the servo can be measured directly by a force transducer. By placing a force transducer in the shaft of an actuator, the dynamic force is directly measured. The signal from the force transducer is used as a feedback to the actuator input (multiplied with the global gain k_i^G). By this, the signal is generated at the actuator input which is proportional to the actual dynamic force and produces the additional driving torque to compensate for the dynamic moment. Global control is realized by force feedback. The problems of this option are related to the elastic effects introduced when the force tranducer is installed in the actuator shaft.
2. The second option is by on-line computation of the dynamic moment. Based on the information of the actual robot state (joint coordinates, velocities, and accelerations) obtained from the sensors, the control computer computes the dynamic moments P_i according to the model (22.35). Thus, the computer generates the signal at the actuator input which is proportional to the computed value of the dynamic moment. As we have already explained, the model of the robot dynamics may be very complex, which in turn requires fast computers capable of computing on-line the dynamic moments as a function of the actual state of the robot. This is easy achievable with standard computers. The main problem lies in the accuracy of the model of robot dynamics and in the identification of parameters. However, the effects of all dynamic forces need not be significant, and in many cases it is not obligatory to compute all the dynamic moments of the complete model. Instead, it may be sufficient to compute certain components of the dynamic moments (e.g., gravity moments, cross-inertia members, etc.). In other words, the dynamic moments can be computed by approximative models. The problem is determining which components of the dynamic moments must be compensated for by global control. Generally, this analysis requires application of the computer, i.e., the selection of the adequate dynamic control is most effectively performed by computer-aided control synthesis.

It should be mentioned that various combinations of the control laws are possible. It is possible to apply a nominal programmed control (which compensates for the nominal dynamic moments), and the local servosystems and global control (which in this case has to compensate for the deviation of the real moments from the nominal ones, ΔP_i).

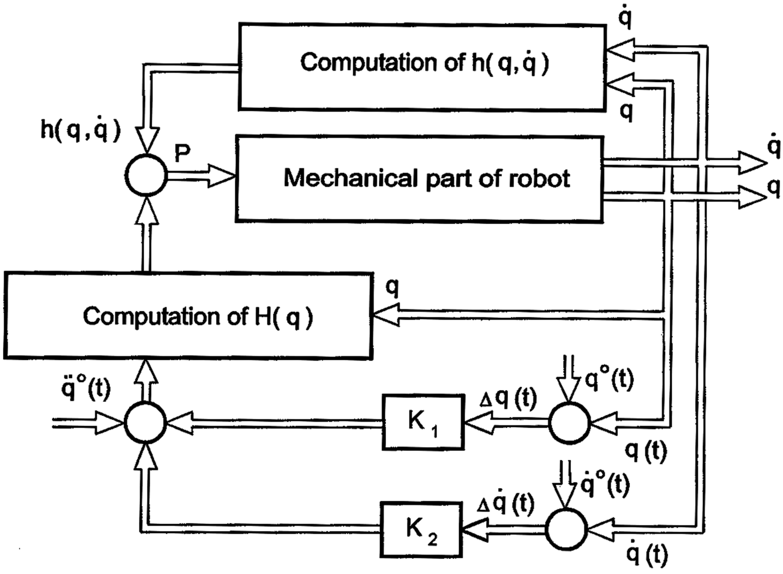

FIGURE 22.16 Control scheme of the inverse problem technique.

22.4.3 Inverse Problem Technique

One of the most investigated dynamic control law is the inverse problem technique. Paul[13] investigated the inverse problem technique, which was called the "computed torque" technique by Bejczy.[14] A similar approach has been taken by Pavlov and Timofeyev.[15] Their approaches include on-line computation of the complete model of robot dynamics, i.e., they involve computation of driving torques by Equation (22.35) using the measured values of internal coordinates q and velocities $\dot{q}$ of the robot and the computed values of internal accelerations $\ddot{q}^o(t)$. Namely, if the desired trajectory is computed, we can obtain $q^o(t)$, $\dot{q}^o(t)$, $\ddot{q}^o(t)$. It has been shown[15] that the robot is asymptotically stable around the nominal trajectory if the driving torques are computed by:

$$P(t) = H(q) \cdot [\ddot{q}^o(t) + K_1(q(t) - q^0(t)) + K_2(\dot{q}(t) - \dot{q}^o(t))] + h(q, \dot{q}) \tag{22.42}$$

where K_1 is an nxn matrix of position feedback gains and K_2 is an nxn matrix of velocity feedback gains; K_1 and K_2 must be chosen in such a way that a trivial solution of

$$\ddot{e} = K_1 e + K_2 \dot{e}$$

is asymptotically stable, where e is $nx1$ vector. However, only driving torques are computed in (22.42). It is necessary to also include the models of actuators (22.5), i.e., to calculate inputs u_i that correspond to computed driving torques (22.42).

The control scheme is presented in Figure 22.16. Obviously, this scheme combines a closed-loop controller with nominal control signals computed on the basis of the equations of motion. In this scheme compensation is provided for time-varying gravitational, centrifugal, and Coriolis forces; the feedback gains are adjusted according to the changes in matrix $H(q)$ of moments of inertia; an acceleration feedforward term is also included to compensate for changes along nominal trajectory.

However, these approaches suffer from several disadvantages. The main problem is that in (22.42) computation of the complete dynamic model of robot is required. For complex robot structures this

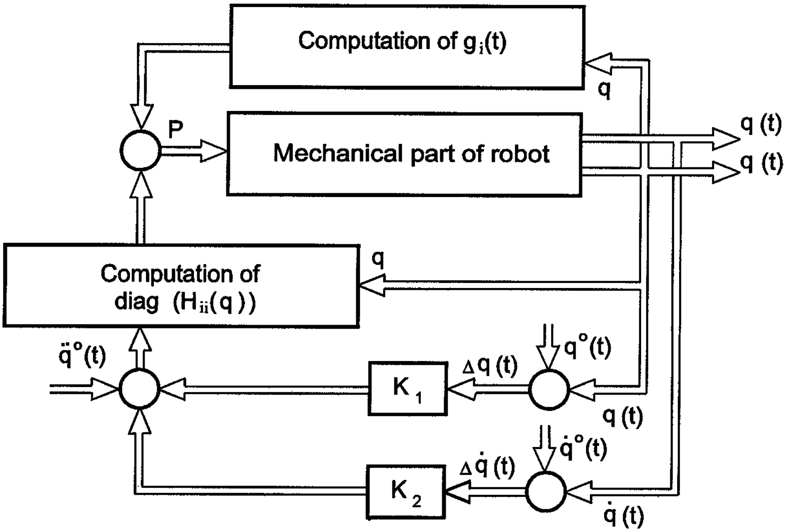

FIGURE 22.17 Control scheme of the inverse problem technique, simplified computation.

requirement may be difficult to satisfy. This is the reason why some authors have tried to implement an approximative model of robot dynamics. They have omitted cross-coupling terms of moment of inertia in the matrix $H(q)$ and centrifugal and Coriolis forces. This means that the model is reduced to diagonal terms in the inertia matrix $H(q)$ and to gravity terms, i.e., the control (22.42) is reduced to:

$$P_i(t) = H_{ii}(q) \cdot [\ddot{q}_i^{\,o}(t) + K_1(q(t) - q^0(t)) + K_2(\dot{q}(t) - \dot{q}^o(t))] + g_i(q(t)) \qquad (22.43)$$

where g_i is the gravity force (moment) in the i-th joint. The control scheme is now reduced to the one presented in Figure 22.17. This computation is considerably simplified, but it is still cumbersome for some types of manipulators. On the other hand, it is questionable whether or not the control efficiency is lost by these simplifications. Paul[13] has found that in his experiments with the Stanford manipulator, the contribution by Coriolis and centrifugal terms is relatively insignificant.

Raibert and Horn[16] have used a partial table lookup approach to automatically simplify the computation on a digital computer. Rather than compute the coefficients in (22.42) each time they are needed (every sampling period), their approach (called the configuration space method) is to look them up in a predefined multidimensional memory organized by the positional variables $q(t)$ (the configuration space controller).

Thus, it is obvious that the main difficulty with the inverse problem technique is on-line computation of the dynamic model of the robot. Actually, the analysis of the complexity of the model that is needed for this control law to achieve sufficiently good tracking of nominal trajectory has not been given. However, several other problems with this approach exist. The implementation of control law (22.42) requires perfect knowledge of mechanism parameters. So, it is questionable whether the control (22.42) is robust enough to withstand all parameter variations.

Timofeyev has extended this approach to adaptive control in the case of unknown and variable parameters of the robot.[17]

22.4.4 Effects of Payload Variation and the Notion of Adaptive Control

Up to now we have assumed that all the robot's parameters are constant and precisely known in advance. However, some parameters in rotobtic systems (such are the coefficients of viscous and dry friction in the actuators and joints, backlash, and some actuator cofficients) are not sufficiently defined precisely in advance and can vary during task execution. During the operation of the robot these parameters vary, but rather slowly. Often they do not affect robot performance if the values of the parameters stay with the limits allowed. Thus, it is often unnecessary to include any compensation for variations of these parameter in control law. However, some parameters of the robotic system vary significantly and relatively fast, and have considerable effect upon robot performance. Such parameters are masses, dimensions, and moments of inertia of the payload that is carried by the robot.

The presence of the payload causes changes to the moments of inertia of the mechanism around the axes of the robot joints. In Section 22.3.3 we considered the influence of the mechanism's moment of inertia variation upon servosystem performance. We have seen that when selecting the velocity feedback gain we have include the maximum value of the moment of inertia of the mechanism around the joint in order to prevent overshoots. When the gripper (hand) of the robot grasps the payload, the moments of inertia of the mechanism around the joint axis increase. If during synthesis of servosystems we have not included the mass and the moment of inertia of the payload, the servosystems can become underdamped which causes oscillations in the robotic system. Thus, it is necessary to synthesize the servosystem gains taking into account the maximum payload mass and the moments of inertia that can be carried by the robot. However, if a relatively large variation of the payload is assumed, this can cause uneven performance of the servosystems. For example, when the gripper is empty, or when it carries a payload that is lighter than the maximum assumed payload, the servosystem can become very overdamped ($\xi_i >> 1$) with a resulting slow response.

This problem can be solved by introducing the variable velocity feedback gain. Such adaptive control ensures that for each payload the performance of the servosystems will be nearly equal. Such control requires that the parameters of the payload be known. However, generally, the parameters of the payload are not known, so if the control has to adapt to payload parameter variation it is necessary to ensure identification of these parameters. Identification of the payload parameters can be realized in various ways. The most efficient approach is by direct measuring of the forces at the contact points between the gripper and the payload. In these contact points force transducers are implemented that give direct information on the dynamics of the payload.[11]

Adaptive control can be introduced in various ways. If payload variation with respect to the parameters of the robot links and actuators is relatively small, then it is not often necessary to introduce adaptive control. It can be assumed that servosystems synthesized with constant feedback gains are sufficiently robust to withstand payload parameter variations. However, new robots are appearing that can carry payloads with much larger masses than the masses of the links. Obviously, for such robots the influence of payload variations can cause uneven performance of the servosystems, making it necessary to apply adaptive control. With such robots, elastic effects of the links usually appear. In previous considerations, we have assumed that all the robot's links are rigid. If the masses of the links are small with respect to the mass of the payload, the elastic effects can significantly affect system performance. The control system in these cases must be concerned not only about payload parameter variations, but also about the elastodynamic effects that can complicate control laws and their implementations. However, it should be pointed out that the problems related to computer implementation of dynamic on-line control algorithms are constantly diminishing. Thus, the present-day conclusions about these problems, like those concerning many other technological issues, are only conditionally true.

References

1. Popov, E.P, Vereschagin, F.A., and Zenkevich, S.L., Manipulation robots: *Dynamics and Algorithms* (in Russian), in Scientific Fundamentals of Robotics, Nauka, Moscow, 1978.
2. Albus, J.S., McCain, F.G., and Lumia, R., NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM), NIST Tech. Note 1235, 1989.
3. Albus, J.S., Concept for a reference model architecture for real time intelligent control, ARTICS. NIST/NBS. Tech. Note. TN-1227. U.S.A., 1990.
4. Dornier GmbH, A&R Control Development Methodology Definition Report, Doc. No. CT2/CDR/DO Issue 2.0, September 1992.
5. Vukobratović, K. M., *Applied Dynamics of Manipulation Robots: Modeling, Analysis and Examples*, Springer-Verlag, Berlin, 1989.
6. Vukobratović, K. M., *Introduction to Robotics*, Springer-Verlag, 1989.
7. Paul, P. *Robot Manipulators: Mathematics, Programming and Control*, The MIT Press, 1981.
8. Chestnut, H. and Mayer, R.W. Servomechanisms and Regulating System Design. John Wiley and Sons, New York, 1963.
9. Athans, M. and Falb, P.I., *Optimal Control: An Introduction to the Theory and its Application*, McGraw-Hill, New York, 1966.
10. Vukobratović, K. M. and Stokić, M.D., *Control of Manipulation Robots: Theory and Application*, Springer-Verlag, Berlin, 1982.
11. Vukobratović, K.M., Stokić, M.D., and Kircanski M.N., *Non-adaptive and Adaptive Control of Manipulation Robots*, Springer-Verlag, Berlin, 1985.
12. Vukobratović, K.M. and Stokić, M.D., *Applied Control of Manipulation Robots: Analysis, Synthesis and Exercises*, Springer-Verlag, Berlin, 1989.
13. Paul, R.C., Modeling, Trajectory Calculation, and Servoing of a Computer Controlled Arm, A.I. Memo 177, Stanford Artifical Intelligence Laboratory, Stanford University, September 1972, also in Russian, Nauka, Moscow, 1976.
14. Bejczy, K.A., Robot Arm Dynamic and Control, Technical Memorandum 33-669, Jet Propulsion Laboratory, February, 1974.
15. Pavlov, A.V. and Timofejev, V.A., Calculation and stabilization of programmed motion of a moving robot-manipulator, (in Russian), *Teknicheskaya Kibernetika,* 6, 91–101, 1976.
16. Raibert, H.M. and Horn, P.K.B., Manipulator control using the configuration space method, *The Industrial Robot,* 5, 2, 69–73, 1978.
17. Timofeyev, V.A. and Ekalo, V.Yu., Stability and stabilization of programmed motion of robots: manipulators, (in Russian), *Avtomatikaand Telemechanika,* 10, 148–156, 1976.