20

Robot Dynamics

Miomir Vukobratović
Mihajlo Pupin Institute

Viljko Potkonjak
University of Belgrade

We start our discussion on robot dynamics from the standpoint that successful design and control of any system require appropriate knowledge of its behavior. This is certain, but we should discuss what is meant by "appropriate knowledge." Let us consider a robot as an example of a technical system. Appropriate knowledge of its behavior may, but need not, include the mathematical model of its dynamics. In the earlier phases of robotics development, design was not based on the exact calculation of robot dynamics but followed experience from machine design. Control did not take into account many dynamic effects. Large approximations were made to reduce the problem to the well-known theory of automatic control. The undeveloped robot theory could not support a more exact approach. For a long time, the practice of robotics (design, manufacture, and implementation) grew independently of the theory that was too academic. However, this did not stop manufacturers from producing many successful robots.

Presently, the need for complex, precise, and fast robots requires a close connection between theory and practice. Regarding the application of robot dynamics, the main breakthrough was made with the development of computer-aided methods for dynamic modeling.[1-3] Such methods allowed fast and user-friendly calculations of all relevant dynamic effects. In this way dynamic modeling and simulation became the essential tools in robot design. The other possibility for application of robot dynamics is the synthesis of the so-called dynamic control.

In this subsection we first discuss the principles of dynamic modeling, the approach to the description of dynamics, and the derivation of the mathematical model. Then, special attention is

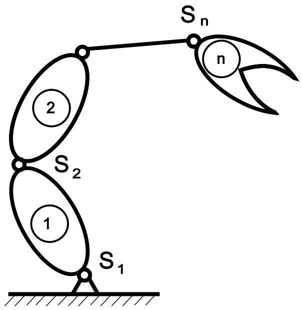


FIGURE 20.1 Robot as a simple and open chain.

paid to computer-based methods. To complete the information on robot dynamics, the mechanism model should be supplemented with the driving system model. After that, we briefly describe the application of the dynamic model. One of the promising directions is the development of CAD systems for robots. The other is dynamic control. Some extension of robot dynamics is made and discuss different effects that were not included in the initial model, trying to locate those of main importance (contact problems, elastic deformations, friction, impact, etc.).

20.1 Fundamentals of Robot Dynamic Modeling

20.1.1 Basic Ideas

From the notion *dynamic modeling* we understand the system of differential equations that describes robotic dynamic behavior. We expect the reader to possess the knowledge necessary to understand the derivation of the model. However, we will try to give enough information at an adequate level of presentation to allow readers to follow the text easily.

Here we consider a manipulation robot as an open and simple kinematic chain (as shown in Figure 20.1) consisting of n rigid bodies (robot links) interconnected by means of n one-degree-of-freedom (one-DOF) joints. A joint allows one relative rotation (revolute joint) or one relative translation (linear joint). Because the complete chain has n DOFs, its dynamics can be described by means of n differential equations of motion. They are second-order equations. This set is called the *dynamic model*.

Several approaches have been used to describe system dynamics: laws of linear momentum and angular momentum.[1-4] Lagrange's equations,[5,6] and Gauss' principle.[7,8] All approaches lead to the same dynamic model but the model formation procedure is different. Here, we use the laws of linear momentum and angular momentum. This approach is often called Newton–Euler equations. In the authors' opinion it is the most appropriate for the majority of readers.

Let us introduce one position coordinate for each joint, angle in the revolute joint, and longitudinal displacement in the linear joint. This set of coordinates uniquely describes the position of the chain. We usually call this set the *internal coordinates* (or joint coordinates, or generalized coordinates). If the coordinate for joint S_j is marked by q_j, then the complete position vector is

$$q = \begin{bmatrix} q_1 & q_2 & \cdots & q_n \end{bmatrix}^T \tag{20.1}$$

20.1.2 Robot Geometry

At this point we have to decide the mathematical presentation of robot geometry and kinematics. Up to now, two ways have been defined. One is based on the Rodrigues' formulae of finite rotation and the other uses the Denavit–Hartenberg parameters. The latter method is more widely accepted

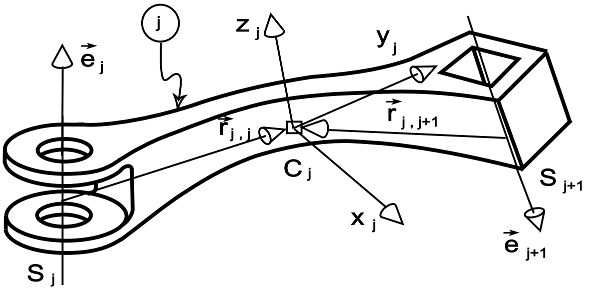


FIGURE 20.2 Geometry of a link.

because it allows simpler expression of transformation matrices and probably faster calculation of robot kinematics. However, if the intention is to discuss dynamics, it should be stressed that the first method is more appropriate. It is more general and follows the rigid body motion approach used in all standard textbooks on mechanics. For these reasons, we utilize the method based on Rodrigues' formulae.

Figure 20.2 shows one link of the robot chain, the j-th one. Joint S_j is shown as revolute and S_{j+1} as linear. To define the link geometry, it is necessary to describe the position and the orientation of the joints with respect to the mass center (MC).

The motion direction in each joint is defined by means of an axis, that is, by a unit vector. It can describe rotation or translation, depending on the type of joint. Thus $\vec{e}_j$ corresponds to joint S_j and $\vec{e}_{j+1}$ to S_{j+1}. The relative position of MC with respect to the joints is defined by means of vectors $\vec{r}_{j,j}$ and $\vec{r}_{j,j+1}$ as shown in the Figure 20.2. MC is marked by C_j.

During robot motion, positions of all links, and accordingly, geometry vectors expressed in the immobile external frame, change. However, if geometry vectors are considered relative to the corresponding link, they become constant and represent the property of the link itself. To express these constant values, we introduce a Cartesian system fixed to the link and with the origin in the MC (link-fixed frame). The axes are x_j, y_j, and z_j. The system may be oriented in an arbitrary way but is most suitable if its axes coincide with the so-called principal axes of inertia. Consider now vector $\vec{e}_j$. It can be expressed by means of three constant projections onto the axes of the frame fixed to the link j: $e_j^{x_j}$, $e_j^{y_j}$, and $e_j^{z_j}$. For this triple we introduce the notation

$$\vec{\tilde{e}}_j = \left(e_j^{x_j},\ e_j^{y_j},\ e_j^{z_j}\right) \tag{20.2}$$

The tilde "~" above the letter indicates that the vector is expressed in the link-fixed frame. Notation $\vec{e}_j$ (without tilde) denotes three projections onto the axes of an external immobile frame. If the same is applied to vectors $\vec{r}_{j,j}$ and, $\vec{r}_{j,j+1}$, two constant triples are obtained

$$\vec{\tilde{r}}_{j,j} = \left(r_{j,j}^{x_j},\ r_{j,j}^{y_j},\ r_{j,j}^{z_j}\right) \tag{20.3}$$

$$\vec{\tilde{r}}_{j,j+1} = \left(r_{j,j+1}^{x_j},\ r_{j,j+1}^{y_j},\ r_{j,j+1}^{z_j}\right) \tag{20.4}$$

Vector $\vec{e}_{j+1}$ is constant if expressed in the frame fixed to link j and a suitable notation is needed for these projections. Notation $\vec{\tilde{e}}_{j+1}$ indicates that the vector is considered relative to link $j + 1$ (analogously to relation (20.2)). Hence, a new notation is introduced to indicate the projections onto link j:

$$\vec{\underset{\sim}{e}}_{j+1} = \left(e_{j+1}^{x_j}, e_{j+1}^{y_j}, e_{j+1}^{z_j}\right) \tag{20.5}$$

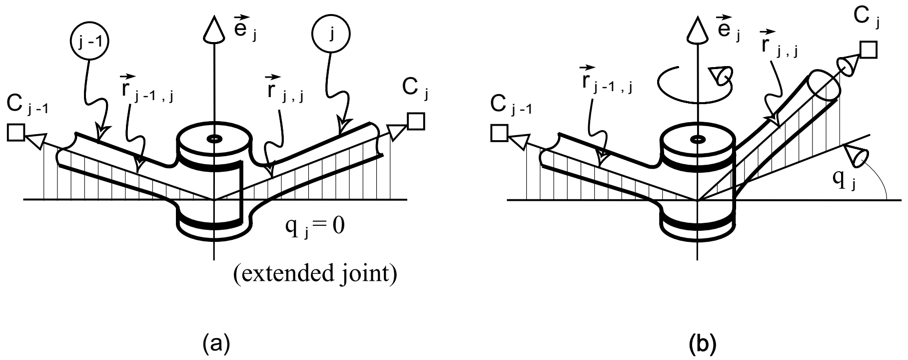


FIGURE 20.3 Definition of a coordinate in a revolute joint.

Generally, for any vector $\vec{a}_j$ having index j, the tilde "~" above the letter ($\tilde{\vec{a}}_j$) indicates the projections onto frame j, while the tilde under the letter ($\underset{\sim}{\vec{a}}_j$) indicates the projections onto the preceding frame, $j - 1$. Notation without the tilde ($\vec{a}_j$) indicates projections onto the external immobile frame.

Now, it should be stated that four vectors, $\tilde{\vec{e}}_j$, $\underset{\sim}{\vec{e}}_{j+1}$, $\tilde{\vec{r}}_{j,j}$, $\tilde{\vec{r}}_{j,j+1}$, define the geometry of link j. To define the geometry of the complete chain, one has to prescribe these four vectors for all links.

It is still necessary to distinguish between the revolute and linear joints. For this purpose we introduce the indicator s_j for each joint:

$$s_j = \begin{cases} 0, & \text{if} \quad S_j \quad \text{is a revolute joint} \\ 1, & \text{if} \quad S_j \quad \text{is a linear joint.} \end{cases} \tag{20.6}$$

Now, it is possible to define the joint coordinates more precisely. We consider the revolute joints first. If S_j is a revolute joint, then coordinate q_j represents the angle of rotation measured from the extended position. The exact definition is shown in Figure 20.3. The angle lies in a plane perpendicular to axis $\vec{e}_j$. The negative projection of $\vec{r}_{j-1,j}$ defines the extended position ($q_j = 0$) and the angle is measured to the projection of $\vec{r}_{j,j}$. Figure 20.3a shows the extended position and Figure 20.3b the rotated position.

Suppose now that joint S_j is linear. Coordinate q_j defines the length of translation along $\vec{e}_j$ and its precise definition requires previous introduction of the zero position. This zero-point can be adopted anywhere on the axis of translation. It is marked by S'_j in Figure 20.4. Once adopted, this point determines the vector $\vec{r}_{j-1,j}$. Coordinate q_j is defined as the displacement $\overline{S'_jS''_j}$ with the proper sign with respect to $\vec{e}_j$ (see Figure 20.4).

It is necessary to introduce an additional vector. $\overrightarrow{S'_jC_j} = \vec{r}'_{j,j}$. It follows from Figure 20.4. that

$$r'_{j,j} = \overrightarrow{S'_jS''_j} + \overrightarrow{S''_jC_j} = q_j\vec{e}_j + \vec{r}_{j,j} \tag{20.7}$$

or, more generally,

$$\vec{r}'_{j,j} = \vec{r}_{j,j} + s_jq_j\vec{e}_{j.} \tag{20.8}$$

For a linear joint ($s_j = 1$) relation (20.8) becomes (20.7) and for revolute joints ($s_j = 0$) it reduces to $\vec{r}'_{j,j} = \vec{r}_{j,j}$. Thus, expression (20.8), i.e., vector $\vec{r}'_{j,j}$, may replace $\vec{r}_{j,j}$ for any type of joint.

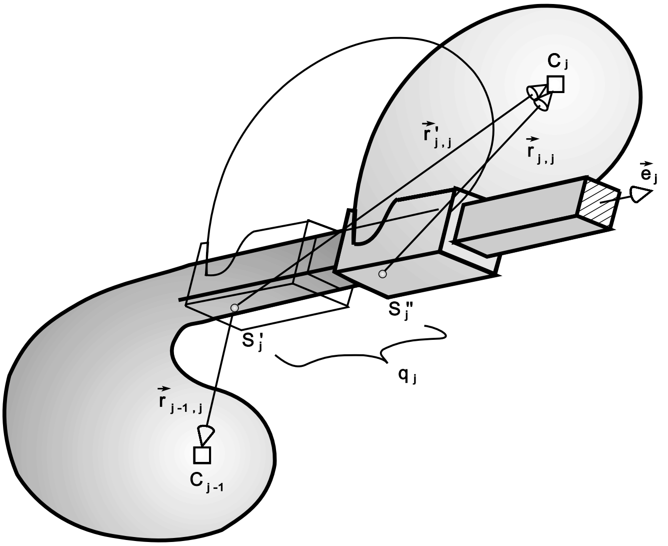


FIGURE 20.4 Definition of coordinate in a linear joint.

After introducing different frames (link-fixed and external immobile) the question arises about the possibility of transforming a vector from one frame to another. Let us consider a vector $\vec{a}_j$. For the transition between the frames, transformation matrices are applied:

From the j-th link-fixed frame to the external one

$$\vec{a}_j = A_j \vec{\tilde{a}}_{j,} \quad \dim A_j = 3 \times 3 \tag{20.9}$$

and in the opposite direction

$$\vec{\tilde{a}}_j = A_j^{-1} \vec{a}_j = A_j^T \vec{a}_j \tag{20.10}$$

From the j-th link-fixed frame to the $(j - 1)$-th one

$$\underset{\sim}{\vec{a}}_j = A_{j-1,j} \vec{\tilde{a}}_j \tag{20.11}$$

and in the opposite direction

$$\vec{\tilde{a}}_j = A_{j,j-1} \underset{\sim}{\vec{a}}_j = A_{j-1,j}^{-1} \underset{\sim}{\vec{a}}_j = A_{j-1,j}^T \underset{\sim}{\vec{a}}_j . \tag{20.12}$$

It could be noted that the matrices are orthogonal and thus the inverse equals the transpose.

A detailed explanation of how to calculate the transformation matrices is given in this chapter's appendix.

20.1.3 Equations of Dynamics

We start by considering dynamics from one of the links, the j-th one. For this purpose we fictively interrupt the chain in joints S_j and S_{j+1}. The disconnection in joint S_j is shown in Figure 20.5a. The

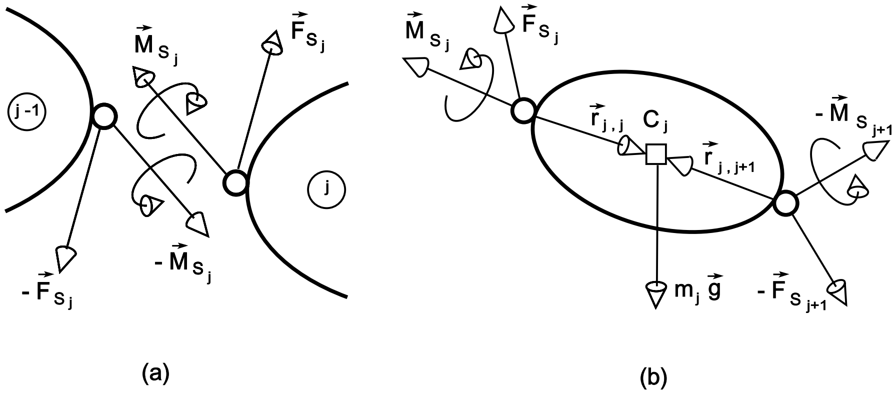


FIGURE 20.5 Extraction of one link from the chain.

mutual influence of the two links is expressed in terms of a force and a couple. $\vec{F}_{S_j}$ denotes the force and $\vec{M}_{S_j}$ denotes the moment of the couple. They act in the forward direction (from link $j - 1$ to link j) while $-\vec{F}_{S_j}$ and $-\vec{M}_{S_j}$ act in the backward direction (from j to $j - 1$).

Figure 20.5b shows the link j together with all forces and couples acting upon it. We use the law of linear momentum (Newton's law) to describe the motion of MC:

$$m_j\vec{w}_j = m_j\vec{g} + \vec{F}_{S_j} - \vec{F}_{S_{j+1}} \tag{20.13}$$

where m_j is the link mass, $\vec{w}_j$ is MC acceleration, and $\vec{g}$ is the acceleration due to gravity (9.81 m/s^2). The law expresses the equilibrium of the inertial and real forces.

Now, we discuss the rotation of the link about its MC. It can be described by the law of angular momentum (Euler's equations):

$$\tilde{J}_j\vec{\tilde{\varepsilon}}_j + \vec{\tilde{\omega}}_j \times \left(\tilde{J}_j\vec{\tilde{\omega}}_j\right) = \vec{\tilde{M}}_{S_j} - \tilde{\vec{M}}_{S_{j+1}} - \vec{\tilde{r}}'_{j,j} \times \vec{\tilde{F}}_{S_j} + \vec{\tilde{r}}_{j,j+1} \times \tilde{\vec{F}}_{S_{j+1}} \tag{20.14}$$

where $\vec{\tilde{\varepsilon}}_j$ and $\vec{\tilde{\omega}}_j$ are the angular acceleration and angular velocity. The tilde "~" indicates that the vectors are expressed in the frame fixed to the link j. $\tilde{J}_j$ is the tensor of inertia calculated for the axes of the link-fixed frame. In a general case, the tensor has the form

$$\tilde{J}_j = \begin{bmatrix} J_{xx_j} & J_{xy_j} & J_{xz_j} \\ J_{yx_j} & J_{yy_j} & J_{yz_j} \\ J_{zx_j} & J_{zy_j} & J_{zz_j} \end{bmatrix} \tag{20.15}$$

However, if the frame axes are placed to coincide with the principal inertial axes, then the tensor takes the diagonal form

$$\tilde{J}_j = \begin{bmatrix} J_{x_j} & & \\ & J_{y_j} & \\ & & J_{z_j} \end{bmatrix} \tag{20.16}$$

where J_{x_j} is the inertial moment with respect to axis x_j and analogously holds for J_{y_j} and J_{z_j}.

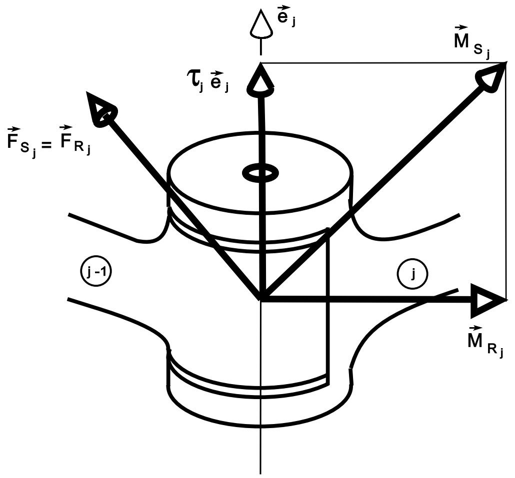


FIGURE 20.6 The total force $(\vec{F}_{S_j})$ and the total couple $(\vec{M}_{S_j})$ in a revolute joint.

It should be mentioned that Equation (20.13) is written in the external frame, while (20.14) is written in the link-fixed frame. Equation (20.13) could be simply rewritten and expressed in the link-fixed frame, while (20.14) cannot change the frame so easily. It would have to be multiplied by transformation matrix A_j.

We now discuss force $\vec{F}_{S_j}$ and couple $\vec{M}_{S_j}$ transmitted through the revolute joint S_j (Figure 20.6). First, we consider the "pure" reactions, force and couple, that follow from the mechanical connection of the two links. Because the joint permits one rotation (about $\vec{e}_j$), the reaction force $\vec{F}_{R_j}$ (from link $j - 1$ to link j) may be of arbitrary direction, while the reaction couple $\vec{M}_{R_j}$ is perpendicular to axis $\vec{e}_j$ (that is, $\vec{M}_{R_j} \perp \vec{e}_j$). In the revolute joint a driving torque τ_j acting about axis $\vec{e}_j$ exists. In the vector form, the drive is $\tau_j \vec{e}_j$. Thus, the total force and couple transmitted through the joint are

$$\vec{F}_{S_j} = \vec{F}_{R_j}$$

$$\vec{M}_{S_j} = \vec{M}_{R_j} + \tau_j \vec{e}_j. \tag{20.17}$$

Consider now a linear joint S_j (Figure 20.7). The total force $\vec{F}_{S_j}$ transmitted through the joint has two components, the reaction force $\vec{F}_{R_j}$, and the driving force $\tau_j \vec{e}_j$. The reaction is perpendicular to the axis $\vec{e}_j$ (that is, $\vec{F}_{R_j} \perp \vec{e}_j$). The total couple $\vec{M}_{S_j}$ consists of reaction only and can be of arbitrary direction. Thus, it holds that

$$\vec{F}_{S_j} = \vec{F}_{R_j} + \tau_j \vec{e}_j$$

$$\vec{M}_{S_j} = \vec{M}_{R_j} \tag{20.18}$$

Equations (20.17) and (20.18) can be written in a unique way that fits both the revolute and the linear joints:

$$\vec{F}_{S_j} = \vec{F}_{R_j} + s_j \tau_j \vec{e}_j$$

$$\vec{M}_{S_j} = \vec{M}_{R_j} + (1 - s_j)\tau_j \vec{e}_j. \tag{20.19}$$

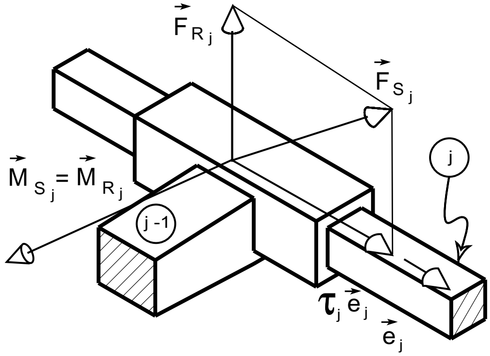


FIGURE 20.7 The total force ($\vec{F}_{S_j}$) and the total couple ($\vec{M}_{S_j}$) in a linear joint.

where s_j indicates the type of joint (see (20.6)).

Equations (20.13) and (20.14), together with (20.19), describe the dynamics of link j. If we apply these equations to all links, we arrive to the system of $2n$ vector equations describing the dynamics of the complete chain:

$$
\begin{gathered}
m_j \vec{w}_j = m_j \vec{g} + \vec{F}_{R_j} + s_j \tau_j \vec{e}_j - \vec{F}_{R_{j+1}} - s_{j+1}\tau_{j+1}\vec{e}_{j+1} \\
\tilde{J}_j \tilde{\vec{\varepsilon}}_j + \tilde{\vec{\omega}}_j \times \left(\tilde{J}_j \tilde{\vec{\omega}}_j\right) = \tilde{\vec{M}}_{R_j} + \left(1 - s_j\right)\tau_j \tilde{\vec{e}}_j - \underset{\sim}{\vec{M}}_{R_{j+1}} - \left(1 - s_{j+1}\right)\tau_{j+1} \underset{\sim}{\vec{e}}_{j+1} \\
- \tilde{\vec{r}}'_{j,j} \times \left(\tilde{\vec{F}}_{R_j} + s_j \tau_j \tilde{\vec{e}}_j\right) + \tilde{\vec{r}}_{j,j+1} \times \left(\underset{\sim}{\vec{F}}_{R_{j+1}} + s_{j+1}\tau_{j+1}\underset{\sim}{\vec{e}}_{j+1}\right) \\
j = 1, \ldots, n
\end{gathered}
\tag{20.20}
$$

Our intention is to find the dynamic model in its minimal form. It should directly connect the input (drives τ_j) and the output (motions q_j). First, it is necessary to express all kinematic variables (velocities and accelerations) in terms of the joint coordinates (q_j) and their derivatives ($\dot{q}_j, \ddot{q}_j$). After that we eliminate all the reactions, $\vec{F}_{R_j}$ and $\vec{M}_{R_j}$, $j = 1, \ldots, n$. Let us analyze the number of equations and the number of unknowns that should be eliminated. System (20.20) consists of $2n$ vector equations, that is, $6n$ scalar equations. The number of reaction vectors is also $2n$, two in each joint. However, the number of unknown scalar components for elimination is $5n$, that is, five in each joint. This is because the reaction vector with an arbitrary direction has three unknown scalar components, while the one perpendicular to the joint axis has two. Thus, after eliminating $5n$ unknowns from $6n$ equations, we obtain a system of n scalar equations that does not contain reactions, but only the drives $\tau_1, \ldots, \tau_n$. The methodology for elimination of joint reactions represents the characteristics of each method for dynamic modeling. One method will be explained in the next paragraph.

Regardless of the method chosen for dynamic modeling, a set of differential equations is obtained. Equations are linear with respect to the second derivatives:

$$
\begin{aligned}
H_{11}(q)\ddot{q}_1 \quad + \quad & \cdots + H_{1n}(q)\ddot{q}_n + h_1(q,\dot{q}) = \tau_1 \\
\vdots \quad & \\
H_{n1}(q)\ddot{q}_1 \quad + \quad & \cdots + H_{nn}(q)\ddot{q}_n + h_n(q,\dot{q}) = \tau_n.
\end{aligned}
\tag{20.21}
$$

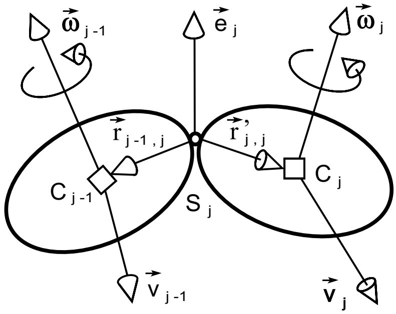


FIGURE 20.8 Velocities of robot links.

Notation $H_{ij}(q)$ indicates that the coefficient depends on all elements of the joint position vector $q = [q_1 \ \dots \ q_n]^T$, and $h_j(q, \ \dot{q})$ indicates that the free member depends on the coordinates q and their derivatives $\dot{q}$. System (20.21) may be written in matrix form

$$H(q)\ddot{q} + h(q,\dot{q}) = \tau \tag{20.22}$$

where $H = [H_{ij}]_{\dim = \, n\times n}$ is called the inertial matrix, $h = [h_j]_{\dim = \, n\times 1}$ takes into account the gravity, centrifugal, and Coriolis effects, and $\tau = [\tau_1 \ \dots \ \tau_n]^T$ is the vector of drives.

Let us discuss drives τ a little more. τ_j represents the driving torque about the joint shaft if the joint is revolute, or the force along the sliding axis if the joint is linear. In any case, the drive is produced by some actuation system (electric, hydraulic, or pneumatic) and then transmitted to the joint (by means of gears, chain, belt, etc.). Thus, τ_j is the output action of an actuator-plus-transmission assembly. This means that the previously derived model (Equation 20.22) describes only one part of robot, its mechanism, while the actuation system model is still missing.

20.2 Recursive Formulation of Robot Dynamics

In the previous paragraph we described robot dynamics by means of a set of $2n$ vector equations (model (20.20)). The model included the reaction forces and couples ($\vec{F}_{R_j}$ and $\vec{M}_{R_j}$) and their elimination was recognized as essential for the minimization of the model form. Here, we use a recursive approach to kinematics and dynamics to carry out this elimination. The method is specially suitable for the creation of a computer procedure for dynamic modeling.

20.2.1 Velocities and Accelerations of Robot Links

Position and speed define the state of each link of a kinematic chain and accordingly the state of the entire chain. The position was discussed previously and it was found that the joint coordinates $q = [q_1 \ \dots \ q_n]^T$ describe the chain position in the most appropriate way. Now, we are going to discuss speed.

If we consider the spatial motion of the link j, we find that it is characterized by the velocity of its MC (e.g., of point C_j in Figure 20.8) and the angular velocity. Let $\vec{v}_j$ and $\vec{\omega}_j$ be these velocities. Our intention is to express these quantities in terms of joint coordinates q and joint velocities $\dot{q}$. The recursive approach will be applied, and hence, we consider two links, $j - 1$ and j, interconnected by means of joint S_j (Figure 20.8).

The angular velocity is obtained by the superposition of rotations. If S_j is a linear joint (indicator $s_j = 1$), it does not contribute to rotation and if it is revolute ($s_j = 0$), its contribution is $\dot{q}_j\vec{e}_j$. Thus, we may write a general expression

$$\vec{\omega}_j = \vec{\omega}_{j-1} + \dot{q}_j(1 - s_j)\vec{e}_j. \tag{20.23}$$

showing the recursive character of angular velocity. Making the derivative of (20.23) one obtains the expression for acceleration:

$$\vec{\varepsilon}_j = \vec{\varepsilon}_{j-1} + (\ddot{q}_j \vec{e}_j + \dot{q}_j (\vec{\omega}_j \times \vec{e}_j))(1 - s_j) \tag{20.24}$$

For the MC position vector, $\vec{r}_{c_j}$, the recursive expression holds:

$$\vec{r}_{c_j} = \vec{r}_{c\,j-1} - \vec{r}_{j-1,j} + \vec{r}\,'_{j,j} \tag{20.25}$$

from which the first derivative gives MC velocities

$$\vec{v}_j = \vec{v}_{j-1} - \vec{\omega}_{j-1} \times \vec{r}_{j-1,j} + \vec{\omega}_j \times \vec{r}\,'_{j,j} + \dot{q}_j s_j \vec{e}_j \tag{20.26}$$

and the second derivative gives the MC accelerations

$$\begin{aligned} \vec{w}_j = {} & \vec{w}_{j-1} - \vec{\varepsilon}_{j-1} \times \vec{r}_{j-1,j} - \vec{\omega}_{j-1} \times (\vec{\omega}_{j-1} \times \vec{r}_{j-1,j}) + \vec{\varepsilon}_j \times \vec{r}\,'_{j,j} \\ & + \vec{\omega}_j \times (\vec{\omega}_j \times \vec{r}\,'_{j,j}) + (\ddot{q}_j \vec{e}_j + 2\dot{q}_j (\vec{\omega}_j \times \vec{e}_j)) s_j \end{aligned} \tag{20.27}$$

All the vectors in Equations (20.23) to (20.27) are expressed in the external immobile frame. Note that it is possible to transform the equations to the link-fixed frame j or $j - 1$.

We now express the velocities and accelerations of link j in terms of coordinates q and derivatives $\dot{q}$ and $\ddot{q}$. Velocities $\vec{\omega}_j$ and $\vec{v}_j$ represent linear forms with respect to joint velocities:

$$\vec{\omega}_j = \sum_{k=1}^{j} \vec{\lambda}_k^j \dot{q}_k \tag{20.28}$$

$$\vec{v}_j = \sum_{k=1}^{j} \vec{\xi}_k^j \dot{q}_k \tag{20.29}$$

where j in $\vec{\lambda}_k^j$ and $\vec{\xi}_k^j$ represents an upper index and not an exponent. Accelerations of the link are linear forms with respect to the joint accelerations:

$$\vec{\varepsilon}_j = \sum_{k=1}^{j} \vec{\lambda}_k^j \ddot{q}_k + \vec{\gamma}^j \tag{20.30}$$

$$\vec{w}_j = \sum_{k=1}^{j} \vec{\xi}_k^j \ddot{q}_k + \vec{\delta}^j \tag{20.31}$$

We now turn to matrix notation. For this reason we introduce a 3×1 matrix for each vector and use a proper notation. For instance, ω_j denotes the 3×1 matrix corresponding to vector $\vec{\omega}_j$, and analogously holds for all other vectors (we simply omit "$\rightarrow$"). Relations (20.28) to (20.31) can now be written in the form

$$\omega_j = \Lambda^j \dot{q} \tag{20.32}$$

$$v_j = \Xi^j \dot{q} \tag{20.33}$$

$$\varepsilon_j = \Lambda^j \ddot{q} + \Gamma^j \tag{20.34}$$

$$w_j = \Xi^j \ddot{q} + \Delta^j \tag{20.35}$$

where Λ^j and Γ^j are $3 \times n$ and 3×1 matrices (respectively) containing the coefficients and the free member of the linear forms (20.28) and (20.30):

$$\Lambda^j = [\lambda_1^j \cdots \lambda_j^j 0 \cdots 0]_{\dim = 3\times n} \tag{20.36}$$

$$\Gamma^j = [\gamma^j]_{\dim = 3\times 1} \tag{20.37}$$

and Ξ^j and Δ^j are $3 \times n$ and 3×1 matrices that contain the coefficients and the free member of (20.29) and (20.31):

$$\Xi^j = [\xi_1^j \ldots \xi_j^j 0 \ldots 0]_{\dim = 3\times n} \tag{20.38}$$

$$\Delta^j = [\delta^j]_{\dim = 3\times 1} \tag{20.39}$$

Matrices Λ, Γ, Ξ, and Δ are very often written without the upper index j. This is because in software realization one variable is used for each matrix and it is modified when a new index j is considered. Thus, j is the index of iteration. Now, let us analyze how matrices Λ, Γ, Ξ and Δ change when j increases. Starting from (20.23) to (20.27), we find that when passing from $j - 1$ to j, the following modifications of the matrices are needed:

$$\begin{cases} \vec{\lambda}_k^j = \vec{\lambda}_k^{j-1}, k = 1,\ldots, j-1 \\ \vec{\lambda}_j^j = (1 - s_j)\vec{e}_j \end{cases} \tag{20.40}$$

$$\vec{\gamma}^j = \vec{\gamma}^{j-1} + \dot{q}_j(\vec{\omega}_j \times \vec{e}_j)(1 - s_j) \tag{20.41}$$

$$\begin{cases} \vec{\xi}_k^j = \vec{\xi}_k^{j-1} - \vec{\lambda}_k^{j-1} \times \vec{r}_{j-1,j} + \vec{\lambda}_k^j \times \vec{r}'_{j,j}, k = 1,\ldots, j-1 \\ \vec{\xi}_j^j = \vec{e}_j s_j + \vec{\lambda}_j^j \times \vec{r}'_{j,j} \end{cases} \tag{20.42}$$

$$\vec{\delta}^j = \vec{\delta}^{j-1} - \vec{\gamma}^{j-1} \times \vec{r}_{j-1,j} + \vec{\gamma}^j \times \vec{r}'_{j,j} - \vec{\omega}_{j-1} \times (\vec{\omega}_{j-1} \times \vec{r}_{j-1,j}) + \vec{\omega}_j \times (\vec{\omega}_j \times \vec{r}'_{j,j}) + 2\vec{\omega}_j \times \vec{e}_j s_j \dot{q}_j \tag{20.43}$$

Thus, it is possible to form matrices Λ, Γ, Ξ, and Δ in a recursive manner. In each iteration, a new link is added to the chain (e.g., link j). Transformation matrix A_j is calculated to allow expressing the geometry vectors in the external frame. Now, applying the recursive expressions (20.40) to (20.43), matrices Λ^j, Γ^j, $\Xi,^j$ and Δ^j are found starting from Λ^{j-1}, Γ^{j-1}, Ξ^{j-1}, and Δ^{j-1}.

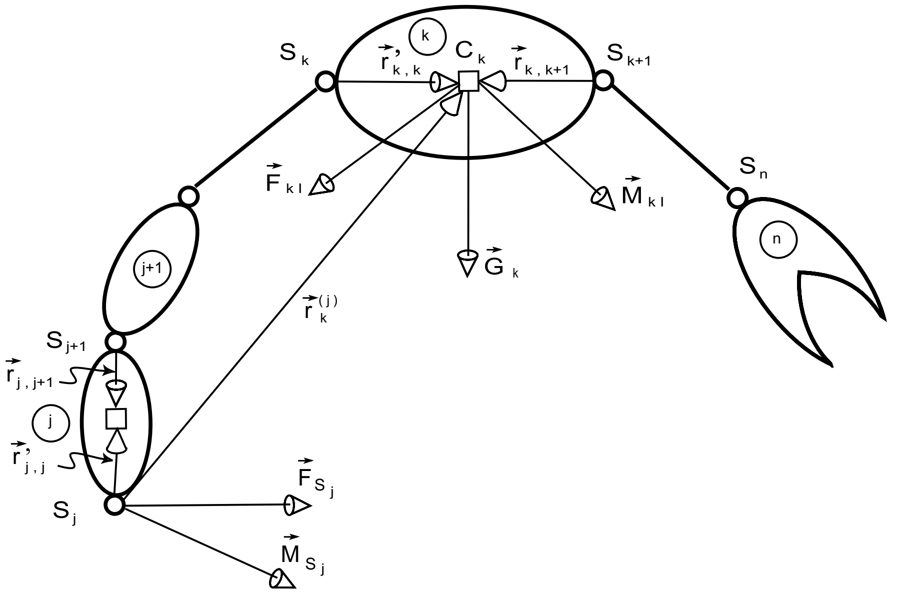


FIGURE 20.9 A part of the robot chain.

20.2.2 Elimination of Reactions — Minimization of Dynamic Model Form

Dynamics of the robot chain was described by means of $2n$ vector Equations (20.20). It was stressed that reactions in joints should be eliminated to reduce the dynamic model to its minimal form (20.21), that is, to n scalar equations. Here, we describe the recursive procedure for the elimination. The procedure begins by considering a part of the kinematic chain, from a selected joint to the free end (Figure 20.9). Let the selected joint be S_j. We now consider the subchain that consists of the links $j, j+1, \ldots, n$. The fictive break is made in joint S_j and the influence of the preceding link is expressed in terms of force $\vec{F}_{S_j}$ and couple $\vec{M}_{S_j}$. Now, we apply D'Alambert's principle and establish the equilibrium of the real and inertial forces acting on the considered part of the chain. The system of forces includes gravity forces, inertial forces, joint force $\vec{F}_{S_j}$, and joint couple $\vec{M}_{S_j}$.

The inertial load of a link (let it be link k) is distributed all over it, but can be reduced to a resultant inertial force acting at the MC, and a resultant inertial couple. The force is

$$\vec{F}_{kI} = -m_k \vec{w}_k \tag{20.44}$$

and acts in point C_k; m_k is the link mass. The moment of the couple can be expressed in Euler's form

$$\tilde{\vec{M}}_{kI} = -\left(\tilde{J}_k \tilde{\vec{\varepsilon}}_k + \tilde{\vec{\omega}}_k \times \left(\tilde{J}_k \tilde{\vec{\omega}}_k\right)\right) \tag{20.45}$$

where $\tilde{J}_k$ is the tensor of inertia. Couple (20.45) is expressed in the link-fixed frame and transformation matrix A_k could be used to transfer it to the external frame: $\vec{M}_{kI} = A_k \tilde{\vec{M}}_{kI}$. The gravity load of the link is

$$\vec{G}_k = m_k \vec{g} \ . \tag{20.46}$$

where $\vec{g}$ denotes the gravity acceleration.

D'Alambert's equilibrium concerns $\vec{F}_{kI}, \vec{M}_{kI}, \vec{G}_{k,}\ k = j, \ldots n, \vec{F}_{S_j}, \vec{M}_{S_j}$. Equilibrium of forces gives

$$\vec{F}_{S_j} = -\sum_{k=j}^{n}\left(\vec{G}_k + \vec{F}_{kI}\right) \tag{20.47}$$

Equilibrium of moments of forces is found with respect to point S_j, yielding

$$\vec{M}_{S_j} = -\sum_{k=j}^{n}\left(\vec{M}_{kI} + \vec{r}_k^{\,(j)} \times \left(\vec{G}_k + \vec{F}_{kI}\right)\right) \tag{20.48}$$

where

$$\vec{r}_k^{\,(j)} = \overrightarrow{S_jC_k} = \sum_{p=j}^{k-1}\left(\vec{r}\,'_{p,p} - \vec{r}_{p,p+1}\right) + \vec{r}\,'_{k,k} \tag{20.49}$$

Suppose now that joint S_j is linear (indicator $s_j = 1$), as shown in Figure 20.7. According to (20.18), force $\vec{F}_{S_j}$ consists of the driving component $\tau_j\vec{e}_j$ acting along the joint axis and the reaction force $\vec{F}_{R_j}$ perpendicular to that axis. To eliminate the reaction, we multiply Equation (20.47) by the joint axis vector $\vec{e}_j$, obtaining the scalar expression for the driving force

$$\tau_j = \vec{F}_{S_j}\vec{e}_j = \sum_{k=j}^{n} m_k\left(\vec{w}_k - \vec{g}\right)\vec{e}_j. \tag{20.50}$$

where (20.44) and (20.46) are substituted.

In the other case we suppose that S_j is a revolute joint (indicator $s_j = 0$) as shown in Figure 20.6. According to (20.17), the moment of couple $\vec{M}_{S_j}$ consists of the driving torque $\tau_j\vec{e}_j$ acting about the joint axis, and the reaction couple $\vec{M}_{R_j}$ perpendicular to the axis. Multiplication of Equation (20.48) by axis vector $\vec{e}_j$ eliminates the reaction, thus giving the scalar expression for the driving torque

$$\tau_j = \vec{M}_{S_j}\vec{e}_j = \sum_{k=j}^{n}\left(A_k\left(\tilde{J}_k\vec{\tilde{\varepsilon}}_k + \vec{\tilde{\omega}}_k \times \left(\tilde{J}_k\vec{\tilde{\omega}}_k\right)\right) + \vec{r}_k^{\,(j)} \times m_k\left(\vec{w}_k - \vec{g}\right)\right)\vec{e}_j\,. \tag{20.51}$$

where expression (20.45) multiplied by A_k, and expressions (20.46) and (20.44) are substituted.

To summarize, in the case of a linear joint, the drive is expressed in form (20.50) and for a revolute joint in form (20.51). This can be applied to joints $j = 1, \ldots, n$. So, dynamics of the robot chain is described by means of n scalar equations expressing the drives. In relations (20.50) and (20.51), MC accelerations, $\vec{w}_k$, and angular accelerations, $\vec{\varepsilon}_k$, appear. These accelerations can be expressed in terms of joint accelerations by using linear forms (20.34) and (20.35). In this way (20.50) is transformed to

$$\tau_j = \sum_{k=j}^{n} e_j^T m_k\left(\Xi^k\ddot{q} + \Delta^k - g\right). \tag{20.52}$$

where vector notation is replaced by 3 × 1 matrices (omitting "→"). Further transformation yields

$$\tau_j = H'_j\ddot{q} + h'_j \tag{20.53}$$

where

$$H'_j = \sum_{k=j}^{n} e_j^T m_k \Xi^k, \quad h'_j = \sum_{k=j}^{n} e_j^T m_k \left(\Delta^k - g\right) \tag{20.54}$$

are of dimensions $1 \times n$ and 1×1, respectively.

If (20.51) is transformed to depend on joint accelerations (introducing (20.34) and (20.35)), one obtains

$$\tau_j = \sum_{k=j}^{n} e_j^T A_k \left(\tilde{J}_k A_k^{-1}\left(\Lambda^k \ddot{q} + \Gamma^k\right) + \underline{\underline{\tilde{\omega}}}_k \tilde{J}_k \tilde{\omega}_k\right) + e_j^T \underline{\underline{r}}_k^{(j)} m_k \left(\Xi^k \ddot{q} + \Delta^k - g\right) \tag{20.55}$$

where for any vector, e.g., $\vec{a}$, notation $\underline{\underline{a}}$ understands the matrix

$$\underline{\underline{a}} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \tag{20.56}$$

that serves to perform the vector product in matrix form. Further transformation of (20.55) yields

$$\tau_j = H''_j \ddot{q} + h''_j \tag{20.57}$$

where

$$H''_j = \sum_{k=j}^{n} e_j^T \left(A_k \tilde{J}_k A_k^{-1} \Lambda^k + \underline{\underline{r}}_k^{(j)} m_k \Xi^k\right),$$

$$h''_j = \sum_{k=j}^{n} e_j^T \left(A_k \tilde{J}_k A_k^{-1} \Gamma^k + A_k \underline{\underline{\tilde{\omega}}}_k \tilde{J}_k \tilde{\omega}_k + \underline{\underline{r}}_k^{(j)} m_k \left(\Delta^k - g\right)\right) \tag{20.58}$$

are of dimensions $1 \times n$ and 1×1, respectively.

In the described way we have found the expressions for the joint drives in terms of joint accelerations. For a linear joint, form (20.53), along with (20.54), holds while for a revolute joint form (20.57), along with (20.58), applies. For the complete chain (all joints), the following matrix relation can be written

$$\tau = H\ddot{q} + h \tag{20.59}$$

where

$$\tau = \begin{bmatrix} \tau_1 \\ \vdots \\ \tau_j \\ \vdots \\ \tau_n \end{bmatrix}, \quad H = \begin{bmatrix} H_1 \\ \vdots \\ H_j \\ \vdots \\ H_n \end{bmatrix}, \quad h = \begin{bmatrix} h_1 \\ \vdots \\ h_j \\ \vdots \\ h_n \end{bmatrix}, \quad (H_j \text{ and } h_j) = \begin{cases} H'_j \text{ and } h'_j, & \text{for } s_j = 1 \\ H''_j \text{ and } h''_j, & \text{for } s_j = 0 \end{cases} \tag{20.60}$$

are of dimensions: τ $(n \times 1)$, H $(n \times n)$, h $(n \times 1)$. Thus, we have come to the previously stated form of the dynamic model, that is, Equation (20.22). Expressions (20.54) and (20.58) offer the possibility of recursive calculation of matrices H and h.

20.2.3 Calculation of Direct and Inverse Dynamics

The notions of *direct* and *inverse dynamics* are introduced in theoretical mechanics when considering the behavior of a mechanical system under the action of forces. The direct dynamic problem understands that system motion is known and the forces that cause this motion are to be calculated. The opposite problem, calculation of motion for the given forces, is called inverse dynamics. One should note that some authors interchange these terms. Although one might find some justification for this, we keep the definition as given above because "our" inverse dynamics needs the inversion of the inertial matrix, thus making a suitable association. For a robotic chain, direct dynamics means calculating drives τ for the prescribed robot motion $q(t)$, while inverse dynamics understands the calculation of motion q for the given drives τ. Model (20.22) offers the possibility of solving both problems. Here, we give a general approach to problem solution. Details cannot be discussed, because they depend on the used method. For instance, if the dynamic model was found in its symbolic form, the procedure for the solution would differ to some extent from the procedure used with numerical models.

Consider the direct dynamics first. When we say that the motion $q(t)$ is known, it means that velocity $\dot{q}(t)$ and acceleration $\ddot{q}(t)$ are known, too. Hence, direct dynamics generally involves derivation. Note that sometimes the motion is given by directly prescribing the acceleration. This is the case if a trapezoidal velocity profile is required (constant acceleration is followed by a uniform motion, and finally, constant deceleration stops the system). In such cases the solution includes integration to find the velocity and the position. In any case, relation (20.22) is used to calculate the drives τ. Calculation of direct dynamics is needed for several reasons. First, it is possible to obtain the preliminary information about the drive and power requirements for different robot tasks without an actual experiment. This is important in the process of robot design and enables the development of CAD systems for robots. Second, direct dynamics is used to create dynamic control. It leads to the introduction of feedforward to the robot control scheme.

Inverse dynamics understands the integration of the dynamic model (20.22). Initial state (position $q(0)$ and velocity $\dot{q}(0)$) are needed. Most standard procedures for numerical integration require the differential equations systems be written in a canonical form. Hence, model (20.22) is rewritten:

$$\frac{d}{dt}\begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \dot{q} \\ H^{-1}(q)(\tau - h(q,\dot{q})) \end{bmatrix} \tag{20.61}$$

The calculation of inverse dynamics plays the central role in any simulation system. This fact sufficiently explains the great importance of this problem.

The discussion in Section 20.2 proves that the computer can successfully be used for forming and solving the robot dynamic model. This conclusion is important because any attempt to write the dynamic model of a multiple joint system "by hand" would probably lead to numerous errors. This is because of extreme complexity of the equations that describe the dynamics of spatial multi-joint chains. The research efforts in the field of computer-aided dynamic modeling have resulted in several commercially available program packages.[9,10]

20.3 Complete Model of Robot Dynamics

In the previous paragraphs discussion on dynamics was restricted to robot mechanisms. Dynamics was understood as a relation between joint drives τ and joint motions q. Now, we note that with a real robotic system, joint torques (or forces) cannot be considered as control variables. Some actuators produce them and the actuators introduce their own dynamics. Each actuator has an input control variable and the drive it produces depends on the control as well as on the dynamic behavior of the entire system. Hence, we may say that the joint torques (or forces) τ describe the interaction between the two subsystems that form a robot: the chain and the actuators.

To describe the dynamics of the entire system, it is necessary to discuss the possible actuators. A completely general discussion that would cover all the possible types of actuators would be very extensive, because robots can be equipped with many types of different actuators: DC motors, synchronous AC motors, stepping motors, different electrohydraulic actuators, and pneumatic systems. Our intention is to show the modeling methodology for the dynamics of the entire robot. For this reason we first choose one type of actuator, a DC motor. After completing the model for this particular case, we make some generalizations to cover different types of actuators. More detailed discussion on robot driving system is given in Chapter 21.

20.3.1 Dynamic Model of a DC-Driven Robot

Permanent magnet DC motors are very common actuators with robotic systems. Their main advantage is simple control by varying the input voltage. The main disadvantage, however, follows from graphite brushes. Some other problems that are common characteristics of all electric drives should be mentioned. It is a fact that they rotate fast and produce relatively small torques. Thus, a gear-box is usually needed to reduce speed and increase torque. Further, motors are very often displaced from joints and moved toward the robot base to unload the arm statically. Hence, different transmissions (chains, belts, gears, shafts, etc.) are needed. This complicates robot construction and may influence the accuracy of motion. However, good controllability still makes DC motors very popular.

The dynamics of a DC motor that drives a robot joint S_j is described by the following relations expressing the mechanical and electrical equilibrium:

$$J_j\ddot{\theta}_j = C_{M_j} i_j - B_j\dot{\theta}_j - M_j \tag{20.62}$$

$$u_j = R_j i_j + L_j \frac{di_j}{dt} + C_{E_j}\dot{\theta}_j \tag{20.63}$$

where θ_j is the angle of the motor shaft rotation, i_j is the armature current, M_j is the output torque, and u_j is the input voltage. Motor parameters are J_j the rotor moment of inertia; C_{M_j} and C_{E_j} the constants of torque and counter electromotive force; B_j, the viscous friction coefficient; R_j, the armature resistance; and L_j, inductivity. The dynamic equations can be united to obtain a more compact canonical form:

$$\dot{x}_j = C_j x_j + f_j M_j + d_j u_j \tag{20.64}$$

where the state vector x_j has dimension three. The state vector and the system matrices are

$$x_j = \begin{bmatrix} \theta_j \\ \dot{\theta}_j \\ i_j \end{bmatrix}, \quad C_j = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -B/J & C_M/J \\ 0 & -C_E/L & -R/L \end{bmatrix}, \quad f_j = \begin{bmatrix} 0 \\ -1/J \\ 0 \end{bmatrix}, \quad d_j = \begin{bmatrix} 0 \\ 0 \\ 1/L \end{bmatrix}. \tag{20.65}$$

where, for simplicity, index j is omitted from the elements of the system matrices.

If inductivity L is small enough (it is a rather common case), the term Ldi/dt can be neglected. Equation (20.63) now becomes

$$u_j = R_j i_j + C_{E_j} \dot{\theta}_j \tag{20.66}$$

and the number of state variables reduces to two. The state vector and the system matrices in Equation (20.64) are

$$x_j = \begin{bmatrix} \theta_j \\ \dot{\theta}_j \end{bmatrix}, \quad c_j = \begin{bmatrix} 0 & 0 \\ 0 & -C_M C_E / RJ - B / J \end{bmatrix}, \quad f_j = \begin{bmatrix} 0 \\ -1/J \end{bmatrix}, \quad d_j = \begin{bmatrix} 0 \\ C_M/RJ \end{bmatrix}. \tag{20.67}$$

It was stated that DC motors are usually followed by some transmission system. The transmission defines the relation between the joint coordinate q_j and the motor coordinate θ_j. If the transmission is considered ideal (no backlash, no elastic deformation), the transmission ratio is constant:

$$q_j = \theta_j / N_j \tag{20.68}$$

If the dynamics of the transmission system (inertial properties) and the loss due to friction are neglected, it holds that

$$\tau_j = M_j N_j \tag{20.69}$$

If friction has to be discussed, it is considered through power loss. The efficiency coefficient is introduced: $0 < \eta_j < 1$. Now, Equation (20.69) is modified. If the motion is in the direction of the drive, $N_j \eta'_j$ is used instead of N_j. However, if the motion is opposite to the action of the drive, then N_j / η''_j applies. Note that η'_j and η''_j are generally different. The efficiency of a gear-box in the reverse direction is smaller ($\eta''_j < \eta'_j$).

The dynamics of the robot chain was discussed in Sections 20.1 and 20.2. It was described by means of n scalar Equations (20.21) or by means of the matrix relation (20.22). If the chain is considered as one subsystem of the robot and the actuators as the other, then the complete dynamics can be described by combining the two models: (20.21) for the chain and (20.62), (20.63) for the motors. To simplify the formulation and stress the main dynamic effects, we neglect the inductivity L and friction B (real numerical values justify this approximation). Equations (20.62) and (20.63) now yield

$$J_j \ddot{\theta}_j = \frac{C_{M_j}}{R_j} u_j - \frac{C_{M_j} C_{E_j}}{R_j} \dot{\theta}_j - M_j \tag{20.70}$$

Relations (20.68) and (20.69) that describe the transmission are needed to connect the motor variables and the joint ones. The motor variables in Equation (20.70), θ and M, are replaced by the joint variables, q and τ. The torque τ is then substituted from such a modified relation (20.70) into system (20.21) thus yielding the model

$$\begin{aligned} &H_{j1}\ddot{q}_1 + \ldots + \left(H_{jj} + J_j N_j^2\right)\ddot{q}_j + \ldots + H_{jn}\ddot{q}_n + h_j + \left(C_{M_j} C_{E_j} / R_j \right) N_j^2 \dot{q}_j \\ &= \left(N_j C_{M_j} / R_j \right) u_j, \; j = 1, \ldots, n \end{aligned} \tag{20.71}$$

or in the matrix form

$$H^*(q)\ddot{q} + h^*(q, \dot{q}) = Du \tag{20.72}$$

where

$$H^*_{kj} = \begin{cases} H_{kj}, \text{ for } k \neq j, \\ H_{jj} + J_j N_j^2, \text{ for } k = j, \end{cases}, \quad h^*_j = h_j + \left(C_{M_j} C_{E_j} / R_j\right) N_j^2 \dot{q}_j, \quad D = diag\left[N_j C_{M_j} / R_j \right] \tag{20.73}$$

By comparing models (20.21) and (20.71) one may conclude that the introduction of motor dynamics results in increased proper inertia of the joint (diagonal coefficient H_{jj} is augmented) while dynamic coupling between joints remains the same (nondiagonal coefficients H_{kj} do not change).

20.3.2 Generalized Form of the Dynamic Model

Here we present the procedure for obtaining the complete dynamic robot model for any kind of actuators.[11] The only restriction is that the actuator can be described by a linear model. Let the dynamics of the robot chain be described by model (20.22) and let the actuator for joint S_j have a linear model of the form

$$\dot{x}_j = C_j x_j + f_j M_j + d_j u_j \tag{20.74}$$

where x_j is a state vector of dimension n_j (e.g., $n_j = 3$ for a DC motor as shown in Equation (20.65)), M_j is the output torque, and u_j is the control variable. Note that u_j is subject to the saturation-type constraint: $-u_{j_{max}} < u_j < u_{j_{max}}$.

To simplify the derivation we do not discuss the transmission, that is, we assume a direct connection between the actuators and the joints. In this case, the motion of a joint and that of the corresponding actuator are equal and both are defined by means of q_j, and the torques M_j and τ_j (motor and joint) coincide. This simplification does not compromise the generality of presentation, because the transmission ratio can easily be incorporated when needed.

Let the dynamic model of the chain (Equation (20.22)) be rewritten in a canonical form, according to (20.61):

$$\frac{d\zeta}{dt} = \begin{bmatrix} \zeta_2 \\ H^{-1}(\zeta_1)(\tau - (\zeta_1, \zeta_2)) \end{bmatrix} \tag{20.75}$$

where $\zeta_1 = q$, $\zeta_2 = \dot{q}$, and $\zeta = \left[\zeta_1^T \zeta_2^T\right]^T$. Thus, ζ is the column vector defining the state of the chain and has the dimension $2n$. This equation can further be rewritten:

$$\dot{\zeta} = K(\zeta) + V(\zeta)\tau \tag{20.76}$$

where

$$K(\zeta) = \begin{bmatrix} \zeta_2 \\ -H^{-1}(\zeta_1) h(\zeta_1, \zeta_2) \end{bmatrix}, \quad V(\zeta) = \begin{bmatrix} 0 \\ -H^{-1}(\zeta_1) \end{bmatrix} \tag{20.77}$$

Let k_j elements of vector x_j coincide with elements of ζ, that is, k_j state coordinates of the j-th actuator are already included in the state vector of the chain. Joint position q_j and joint velocity $\dot{q}_j$ are usually part of the state vector x_j. This means that $k_j = 2$, and summation over all joint actuators, $\sum_{j=1}^{n} k_j = 2n$, covers the complete vector ζ. Note that in a general case q_j and $\dot{q}_j$ need not directly be the elements of x_j. Instead, linear or nonlinear dependence may exist between q_j, $\dot{q}_j$ and some elements of x_j.

The dynamics of all actuators can be described if Equations (20.74), for $j = 1, \ldots, n$, written in the compact form:

$$\dot{x} = Cx + F\tau + Du \tag{20.78}$$

where vector $x = [x_1^T \ldots x_n^T]^T$, of dimension $N = \sum_{j=1}^{n} n_j$, defines the state of the entire system.

Vector $\tau = [\tau_1 \ldots \tau_n]^T$ contains the drives, and vector $u = [u_1 \ldots u_n]^T$ the control inputs. In addition, $C = diag[C_j]$, $F = diag[f_j]$, and $D = diag[d_j]$.

Now, we are going to unite the model of the chain, Equation (20.22), and the model of the actuators, Equation (20.78). Let us introduce the matrix T_j of dimension $1 \times n_j$ such that $\ddot{q}_j = T_j \dot{x}_j$. For instance, $T_j = [010]$ for the DC drive that has the state $x_j = [q_j \dot{q}_j i_j]^T$. Model (20.22) can now be rewritten:

$$\tau = HT\dot{x} + h \tag{20.79}$$

where $T = diag[T_j]$ is an $n \times N$ matrix. Substituting $\dot{x}$ from (20.78) into (20.79) one obtains

$$\tau = (E_n - HTF)^{-1} \; (HT(Cx + Du) + h) \tag{20.80}$$

where E_n is the n-dimensional unit matrix. After substituting τ from (20.80) into (20.78), the complete robot dynamics model is obtained in the form

$$\dot{x} = \hat{C}(x) + \hat{D}(x)u. \tag{20.81}$$

The system matrices are

$$\hat{C} = Cx + F(E_n - HTF)^{-1}(HTCx + h), \quad \hat{D} = D + (E_n - HTF)^{-1} HTD \tag{20.82}$$

with dimensions $N \times 1$ and $N \times n$, respectively.

It is clear that the new form of robot dynamics model requires reformulation of the direct and the inverse problem. The direct dynamics understands the calculation of the control u that is needed to produce the prescribed robot motion. Inverse dynamics means the solution of motion for the prescribed control inputs. This latter problem is called simulation.

20.4 Some Applications of Computer-Aided Dynamics

The formulation of computer procedures for modeling robot kinematics and dynamics made possible the implementation of robot theory for practical work in design and control problems. Before that, handwritten dynamics were limited to simple cases and thus could not be successfully applied. Computer-aided kinematics and dynamics can be used to derive more sophisticated control algorithms and, on the other hand, to assist in robot design. Chapter 21 is devoted to the problem of robot design. Control issues are elaborated in Chapter 22. A survey of advanced results in these fields is given there. Hence, we briefly present only some ideas and the principal references. For historical reasons, we discuss design issues first and then control.

20.4.1 Dynamics and Robot Design

Computer-aided kinematics enables the transformation of robot coordinates from internal (joints) to external (end-effector) and vice versa. Computer procedures to calculate dynamics solve the direct and inverse problems. The direct procedure starts from the prescribed motion and calculates torques and control inputs. These are primary results, but a lot of additional characteristics can be found, too. Let us discuss these characteristics briefly.

Based on the dynamic equations for robot links, it is possible to find the vectors of reactions in robot joints. Further, the distribution of forces along each link can be solved. This result enables calculation of the mechanical stresses and elastic deformation (bending and torsion of links). For such calculation, some supplementary dynamic blocks (approximate or exact) would be needed to accomplish the model explained here. These possibilities show that robot dynamics can be successfully applied in the design of robot mechanical structure (geometry, dimensions, cross sections, choice of materials, etc.).[12-14]

Calculation of robot dynamics offers a lot of results useful for choosing the appropriate drives and the design of control systems. In addition to the torques, we can compute the power and energy requirements, form the diagrams speed vs. torque, etc.[13,14] Supplementary software calculates motor heating.[15] Because the dynamic model relates motion and driving input, it can be used to synthesize the control parameters (e.g., the feedback gains).

The variables calculated for nonperturbed mode by means of the direct dynamic procedure can be recalculated by simulating the perturbed mode (inverse dynamic procedure). In this way, it is very easy to prescribe a robot configuration and a task and then examine robot behavior by computing the different dynamic characteristics. We call this *dynamic analysis*. The software realization of such a procedure represents a very useful tool in the robot design process. A designer can quickly check a large number of different configurations. He or she can vary parameters to see their influence on some dynamic characteristics. This is of considerable help in fast and successful design. The next step is made if the limits that we impose in the design (e.g., maximal elastic deformation) are given to the computer and the software package checks the calculated characteristics against these limits. If a test is negative, some expert system might suggest how to change the relevant constructive parameters. In Vukobratović and Potkonjak[14] this approach is called the *interactive design system*. The final step is formulation of the complete CAD system for robots. The algorithms and appropriate software that would automatically find the optimal robot parameters based on the required performances and the imposed limits should be derived. The criterion of optimality is needed as well as optimization techniques. Certain results in this direction are currently available.[14,16]

20.4.2 Dynamics in On-Line Control

During the early stages of robot theory, an idea to formulate the control algorithm that takes care of robot dynamics, the so-called *dynamic control* appeared.[11] For a long time this idea was just a theoretical possibility. On one hand, the calculation of dynamics was not possible in real time, and on the other, the relatively low speed and acceleration required did not justify dynamic control. However, with current very fast and precise robots, implementation of dynamics in the control algorithm becomes necessary. It is used to form the feedforward control signal. Figure 20.10 shows such control schemes. In case (a) the feedforward signal (nominal control u^*) is found based on referent (prescribed) motion. If the motion is completely defined in advance, it is possible to calculate the direct dynamics offline and store the nominal control in the computer memory. It is recalled when the robot starts to execute the task. If referent motion is defined online (e.g., guided by means of a joystick or a sensor), on-line computation of nominal control is necessary. In case (b), the feedforward signal is found on the basis of a real state (measured data). Such a scheme understands online calculation of robot dynamics. The quality of the dynamic model (effects that are included) is relevant for the quality of robot control.

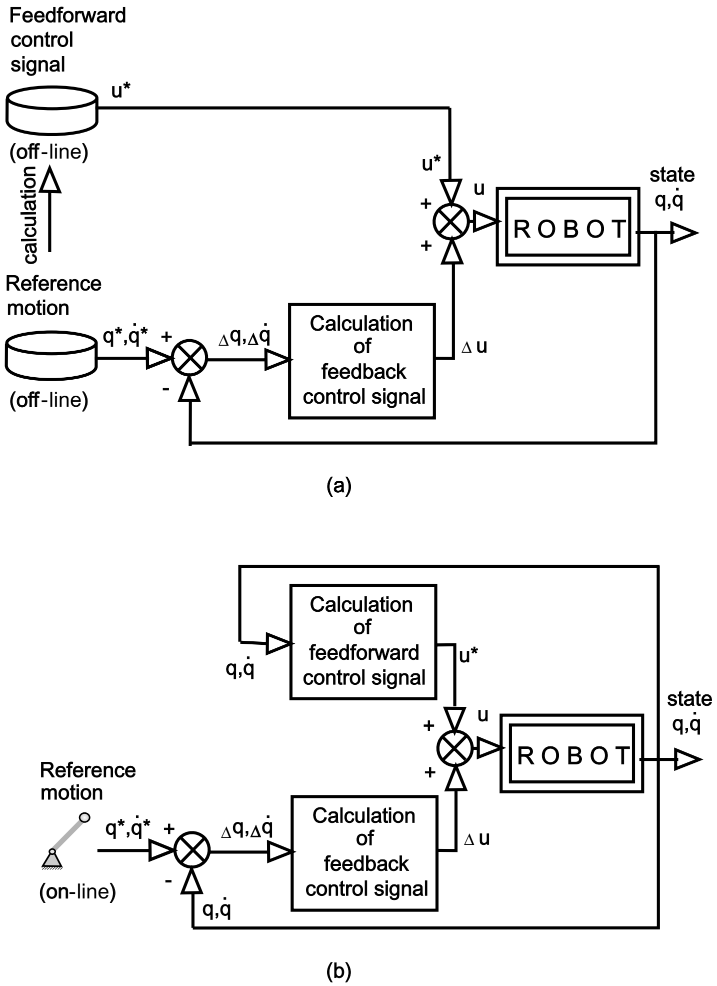


FIGURE 20.10 Scheme of dynamic control with feedforward signal.

20.5 Extension of Dynamic Modeling — Some Additional Dynamic Effects

Our purpose is to explain some dynamic effects that are not usually discussed in books on general robotics. However, these effects could be very important in some theoretical and practical work on control and design. We start with a review of the problems significant in robot dynamics and indicate the sources of particular effects. Next, some of the problems are selected and explained in more detail.

20.5.1 Robot Dynamics — Problems and Research

Beginning research in robot dynamics dealt with robots represented by an open kinematic chain. Robot links, as well as all transmission elements (shafts, gears, etc.), were considered nondeformable (Problem 1 in Figure 20.11). Many authors have worked in this field and we mention only few early results.[1-8] This approach covered many important dynamic effects and for a long time discussions on robot dynamics were restricted to such problems. This is still the case with most textbooks.

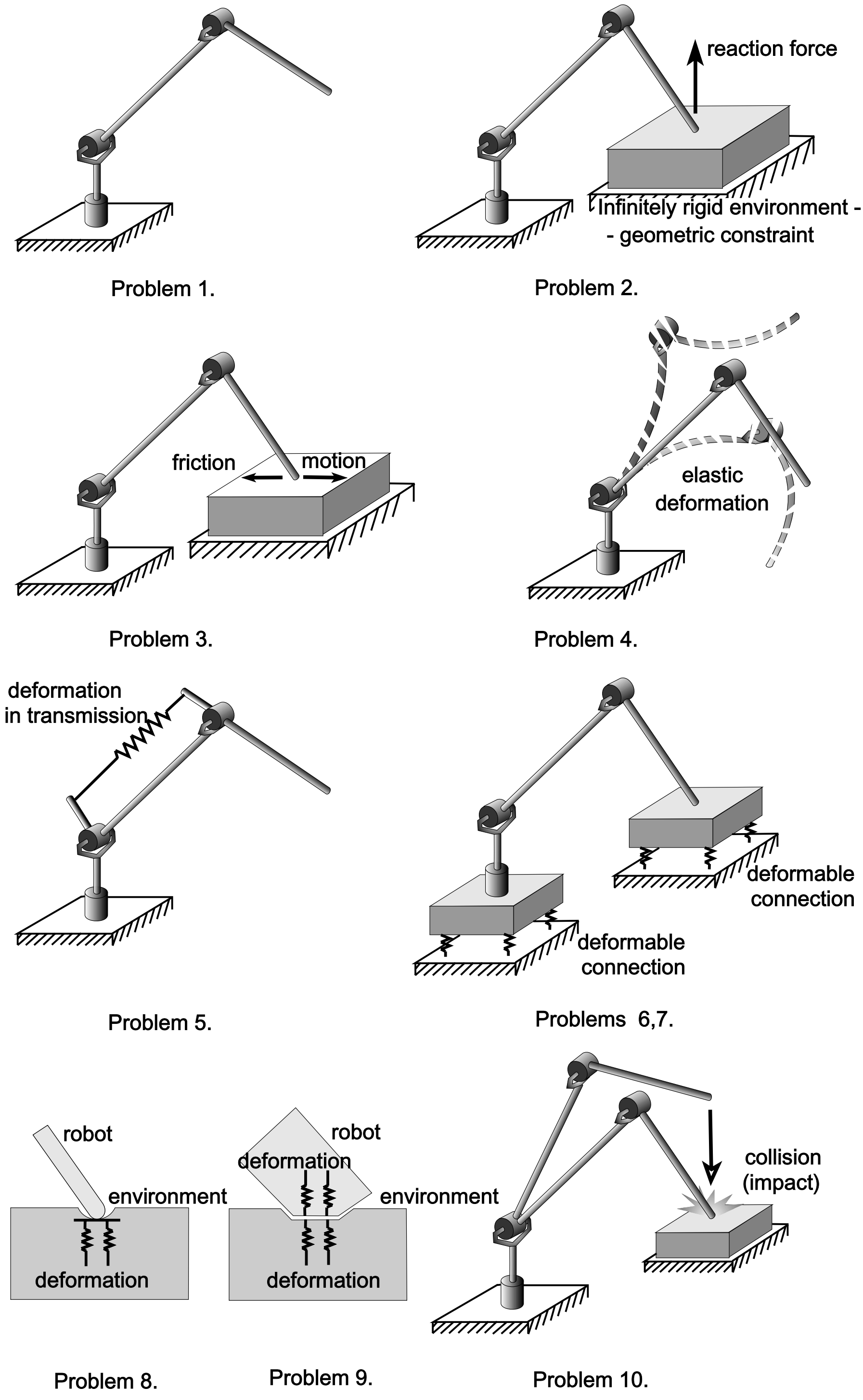


FIGURE 20.11 Different effects in robot dynamics.

The fact that many robot tasks include contact of the end-effector and the robot environment led to the first research on contact dynamics.[14] Robot environment was considered in the form of a geometric constraint (Problem 2 in Figure 20.11). The stationary and nonstationary cases were discussed. The constraints imposed to the end-effector resulted in reaction forces. Dynamic models were derived for arbitrary constraints (restricting one to six degrees of freedom). All robot elements were considered rigid. The derived models enabled solving motion along with computing contact forces. Friction between the robot and the environment was included (Problem 3). Collision problems and the solution to nonelastic impact were discussed. Practical examples considered were writing and assembly task.

After solving the dynamics of rigid robots, researchers attention turned to the elastic effects. The problem of flexible links was considered first (Problem 4). Some of the initial research looked for a simplified but fast solution of a flexible chain of the general structure,[13,14] while others intended to find a more exact model for some practical examples, i.e., single-link or two-link flexible arms.[17,18] Almost all existing dynamic models for flexible multi-link arms are found in some specific discretization methods, such as lumped mass,[13,14] assumed modes,[19] or finite elements.[20] A few researchers considered the computational efficiency of the proposed procedures.[21,22] Real-time calculation has become significant in advanced control algorithms for flexible robots.[23] Many researchers considered the linear deformation and neglected the effects of coupling between the components of deformation. Later research[24,25] takes care of these problems, thus producing more general models.

The next source of elastic effects is the transmission of torque. With electrical drives complex transmission between motor and joint shaft is usually needed. It is necessary to reduce speed and multiply torque. Thus, a kind of gear-box is present. Depending on their construction, gear-boxes introduce smaller or larger elastic deformations. The deformation is specially expressed with a harmonic-drive reducer, because elasticity is the essential property for its operation. If robot construction is such that gravitational load is reduced by placing all motors close to the robot base, then a system is needed to transmit torque and motion from the motor to the corresponding joint. This may be a chain, belt, shaft, etc. Any of these systems introduces its elastic deformation (Problem 5 in Figure 20.11). If the transmission is considered deformable, the joint shaft motion becomes independent of motor motion and only relatively high stiffness makes these motions close to each other. The number of DOFs is at least doubled. The initial results in this field were presented by Spong[26] and Potkonjak,[27] the foundations for further research. A mathematical model was derived to describe the dynamics of robots with elastic transmissions. The torque transmission included the harmonic-drive reducer, gears, and chains. Research[27] followed from practical work in robot design. Special attention was paid to some practical problems in forming the control loop: Should one measure the joint position or the motor angle? Generally speaking, the presence of unpowered DOFs represented the main problem with control of such robots. Further works elaborated this subject in more detail. One way to solve the tracking problem was presented in Kircanski, Timcenko, and Vukobratović[28] and included the measurement of torque for feedback formation. The next step in this research was to introduce constraints upon the motion of the end-effector, and thus, consider elastic joint robots in contact tasks. Several approaches to simultaneous force and position control in constrained robot systems with joint flexibility have been proposed in the literature.[29-32]

The elastic effects can be expressed with the robot support (Problem 6 in Figure 20.11). If connections of robot arm and its support were considered deformable, or if the robot was placed on a platform with pneumatic wheels, then oscillations would appear. However, these effects can be included in the existing models of rigid system by adding passive DOFs with stiffness and damping. A similar problem may appear on the environmental side. If the object to be grasped or processed in some other way is connected to its support by means of deformable connections, then oscillations arise (Problem 7). A dynamic environmental model is then needed.

With contact tasks, the most interesting deformation effects are expressed in the vicinity of contact points (Problems 8 and 9). Two bodies in contact produce a force upon each other and the force depends strongly on the elastic properties. For exact contact modeling, the elastodynamics in the contact zone has to be taken into account. In most research contact deformation was considered on the environment side only (Problem 8). The terminal link of the robot was assumed nondeformable. Such an approach can be justified in many industrial applications. It is due to the fact that tools are generally harder than the objects upon which they are acting. The general approach, however, would require the analysis of deformation on both sides of contact: environment and robot (Problem 9). Such discussion is not needed for pure theoretical reasons. With some applications, such as peg-in-hole assembly, it is a real situation. The same material is used for the peg and for the object with the hole. Hence, it is likely that both bodies in contact will be deformed. Most efforts in the field of contact deformation were made to create control strategies for contact tasks.[33-36] The main problem relates to the need to control position and force simultaneously. Different approaches to solving the problem could be distinguished. In the first approach, robot dynamics was considered in its rigid-body form and deformation in the contact zone was treated through stiffness and damping.[34] Because massless spring and damper were in question, no dynamics of the environment exists. From the standpoint of environment modeling, a more exact method has been proposed in Hogan.[35] The suggested control strategy was called impedance control. Environment was modeled by appropriate impedance. Thus, dynamics took place, but was restricted to the linear model. Complete dynamics of environment, including nonlinear effects, is the topic of position/force control of robot interacting with dynamic environment.[36,37]

Special discussion should be given about the collision (Problem 10). It is an always present effect because no contact can be precisely made to avoid impact. The first study of impact with robotic systems was given in Chumenko and Yuschenko.[38] The nonelastic impact between a robot and an object being grasped was solved. In Vukobratović and Potkonjak,[14] the collision of robot end-effector and a geometric constraint was elaborated. The impact was still considered plastic. The effect of friction was included. Both of these studies followed the classical approach based on the law of momentum. Another early result is Zheng and Hemami.[39] The influence of friction on body collision was discussed in detail in Keller[40] and Stronge.[41]

Hurmuzlu and Marghitu[42] considered a rigid-body collision of planar kinematic chains with multiple contact points. A successful algorithm for the numerical integration of a system subject to impacts was presented in Drenovac and Potknojak.[43] Brogliato and Orhant[44] formed the mathematical model of impulsive collision dynamics through the use of Schwartz's distributions, then studied the relationships between impulsive and continuous dynamic models, and analyzed the difficulties associated with transition phase control. Acaccia et al.[45] modeled the impact as a "black box," without a need to explicitly observe the compression and restitution phases. To achieve better insight, the collision could be modeled through elastodynamics. One way to do this was by means of the lumped mass approach.[46]

The final problem mentioned in this survey is redundancy. In early research in this field, redundancy was considered a problem of kinematics (avoiding obstacles, avoiding singular positions, etc.). Later research, however, saw redundancy as a possibility to improve robot dynamic performance.[47,48] The biomechanical approach to the solution of redundancy of a humanoid robot arm was proposed in Potknojak et al.[49] Special kind of redundancy appears in so-called systems with variable geometry.[50,51] The mechanism is designed to have an augmented number of DOFs (more than the kinematics of the task requires). However, this redundancy does not change end-effector maneuvering capabilities. It changes the inner structure of the robot. For this reason, it is called *internal redundancy.* The role of such redundancy is to avoid limitations imposed to actuators (torque and speed limits) and thus improve robot dynamic capabilities.

Among various problems in robot dynamics, in this chapter we emphasized the following: motion subject to geometric constraints, interaction with the dynamic environment, and effects of elastic transmissions.

20.5.2 Dynamics of Robot in Constrained Motion

Here we discuss contact tasks considering robot environment as a geometric constraint imposed to the motion of the end-effector. The discussion starts from the free motion of a rigid-body robot.

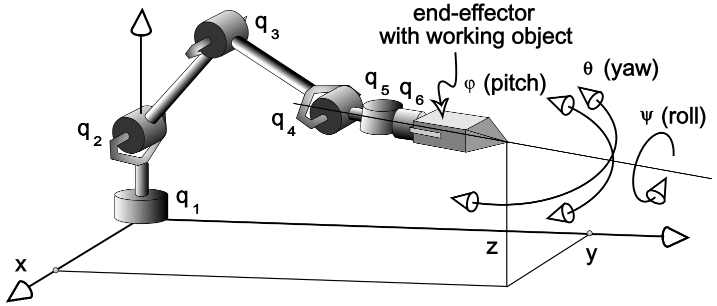


FIGURE 20.12 Internal position $q = (q_1, \ldots, q_6)$ and external position $X = (x, y, z, \theta, \phi, \psi)$.

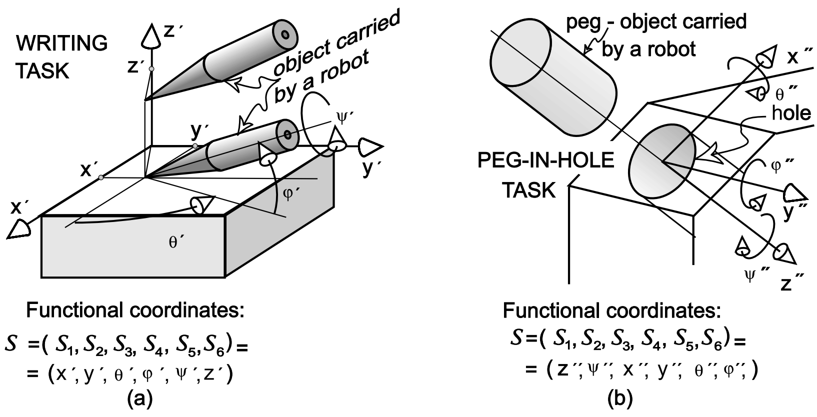


FIGURE 20.13 Two examples of functional coordinates.

The position of the chain, consisting of n links and n one-DOF joints (Figure 20.12), is defined by means of the joint coordinates: $q = [q_1 \;\ldots\; q_n]^T$. In practical operation, joint coordinates are not always suitable for use. Many tasks cannot be described in this way. For this reason, a new set of coordinates is defined, the *external coordinates*. By this term we usually understand position and orientation of the end-effector with respect to the immobile frame: Cartesian coordinates of the robot tip (x, y, z) and yaw, pitch, and roll angles (θ, ϕ, ψ), as shown in Figure 20.12. The complete external position vector is $X = [xyz\theta\phi\psi]^T$ and has a dimension of six. For simplicity, we assume that the robot has six joints and, thus, six joint coordinates ($n = 6$). In this way the elaboration of redundancy is avoided but the intention of our discussion is not compromised. In the majority of manipulation tasks, the external position is very suitable to apply. However, with contact tasks and some other process operations, it is more appropriate if the position of the end-effector is defined relative to the object being processed. Hence, we introduce a frame fixed to the object. Because the object may be mobile, the new frame would incorporate the law of motion. We assume that the object moves according to a given law that cannot be affected by robot action. This is necessary if one intends to describe the environment as a geometric constraint. With respect to the new frame, the position and the orientation of the end-effector are expressed by means of six coordinates: $s = [s_1 \;\ldots\; s_6]^T$. We call them *functional coordinates*. Two examples are shown in Figure 20.13. Case (a) represents the surface-type constraint appropriate for modeling the writing task. Case (b) represents the peg-in-hole assembly task. Note that the order of coordinates in vector s may be adopted arbitrarily, and we adopt the order suitable for the discussion that follows.

Each contact task consists of three phases: approaching, impact, and constrained motion. In the approaching phase the end-effector moves toward the constraint. The motion is usually planned to achieve zero-velocity contact and avoid impact. However, in a real situation, the always present perturbations cause the collision. The impact occurs, producing sudden change in velocities. After the impact phase, regular constrained motion starts. From the standpoint of modeling, the first phase represents free motion. Both internal coordinates (q) and functional coordinates (s) are free. All the discussion from the previous paragraphs holds and the dynamics is described by model (20.22). The kinematics should be described by means of s-coordinates. The relation between coordinates q and s can be expressed via a nonlinear function that can be nonstationary or stationary:

$$s = s(q,\ t) \text{ or } s = s(q) \tag{20.83}$$

With the nonstationary problems, the explicit appearance of time t is due to the mobility of the s-frame. For simplicity, we restrict our discussion to stationary problems (immobile frame s). The second derivative of the latter relation from (20.83) produces

$$\ddot{s} = J(q)\ddot{q} + A(q,\dot{q}) \tag{20.84}$$

where $J = \partial s/\partial q$ is the Jacobian matrix, and $A = (\partial^2 s / \partial q^2)\dot{q}^2$ is the adjoint vector.

Although all the coordinates from the set s are free and independent (in the approaching phase), it is useful to separate them into two subsets, s^f and s^c, of dimensions $6 - m$ and m, respectively. This separation follows from the nature of the constraint being approached, and will be explained later. With vector s separated into two subvectors, relation (20.84) becomes

$$\ddot{s}^f = J_f(q)\ddot{q} + A_f(q,\dot{q}) \tag{20.85}$$

$$\ddot{s}^c = J_c(q)\ddot{q} + A_c(q,\dot{q}) \tag{20.86}$$

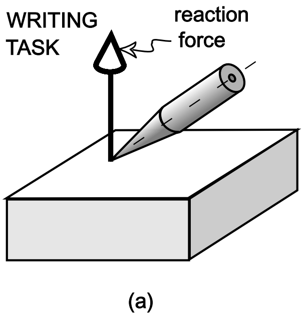
where the dimensions are J_f $((6 - m) \times 6)$, A_f $(6 - m)$, J_c $(m \times 6)$, A_c (m).

The approaching phase ends when the end-effector touches the constraint, that is, when the corresponding coordinate becomes zero. For the example from Figure 20.13a, the contact is established when coordinate s_6 (that is, z') reduces to zero. The impact occurs in the instant of contact. We find it more convenient to discuss impact dynamics later and elaborate the constrained motion (the third phase) now. In this case, we assume that the contact has already been established and the effects characteristic for the transient process have been finished. The imposed constraint restricts some motion of the end effector. Let this restriction be expressed in terms of s-coordinates by means of the condition

$$s^c = 0. \tag{20.87}$$

For example, (a) from Figure 20.13, the separation of vector s should be: $s^f = [s_1 \ldots s_5]^T$, $s^c = [s_6]$, $m = 1$; while for example (b) separation $s^f = [s_1\ s_2]^T$, $s^c = [s_3\ s_4\ s_5\ s_6]^T$, $m = 4$, is applied. Thus, s^c represents the set of constrained coordinates (upper index "c" stands for "constrained"). In example (a) there is only one such coordinate and accordingly one DOF is lost. In example (b), four DOFs are lost. In a general case, the constrained robot has 6-m remaining DOFs and its motion is described by means of $6 - m$ coordinates forming the vector s^f (upper index "f" stands for "free").

The restriction of motion results in reaction forces. Reaction will appear in the directions of the constrained coordinates s^c. Thus, there will be m independent components of reaction. Let these components form the vector F. Reaction may be in the form of a force, if translation is constrained, or in the form of a torque, if rotation is constrained. Figure 20.14 shows the reactions for the examples defined in Figure 20.13.



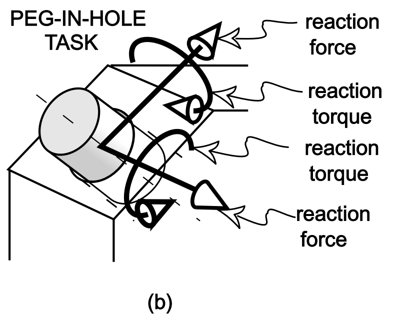


FIGURE 20.14 Two examples of reactions.

The dynamic model (20.22) used for free motion should now be reformulated to incorporate the reaction effects. Because F is the external force acting along directions s^c, the model can be written in the form:

$$H(q)\ddot{q} + h(q,\dot{q}) = \tau + J_c^T(q)F \tag{20.88}$$

where $J_c = \partial s^c/\partial q$ was introduced in (20.86). To solve the dynamics of the constrained motion, model (20.88) has to be considered along with the condition (20.87). It is more suitable if the constraint is expressed in differential form, and hence, (20.86) should be used, yielding

$$\ddot{s}^c = J_c(q)\ddot{q} + A_c(q,\dot{q}) = 0. \tag{20.89}$$

Equations (20.88) and (20.89) complete the dynamic model. They consist of 6 + m scalar equations and can be solved for 6 + m unknowns, accelerations $\ddot{q}$ and reactions F.

Now, we return to the second phase of the contact task, the problem of impact. At the beginning, we consider the simple surface-type constraint shown in Figures 20.13a and 20.14a. Impact occurs when s^c, that is, coordinate s_6, becomes zero and the impact action is directed along this coordinate. The impact may be more or less elastic and depending on that skipping could appear. To avoid such a complex discussion, we restrict consideration to the nonelastic problem. In this case, the end-effector will not leave the surface after the first contact, but move along it.

A slightly more complex problem would be constraint in the form of two surfaces. Although the intention is to move the end-effector along the line of intersection, in a real situation collision with one surface would happen first. Thus, two impacts would occur, one after the other. However, one might neglect this effect and consider the two contacts as simultaneous. In this case a complex impact has two components acting along the two constrained coordinates (dimension $s^c = 2$). Further generalization leads to the m-component constraint, that is, m restricted coordinates in vector s^c. The complex impact force F would have m components acting simultaneously. The example for $m = 4$ is shown in Figures 20.13b and 20.14b. Note that in real motion the complex impact represents a series of collisions with surfaces, but in the discussion that follows we neglect this effect. To solve the dynamics of the impact, we integrate the model of constrained motion over the impact interval (interval of transition). Let t' denote the beginning of transition, that is, the instant when s^c reduces to zero and the contact occurs. Let t'' be the instant when transition effects may be considered finished. The impact interval is then $\Delta t = [t', t'']$. For a geometric constraint and a nonelastic impact, it is justified to consider this interval infinitely short, that is, $\Delta t \to 0$ and $t'' \to t'$. We integrate model (20.88) over this interval to obtain:

$$H(q')\Delta\dot{q} = J_c^T(q')F\Delta t, \ \Delta\dot{q} = \dot{q}'' - \dot{q}' \tag{20.90}$$

where $\dot{q}' = \dot{q}(t')$ and $\dot{q}'' = \dot{q}(t'')$. Equation (20.89) gives:

$$J_c(q')\Delta\dot{q} = -J_c(q')\dot{q}' \tag{20.91}$$

Relations (20.90) and (20.91) describe the dynamics of impact. They represent the set of 6 + m scalar equations and can be solved for $\Delta\dot{q}$ and $F\Delta t$. $\Delta\dot{q}$ is the change in joint velocity (6 components) and $F\Delta t$ is the impact impulse (m components). Note that $F\Delta t \neq 0$ although $\Delta t \to 0$. This means that $F \to \infty$.

Let us briefly explain how we use the dynamic models to solve the three phases of the contact task numerically. In the approaching phase, model (20.22) is integrated checking the value of s^c. When s^c becomes zero, we turn to the impact model (20.90), (20.91) and solve the change in velocity. With this new initial state we start solving the dynamics of the constrained motion by integrating model (20.88) and (20.89).

20.5.3 Robot in Contact with Dynamic Environment

In the preceding paragraphs we established the contact problem as an important part of most robot tasks in industry. Geometric constraints were one way to form the dynamic model for such systems. The geometric constraints understood rigid-body contact and could be called "rigid constraints." However, with some theoretical and practical problems it was shown that this concept was not justified. The dynamic behavior of the environment appeared to be important. This led first to the concept of *soft constraint* and later to the idea of *dynamic environment*.

We do not intend to solve the general case of a robot interacting with a dynamic environment, but only to illustrate the idea. We model the robot's dynamics as if it were a rigid-body system and the dynamics of the environment is reduced to deformation and elastodynamic effects in the contact zone.

Let us consider the surface-type constraint (e.g., a writing task) shown in Figures 20.13a and 20.14a. Writing along the base has to be performed by applying some pressure force upon it. With the rigid base, coordinate z', that is, s_6, was constrained. If the base is not considered as infinitely rigid, but deformable, the motion in the perpendicular direction (negative s_6) will be possible. Thus, from the standpoint of kinematics there is no restriction on motion and no degrees of freedom are lost. Some kind of restriction follows from the base dynamics. Its elastic properties will keep the perpendicular motion small. Now the question arises of how to model environment dynamics (base, in this example). To avoid too complex a discussion on elastodynamics, we adopt the lumped-mass approach. Figure 20.15 shows the steps in the introduction of environment dynamics. Here, we elaborate model (c). Model (d) will be mentioned only briefly.

The contact task, as explained before, consists of three phases. In the first, approaching the surface, the problem of dynamic environment does not differ from the problem of geometric constraint. Robot dynamics is integrated using q-coordinates, but at the same time, we use functional s-coordinates to check whether the surface is reached. When s_6 (that is, s^c in a general case) reduces to zero, contact is present. At that instant impact occurs, and we assume that it is completely nonelastic. Impact is the second phase of the task. However, it is more convenient to explain the dynamics of contact motion (the third phase) before discussing the impact.

Figure 20.16 shows contact of the robot end-effector and the elastic base. Robot dynamics is described by model (20.88) and Equation (20.86) is used to express the relation between coordinates q and s^c. For the constraints of the surface-type, vector s^c reduces to one component (coordinate s_6), and hence, we rewrite (20.88) and (20.86) as

$$H(q)\ddot{q} - h(q,\dot{q}) = \tau + J_6^T(q)F \tag{20.92}$$

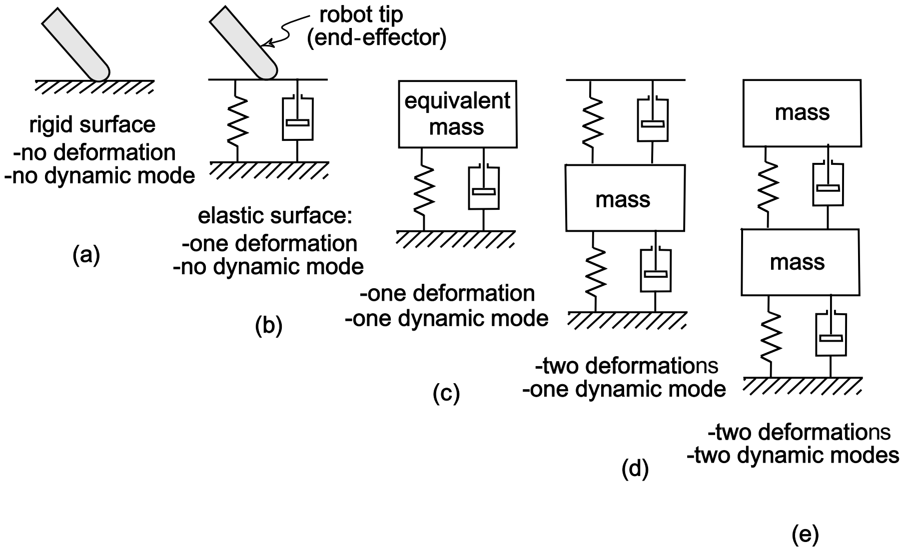


FIGURE 20.15 Modeling environment dynamics.

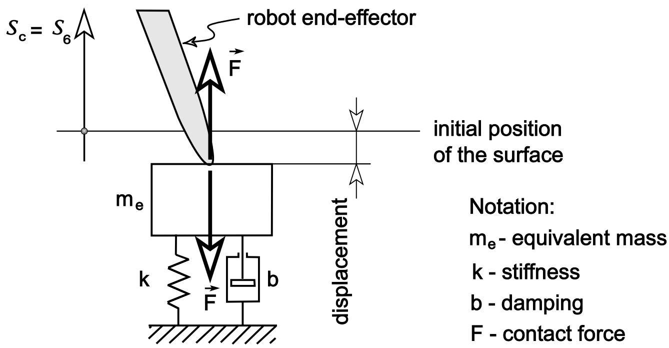


FIGURE 20.16 Elastodynamics of the base.

and

$$\ddot{s}_6 = J_6(q)\ddot{q} + A_6(q,\dot{q}) \tag{20.93}$$

where dimensions of Jacobian matrix J_6 are 1×6, and variables s_6, F, and A_6 are scalars. The dynamics of the base can be described by means of Newton's law applied to the equivalent mass m_e:

$$m_e\ddot{s}_6 = -F - ks_6 - b\dot{s}_6 \tag{20.94}$$

(Note that s_6 is negative all the time.) The set of Equations (20.92), (20.93), and (20.94) defines the dynamics of contact motion. The set consists of $6 + 1 + 1 = 8$ scalar equations and could be solved for 8 scalar unknowns: 6- component $\ddot{q}$, and scalars $\ddot{s}_6$ and F.

We now return to impact. It has been assumed nonelastic and the integration of Equations (20.92)–(20.94) over the impact interval $\Delta t \to 0$ yields

$$H(q')\Delta\dot{q} = J_6^T(q')F\Delta t \tag{20.95}$$

$$\Delta\dot{s}_6 = J_6(q')\Delta\dot{q} \tag{20.96}$$

$$m_e\Delta\dot{s}_6 = -F\Delta t \tag{20.97}$$

where q' refers to the instant of contact (t') and $\Delta\dot{q}$ and $\Delta\dot{s}_6$ represent the changes in velocities. Equations (20.95), (20.96), and (20.97) define the impact dynamics and could be solved for $\Delta\dot{q}$, $\Delta\dot{s}_6$, and the impact impulse $F\Delta t$ (since $\Delta t \to 0$, it has to be $F \to \infty$ in order to keep $F\Delta t \neq 0$).

If the environment is modeled as shown in Figure 20.15d, the entire deformation consists of the external and internal components. Thus, one additional DOF appears. The next specific of this model is that it does not require special impact treatment. The external deformation in the form of a massless spring eliminates the impact, and thus, after approaching, the system immediately enters the contact motion phase. One may say that the impact represents an initial period of contact motion while transient effects are exhibited.

20.5.4 Effects of Elastic Transmissions

Here, we consider an open-chain mechanism (unconstrained motion) and concentrate on the problem of transmitting torque from motors to joints. With robots driven by electrical motors, it is very common that motors are displaced from joints. They are moved toward the robot base to achieve better statics (that is, to unload the arm). In such cases, a different transmission could be applied between the motor and joint shafts. Some kind of a gear-box is usually present followed by chains, belts, or some other elements. Such transmission is a source of deviation, because each element may introduce its elastic deformation. Here, we assume that each joint has only one deformable element in the transmission. To make the discussion clearer, let this elastic element be the harmonic-drive reducer (HD). This will not compromise the generality because any other transmission element may be modeled in the same manner.

Figure 20.17a shows joint S_j with its motor and transmission. Motor dynamics can be described by mechanical Equation (20.62) and electrical Equation (20.63). If we neglect friction and inductivity, then these two equations could be combined to give the form (20.70). This model is expressed in terms of motor shaft angle θ_j. M_j that appears in the model represents motor output torque. The model holds for each motor, $j = 1, \ldots, n$. Dynamics of robot links are described by model (20.22) and expressed in terms of joint coordinates q_j, $j = 1, \ldots, n$. Vector $\tau = [\tau_1 \ \ldots \ \tau_n]^T$ that appears in the model contains the input torques for robot links (joint shaft). In Section 20.3, when nondeformable transmission was assumed, the relation between the motor variables (θ_j, M_j) and joint variables (q_j, τ_j) was defined as linear. It was expressed by means of Equations (20.68) and (20.69). The motion of the motor and that of the corresponding joint were dependent on each other. If flexibility of transmission is introduced, the situation becomes rather different. Motor motion (coordinate θ_j) and joint motion (coordinate q_j) become kinematically independent and only high stiffness keeps q_j close to θ_j/N_j. This means that the overall number of DOFs is doubled. We now concentrate on the concrete example of transmission shown in Figure 20.17. To generalize the discussion, in addition to the elasticity we take care of transmission inertia. In the schematic presentation (Figure 20.17b), the dashed line indicates the transmission stage. Its input torque is M_j and the output is τ_j. If the moments of inertia of the gears are I_j' and I_j'' and if other inertial effects are neglected, dynamics can be described by means of equation

$$(I_j' + I_j''/N_j^2)\ddot{\theta}_j = M_j - \tau_j/N_j. \tag{20.98}$$

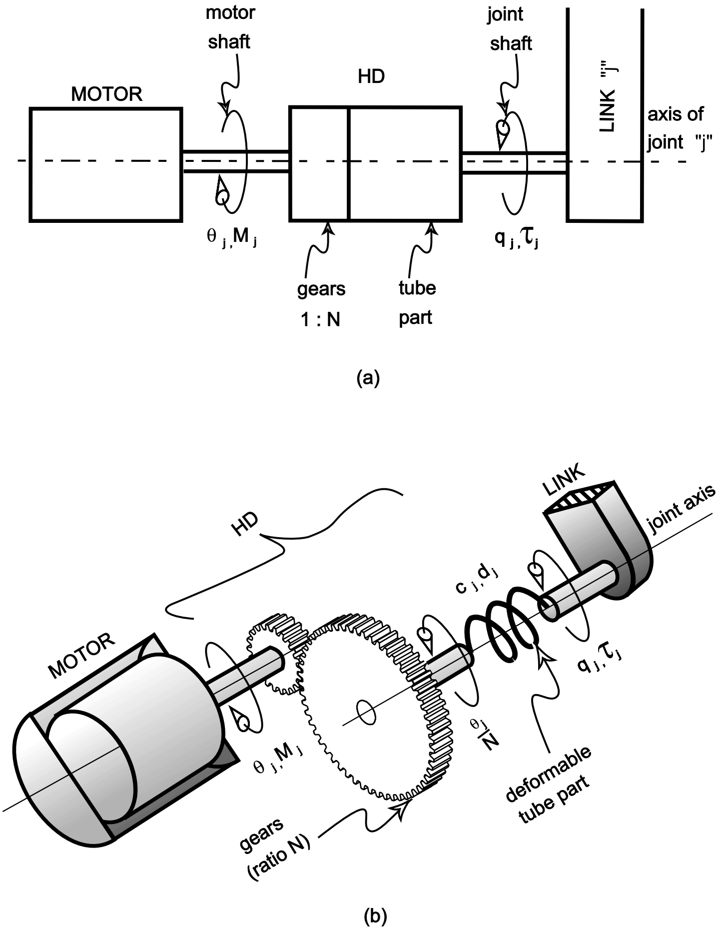


FIGURE 20.17 Deformable transmission.

It should be noted that any other transmission element (chain, belt, etc.) can be modeled in the same way, that is, as a combination of two rigid gears and a torsion spring element. In a generalized case, the transmission may consist of several stages, one after the other. Each stage would be treated as explained for the HD reducer in the present example. Inertia of gears may be expressed stronger or weaker, especially if compared with transmission friction that we neglected. Let us return to Figure 20.17. Joint torque τ_j follows from deformation of spring and hence

$$\tau_j = c_j(\theta_j / N_j - q_j) + d_j(\dot{\theta}_j / N_j - \dot{q}_j) \tag{20.99}$$

where c_j and d_j denote torsion stiffness and damping coefficients.

The complete dynamic model now includes Equations (20.22) for links, (20.70), $j = 1, \ldots, n$ for motors, and (20.98) and (20.99), $j = 1, \ldots, n$ for transmissions. The model can be integrated to give motions θ_j and q_j, $j = 1, \ldots, n$.

Appendix: Calculation of Transformation Matrices

We consider the mechanism as a kinematic chain consisting of n rigid links interconnected by one-DOF joint which can be either revolute or linear. The position of the chain is described by means

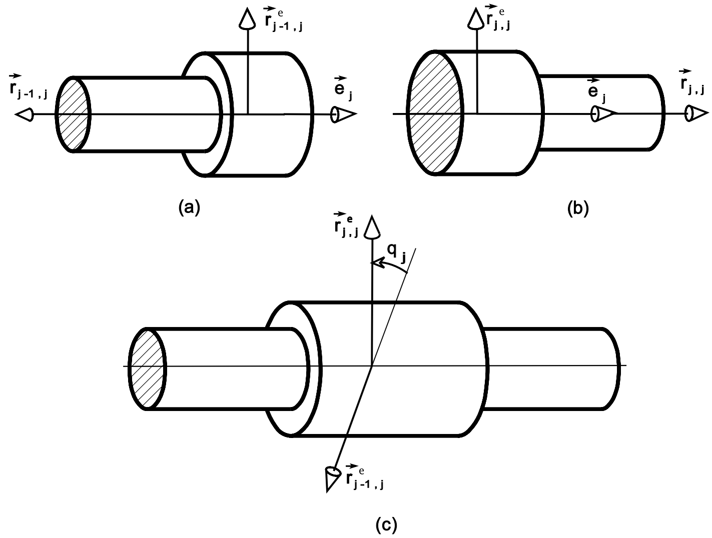


FIGURE 20.18 Definition of joint coordinate in the case of "specificity."

of coordinates $q_1, \ldots, q_n$. The definition of revolute coordinate requires some additional discussion. We start by referring to Figure 20.2 where the geometry of a link is defined and to Figure 20.3 that introduces the coordinate for a revolute joint. However, this definition of coordinate is not general. It requires that neither of vectors $\vec{r}_{j-1,j}$ or $\vec{r}_{j,j}$ is parallel with the axis $\vec{e}_j$. If it happens that one or both vectors are parallel with the joint axis, then modification of the definition is needed. Let us discuss this problem in detail.

If $\vec{r}_{j-1,j} \parallel \vec{e}_j$, then we say that there exists a specificity at the upper end of link j – 1 (the upper end of a link is the end oriented to the next link; e.g., joint S_j represents the upper end of link j – 1). In this case, it is necessary to introduce an additional vector, $\vec{r}^{\,*}_{j-1,j}$, which is not parallel with $\vec{e}_j$, and use it instead of $\vec{r}_{j-1,j}$ when defining the coordinate q_j (Figure 20.18a,c). The condition that $\vec{r}^{\,*}_{j-1,j}$ is not parallel with $\vec{e}_j$ is the only condition imposed in the choice of this additional vector. However, it is most convenient to define it as a unit vector perpendicular to $\vec{e}_j$.

If $\vec{r}_{j,j} \parallel \vec{e}_j$, then we say that there exists a "specificity" at the lower end of the j-th link (joint S_j represents the lower end of link j). In this case we introduce an additional vector, $\vec{r}^{\,*}_{j,j}$, which is not parallel with $\vec{e}_j$, and use it instead of $\vec{r}_{j,j}$ when defining the coordinate q_j (Figure 20.18b,c).

In this way, vectors $\vec{r}^{\,*}_{j-1,j}$ and $\vec{r}^{\,*}_{j,j}$ become the input data for the software that calculates the position or kinematics of the robot chain. Vector $\vec{r}^{\,*}_{j-1,j}$ is expressed in frame j – 1, and vector $\vec{r}^{\,*}_{j,j}$ in frame j.

We are now going to explain the calculation of the transformation matrix between the frame fixed to link j and the frame fixed to link j – 1. It is called the relative transformation matrix and is marked by $A_{j-1,j}$. The matrix is used to turn some vector from frame j to j – 1. If some vector $\tilde{\vec{c}}_j$ is considered, then $\underset{\sim}{\vec{c}}_j$ can be calculated starting from $\tilde{\vec{c}}_j$:

$$\underset{\sim}{\vec{c}}_j = A_{j-1,j}\tilde{\vec{c}}_j \tag{20.100}$$

To calculate the transformation matrix we need three vectors linearly independent of each other. We have to know the projections of these vectors to both frames, j – 1 and j. The calculation starts with solving the transformation matrix that corresponds to the extended joint. The first vector of the triple is $\vec{e}_j$. Its projections are known: $\underset{\sim}{\vec{e}}_j$ onto frame j – 1 and $\tilde{\vec{e}}_j$ onto frame j. We now introduce vector $\vec{a}_j$ as a unit vector of axis "a" shown in Figure 20.19. Axis "a" defines the extended position of the joint. The projections of the vector $\vec{a}_j$ onto frames j – 1 and j are

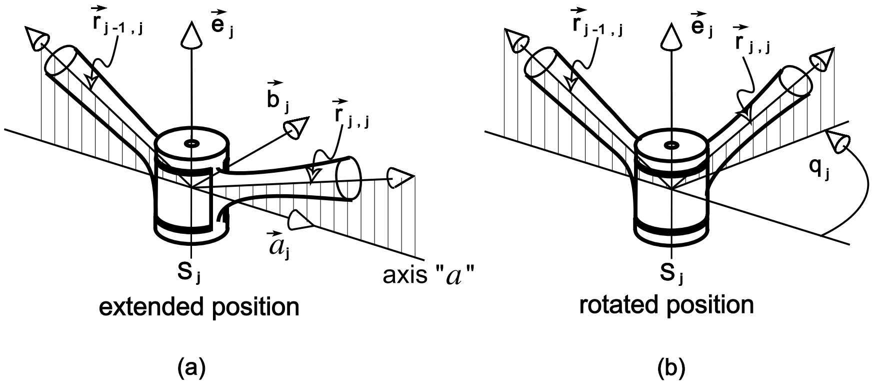


FIGURE 20.19 Revolute joint: extended and rotated position.

$$\underset{\sim}{\vec{a}}_j = \frac{-\underset{\sim}{\vec{e}}_j \times (\vec{\tilde{r}}_{j-1,j} \times \underset{\sim}{\vec{e}}_j)}{|\underset{\sim}{\vec{e}}_j \times (\vec{\tilde{r}}_{j-1,j} \times \underset{\sim}{\vec{e}}_j)|}, \quad \vec{\tilde{a}}_j = \frac{\vec{\tilde{e}}_j \times (\vec{\tilde{r}}_{j,j} \times \vec{\tilde{e}}_j)}{|\vec{\tilde{e}}_j \times (\vec{\tilde{r}}_{j,j} \times \vec{\tilde{e}}_j)|} \tag{20.101}$$

Vector $\vec{a}_j$ is the second vector of the triple. The third vector is obtained as

$$\vec{b}_j = \vec{e}_j \times \vec{a}_j \tag{20.102}$$

and has the projections

$$\underset{\sim}{\vec{b}}_j = \underset{\sim}{\vec{e}}_j \times \underset{\sim}{\vec{a}}_j, \quad \vec{\tilde{b}}_j = \vec{\tilde{e}}_j \times \vec{\tilde{a}}_j \tag{20.103}$$

Vectors $\vec{e}_j, \vec{a}_j$, and $\vec{b}_j$ are perpendicular to each other and thus linearly independent. The relation between projections onto two frames may be expressed (according to (20.100)) in the form:

$$\underset{\sim}{\vec{e}}_j = A^0_{j-1,j}\vec{\tilde{e}}_j, \quad \underset{\sim}{\vec{a}}_j = A^0_{j-1,j,}\,\vec{\tilde{a}}_j \quad \underset{\sim}{\vec{b}}_j = A^0_{j-1,j}\vec{\tilde{b}}_j \tag{20.104}$$

where the upper index “0” indicates the extended joint. Relations (20.104) can be united:

$$\left[\underset{\sim}{e}_j\ \underset{\sim}{a}_j\ \underset{\sim}{b}_j\right] = A^0_{j-1,j}\left[\tilde{e}_j \tilde{a}_j \tilde{b}_j\right] \tag{20.105}$$

where e_j is a column vector of dimension 3 × 1 that contains the projections of vector $\vec{e}_j$, and analogously holds for the other two vectors. From Equation (20.105), the transformation matrix of the extended joint is obtained:

$$A^0_{j-1,j} = \left[\underset{\sim}{e}_j\ \underset{\sim}{a}_j\ \underset{\sim}{b}_j\right]\left[\tilde{e}_j \tilde{a}_j \tilde{b}_j\right]^{-1} \tag{20.106}$$

Matrix $A^0_{j-1,j}$ $(\dim = 3 \times 3)$ has three columns that represent the unit vectors of frame j expressed in frame j – 1. Let us denote these columns by $\underset{\sim}{V}^0_{j1}$, $\underset{\sim}{V}^0_{j2}$ and $\underset{\sim}{V}^0_{j3}$. Thus,

$$A^0_{j-1,j} = \left[\underset{\sim}{V}^0_{j1}\ \underset{\sim}{V}^0_{j2}\ \underset{\sim}{V}^0_{j3}\right] \tag{20.107}$$

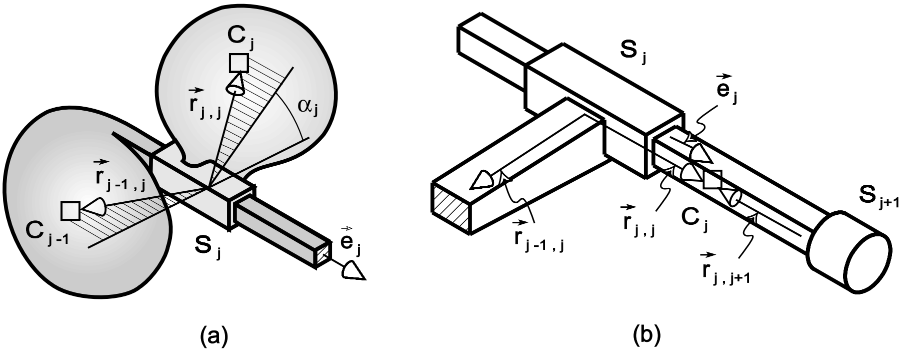


FIGURE 20.20 Linear joint.

Up to this point, the extended joint was considered, that is, the joint coordinate was assumed to be zero ($q_j = 0$). To find the transformation matrix $A_{j-1,\,j}$ that corresponds to some nonzero position ($q_j \neq 0$), it is necessary to rotate each of the unit vectors, $\vec{\underset{\sim}{V}}{}^0_{jk}, k = 1,\ 2,\ 3$, about the axis $\vec{\underset{\sim}{e}}_j$ by the angle q_j. For the rotation, we use the Rodrigues' formulae and mark the rotated vectors by $\vec{\underset{\sim}{V}}_{jk}$:

$$\vec{\underset{\sim}{V}}_{jk} = \vec{\underset{\sim}{V}}{}^0_{jk} \cos q_j + \left(1 - \cos q_j\right)\left(\vec{\underset{\sim}{e}}_j \times \vec{\underset{\sim}{V}}{}^0_{jk}\right) \vec{\underset{\sim}{e}}_j + \vec{\underset{\sim}{e}}_j \times \vec{\underset{\sim}{V}}{}^0_{jk} \sin q_j, \quad k = 1,\ 2,\ 3 \tag{20.108}$$

After rotation, unit vectors $\vec{\underset{\sim}{V}}_{jk}$ define the new transformation matrix. Thus, matrix $A_{j-1,j}$ that corresponds to some given position q_j is

$$A_{j-1,\,j} = \left[\vec{\underset{\sim}{V}}_{j1}\ \vec{\underset{\sim}{V}}_{j2}\ \vec{\underset{\sim}{V}}_{j3}\right] \tag{20.109}$$

We now turn to the problem when two links (j – 1 and j) are connected by means of a linear joint. Figure 20.4 explains the definition of a coordinate in such a joint. Since the relative motion between two links is a translation, it is clear that the transformation matrix is constant. The matrix $A_{j-1,j}$ can be solved in a manner analogous to that used for revolute joints. If that procedure is applied, from Figure 20.20a we conclude that for a linear joint the extended position is only fictive. The rotation is performed by the constant angle α_j. This angle has to be prescribed among the geometrical parameters. Further, Figure 20.20b shows that with linear joints specificity is a very common feature. Additional vectors (like $\vec{r}^{\,*}_{j,j}$) are needed in such cases.

The discussion in this Appendix concerned the transformation between two adjacent frames. However, we are often interested in transforming a vector from a link-fixed frame to the external stationary one. Relation (20.9) introduced notation A_j for such transformation matrix:

$$\vec{c}_j = A_j \vec{\vec{c}}_j \tag{20.110}$$

Moving the vector toward the robot support, from one frame to the other by using recursive expression (20.100), we finally reach the stationary frame. Thus, the transformation from the link-fixed frame to the external one is described by matrix

$$A_j = A_{0,1} A_{1,2} \ldots A_{j-1,\,j.} \tag{20.111}$$

References

1. Stepanenko, Yu., Method of analysis of spatial lever mechanisms, (in Russian), *Mechanics of Machines,* 23, 2, 48–57, 1970.
2. Vukobratović, M. and Stepanenko, Yu., Mathematical models of general anthropomorphic systems, *Math. Biosci.,* 17, 191–242, 1973.
3. Stepanenko, Yu. and Vukobratović, M., Dynamics of articulated open-chain active mechanisms, *Math. Biosci.,* 28, 1/2, 137–170, 1976.
4. Luh, J.Y.S., Walker, M.W., and Raul, R.P.C., On-line computational scheme for mechanical manipulators, *J. Dynam. Syst., Meas. Control,* 102, 1980.
5. Vukobratović, M. and Potkonjak, V., Contribution to computer construction of active chain models via lagrangian form, *J. Appl. Mech. — Trans. ASME,* 46, 1, 181–185, 1979.
6. Hollerbach, J.M., A recursive formulation of Lagrangian manipulator dynamics, *IEEE Trans. on SMC,* 10, 11, 730–736, 1980.
7. Popov, E.P., Vereschagin, A.F., and Zenkevich, S.A., *Manipulation Robots: Dynamics and Algorithms,* (in Russian), Nauka, Moscow, 1978.
8. Potkonjak, V. and Vukobratović, M., Two new methods for computer forming of dynamic equations of active mechanisms, *J. Mechanisms Machine Theory,* 14, 3, 189–200, 1979.
9. Corke, P., A Robotics toolbox for MATLAB. *IEEE Robot. Automat.*, 3, 1, 24–32, 1996.
10. Nethery, J.F., Robotica: A structured environment for computer-aided design and analysis of robots, Master's thesis, Univ. of Illinois, Urbana, 1993.
11. Vukobratović, M. and Stokić, D., *Control of Manipulation Robots: Theory and Application,* Springer-Verlag, Berlin, 1982.
12. Vukobratović, M., Potkonjak, V., and Hristic, D., Dynamic method for the evaluation and choice of industrial manipulators, *Proc. 9th ISIR,* Washington, 1979.
13. Vukobratović, M. and Potkonjak, V., *Dynamics of Manipulation Robots: Theory and Application,* Springer-Verlag, Berlin, 1982.
14. Vukobratović, M. and Potkonjak, V., *Applied Dynamics and CAD of Manipulation Robots,* Springer-Verlag, Berlin, 1985
15. Potkonjak, V., Thermal criterion for the selection of d.c. drives for industrial robots, *Proc. 16th ISIR,* Brussels, 218–224, 1986.
16. Inoue, K., Shiina, K., Takano, M., and Sasaki, K., Study on total computer-aided design system for robot manipulators, *Proc. 24th ISIR,* 729–736, 1993.
17. Cannon, R.H. and Schmitz, E., Initial experiments on the end-point control of a flexible one-link robot, *J. Robotics Res.,* 3, 62–75, 1984.
18. Fukuda, T. and Arakawa, A., Modeling and control characteristics for a two-degrees-of-freedom coupling system of flexible robotic arm, *JSME,* 30, 1458–1464, 1987.
19. Truckenbrodt, A., Dynamics and control methods for moving flexible structures and their application to industrial robots, *Proc. 5th World Cong. Theory Machines Mechanisms,* ASME, 1981.
20. Sunada, H.W., Dynamic analysis of flexible spatial mechanisms and robotic manipulators, Ph.D. Thesis, Univ. of California, Los Angeles, 1981.
21. Book, W.J., Recursive Lagrangian dynamics of flexible manipulator arms, *J. Robotics Res.,* 3, 87–101, 1984.
22. King, J.O., Gourishankar, V.G., and Rink, R.E., Lagrangian dynamics of flexible manipulators using angular velocities instead of transformation matrices, *IEEE Trans. SMC,* 17, 1059–1068, 1987.
23. Uchiyama, M. and Conno, A., Computed acceleration control for the vibration suppression of flexible robotic manipulators, *Proc. 5th ICAR,* Pisa, Italy, 126–131, 1991.
24. Šurdilović, D. and Vukobratović, M., One method for efficient dynamic modeling of flexible manipulators, *Mechanisms Machine Theory,* 31, 3, 397–315, 1996.
25. Šurdilović, D. and Vukobratović, M., Deflection compensation for large flexible manipulators, *Mechanisms Machine Theory,* 31, 3, 317–329, 1996.
26. Spong, M.W., Modeling and control of elastic joint robots, *J. Dynam. Syst., Meas. Control,* 109, 309–319, 1987.
27. Potkonjak, V., Contribution to the dynamics and control of robots having elastic transmission, *Robotica,* 6, 63–69, 1988.

28. Kircanski, N., Timcenko, A., and Vukobratović, M., Position control of robot manipulators with elastic joints using force feedback, *J. Rob. Syst.,* 7, 4, 535–554, 1990.
29. Ahmad, S., Constrained motion (force/position) control of flexible joint robots, *Proc. IEEE Conf. Decision Control,* Brighton, England, 1397–1402, 1991.
30. Mills, J.K., Stability and control of elastic-joint robotic manipulators during constrained motion tasks, *IEEE Trans. Robotics and Automation,* 8, 1, 119–125, 1992.
31. Brogliato, B. and Lozano-Leal, R., Adaptive force/position control of constrained manipulators with flexible joints, *Proc. Am. Control Conf.,* Boston, 968–969, 1992.
32. Krishnan, H., Robot force and position control including effects of joint flexibility and actuator dynamics, *Proc. 26th ISIR,* Singapore, 185–190, 1995.
33. Raibert, M.H. and Craig, J.J., Hybrid position/force control of manipulators, *J. Dynam. Syst., Meas. Control,* 103, 126–133, 1981.
34. Khatib, O., A unified approach for motion and force control of robot manipulators: the operational space formulation, *IEEE J. Robotics Automation,* 5, 3, 107–115, 1987.
35. Hogan, N., Impedance control: an approach to manipulation, Part I–III, *J. Dynam. Syst., Meas. Control,* 107, 1–24, 1985.
36. Vukobratović, M. and Ekalo, Y., New approach to control of robotic manipulators interacting with dynamic environment, *Robotica,* 14, 1, 31–39, 1996.
37. Vukobratović, M., How to control robots interacting with dynamic environment, *J. Intelligent Robotic Systems,* 19, 119–152, 1997.
38. Chumenko, V.N. and Yuschenko, A.S., Impact effects upon manipulation robot mechanism, (in Russian), *Technical Cybernetics,* 22, 4, 103–108, 1981.
39. Zheng, Y.F. and Hemami, H., Mathematical modeling of a robot collision with its environment, *J. Robotic Systems,* 2(3), 289–307, 1985.
40. Keller, J.B., Impact with friction, *J. Appl. Mech.,* 53(1–4), 1986.
41. Stronge, W.J., Rigid body collision with friction, *Proc. R. Soc. Lond. A.,* 431, 169–181, 1990.
42. Hurmuzlu, Y. and Marghitu, D.B., rigid body collision of planar kinematic chain with multiple contact points, *Int. J. Robotic Res.,* 13, 1, 82–92, 1994.
43. Drenovac, V. and Potkonjak, V., Contribution to the modeling of impact with robotic systems, *Robotica,* 11, Pt. 4, 445–452, 1993.
44. Brogliato, B. and Orhant, P., On the transition phase in robotics: impact models, dynamics and control, *Proc. IEEE Conf. Robotics Automation,* San Diego, 1994.
45. Acaccia, G.M., Gagetti, P.C., Callegari, M., Michelini, R.C., and Molfino, R.M., Modeling the impact dynamics of robotic manipulators, *Proc. 4th IFAC Symp. Robot Control,* Capri, Italy, 559–564.
46. Tornambe, A., Modeling and controlling two-degrees-of-freedom impacts, *Proc. 3rd IEEE Mediterranean Conf. Control Automation,* Lymassol, Cyprus, 1995.
47. Hogan, N., Sharon, A., and Hardt, E.D., High bandwidth force regulation and inertia reduction using a macro/micro manipulator system, *Proc. IEEE Conf. Robotics Automation,* 126–132, 1988.
48. Potkonjak, V. and Krstulovic, A., Contribution to the kinematics and dynamics of redundant robots via distributed positioning, *J. Intelligent Robotic Systems,* 5, 229–239, 1992.
49. Potkonjak, V., Popovic, M., Lazarevic, M., and Sinanovic, J., Redundancy problem in writing: from human to anthropomorphic robot arm, *IEEE Trans. SMC,* 28, 6, 790–806, 1998.
50. Vukobratović, M., Potkonjak, V., Systems with variable geometry: Concept and prospects, *ASME J. Dyn. Syst., Meas. Control,* 2, 121, 308–312, 1999.
51. Vukobratović, M., Potkonjak, V., and Matijevic, V., Internal redundancy — the way to improve robot dynamics and control performances, *J. Intelligent Robotic Systems,* 27, 31–66, 2000.