16

Force/Impedance Control for Robotic Manipulators

16.1 Introduction

The convergence of the fields of robotics and automation in many industrial applications is well established. During the execution of many industrial automation tasks, the robot manipulator is often in contact with its environment (either directly or indirectly via an end-effector payload). For purely positioning tasks such as spray painting that have negligible force interaction with the environment, controlling the manipulator end-effector position results in satisfactory performance. However, in applications such as polishing or grinding, the manipulator end effector will experience interaction forces from the environment. Furthermore, due to contact with the environment, the motion of the end effector in certain directions is restricted. In the field of robotics, this resulting motion is often referred to as constrained motion or compliant motion. This chapter focuses on the control strategies of manipulators for constrained motion.

While there are many techniques that may be used to design controllers for constrained motion, a popular active compliant approach is the use of force control. The fundamental philosophy is to regulate the contact force of the environment. This is often supplemented by a position control objective that regulates the orientation and the location of the end effector to a desired configuration in its environment. For example, in the case of grinding applications, the motion of the manipulator arm is constrained by the grinding surface. It can easily be seen that it is vital to control not only the position of the manipulator end effector to ensure contact with the grinding surface but also the interaction forces to ensure sufficient contact force to enable grinding action.
Another example that illustrates the need for combining force and position controllers is drawing on a blackboard with chalk. If a robot manipulator is required to draw multiple instances of one shape (e.g., a circle) on the blackboard, the end-effector position must follow a specific (in this case, circular) trajectory while ensuring that the chalk applies a certain amount of force on the blackboard. We know that if the force is too small, the drawing may not be visible; however, if the force is too large, the chalk will break. Moreover, as multiple instances of shapes are drawn, the length of the chalk reduces. Thus, one can appreciate complexities and intricacies of simple tasks such as drawing/writing using chalk. This example clearly illustrates the need for desired force trajectory in addition to a desired position trajectory. Another example that motivates the use of a position/force control strategy is the pick-and-place operations using grippers for glass tubes. Clearly, if the gripper closes too tight, the glass may shatter, and if it is too lose, it is likely the glass may slip. The above examples establish the motivation for position and force control.

In many applications, the position/force control objectives are considerably more intertwined than in the examples discussed thus far. Consider the example of a circular diamond-tipped saw cutting a large block of metal, wherein the saw moves from one end to the other. The motion from one end to the other manifests itself as the position control objective, whereas cutting the block of metal without the saw blade binding is the force control objective. The speed of cutting (position control objective) depends on many parameters, such as geometric dimensions and material composition. For example, for a relatively softer metal such as aluminum, the saw can move safely from one end to the other much faster than in the case of a relatively harder material such as steel. It is obvious that the cutting speed would be much faster for thinner blocks of metal.

To achieve control objectives such as those discussed above, the controller must be designed to regulate the dynamic behavior between the force exerted on the environment (force control objective) and the end effector motion (position control objective). This approach forms the basis of impedance control.

In this chapter, we present the following four control strategies that control the force exerted by the end effector on its environment in conjunction with the typical end effector position control for an \( n \)-link robot manipulator:

- Stiffness control
- Hybrid position/force control
- Hybrid impedance control
- Reduced order position/force control

Furthermore, for each section, an example for a two-degree-of-freedom (2-DOF) Cartesian manipulator is also presented.

**Remark 16.1** It is important to note that for all the aforementioned strategies, the desired velocity and force trajectories should be consistent with the environmental model [1]. If such is not the case, the trajectories may be modified by techniques such as “kinestatic filtering” (see [1] for more details).

### 16.2 Dynamic Model

This section presents the formulation of the manipulator dynamics that facilitates the design of strategies to control position and force. First, the widely known joint-space model is presented. Next, we define some notation for the contact forces (sometimes referred to as interactive forces) exerted by the manipulator on the environment by employing a coordinate transformation.

#### 16.2.1 Joint-Space Model

The dynamics of motion for an \( n \)-link robotic manipulator [2] are constructed in the Euler-Lagrange form as follows:

\[
M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + N(q, \dot{q}) + \tau_e = \tau 
\]  
(16.1)
where
\[ N(q, \dot{q}) = F(\dot{q}) + G(q) \]  
(16.2)

\( M(q) \in \mathbb{R}^{n \times n} \) denotes the inertia matrix, \( V_m(q, \dot{q}) \in \mathbb{R}^{n \times n} \) contains the centripetal and Coriolis terms, \( F(\dot{q}) \in \mathbb{R}^n \) contains the static (Coulomb) and dynamic (e.g., viscous) friction terms, \( G(q) \in \mathbb{R}^n \) is the gravity vector, \( \tau_e(t) \in \mathbb{R}^n \) represents the joint-space end-effector forces exerted on the environment by the end-effector, \( q(t) \in \mathbb{R}^n \) represents the joint-space variable vector, and \( \tau(t) \in \mathbb{R}^n \) is the torque input vector.

### 16.2.2 Task-Space Model and Environmental Forces

In order to facilitate the design of force controllers, the forces are commonly transformed into the task-space via a Jacobian matrix by using a coordinate transformation [2]. Note that this Jacobian matrix is defined in terms of the task-space coordinate system and dependent on the robot application. Specifically, depending on the type of application, the axes (directions) of force control and motion may vary and are captured by the definition of the Jacobian.

Toward formulating the task-space dynamics, we define a task-space vector \( x \in \mathbb{R}^n \) as follows:
\[ x = h(q) \]  
(16.3)

where \( h(q) \) is obtained from the manipulator kinematics and the joint and task-space relationships. The derivative of the task-space vector is defined as
\[ \dot{x} = J(q) \dot{q} \]  
(16.4)

where the task-space Jacobian matrix \( J(q) \in \mathbb{R}^{n \times n} \) is defined as [2]
\[ J(q) = \begin{bmatrix} I & 0 \\ 0 & T \end{bmatrix} \frac{\partial h(q)}{\partial q} \]  
(16.5)

with \( I \) being the identity matrix, \( 0 \) being the zero matrix, and the transformation matrix \( T \) is used to convert joint velocities to derivatives of roll, pitch, and yaw angles associated with end-effector orientation. It should be noted that the joint-space representation of the force exerted on the environment (defined first in Equation (16.1)) can be rewritten as
\[ M(q) \ddot{q} + V_m(q, \dot{q}) \dot{q} + N(q, \dot{q}) + J^T(q) f = \tau \]  
(16.6)

where \( f(t) \in \mathbb{R}^n \) denotes the vector of forces and torques exerted on the environment in task-space.

**Example 16.1 Task-Space Formulation for a Slanted Surface**

The following example presents the task-space formulation of the dynamics for a 2-DOF Cartesian manipulator (i.e., both joints are prismatic) moving along a slanted surface as shown in Figure 16.1. To that end, the objectives are to formulate the manipulator dynamics and to decompose the forces exerted on the slanted surface into tangential and normal components. The motion component of the dynamics can easily be formulated from the unconstrained robot model. Thus, after removing the surface, and hence the interaction forces \( f_1 \) and \( f_2 \), the manipulator dynamics are constructed as follows:
\[ M\ddot{q} + F(\dot{q}) + G = \tau \]  
(16.7)

where
\[
q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}, \quad M = \begin{bmatrix} m_1 & 0 \\ 0 & m_1 + m_2 \end{bmatrix}, \quad F = \begin{bmatrix} F_1(\dot{q}_1) \\ F_2(\dot{q}_2) \end{bmatrix}, \quad \tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ (m_1 + m_2)g \end{bmatrix}
\]  
(16.8)
FIGURE 16.1 Manipulator moving along a slanted surface [4].

Note that the explicit structure of the friction vector $F$ is beyond the scope of the material presented in this chapter.

The next step is to incorporate the contact forces. To that end, let $x \in \mathbb{R}^2$ denote the task-space vector defined as

$$x = \begin{bmatrix} u \\ v \end{bmatrix}$$  \hspace{1cm} (16.9)

where $u$ represents the normal distance to the surface and $v$ represents the tangential distance to the surface of a fixed coordinate system. As discussed in Equation (16.3), the representation of the joint-space coordinates into task-space is given by

$$x = h(q)$$  \hspace{1cm} (16.10)

where $h(q)$ is constructed based on the system geometry as follows:

$$h(q) = \frac{1}{\sqrt{2}} \begin{bmatrix} q_1 - q_2 \\ q_1 + q_2 \end{bmatrix}$$  \hspace{1cm} (16.11)

Based on Equation (16.5) and Equation (16.11), the task-space Jacobian matrix can be constructed as follows:

$$J = \frac{\partial h(q)}{\partial q} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$  \hspace{1cm} (16.12)

It should be noted that the matrix $T$ in Equation (16.5) is an identity matrix because the end-effector angles are irrelevant to this problem. Following the procedure outlined in the previous section, the robot manipulator equation can be formulated as follows:

$$M\ddot{q} + F(\dot{q}) + G + J^T f = \tau$$  \hspace{1cm} (16.13)

where

$$f = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$  \hspace{1cm} (16.14)

Remark 16.2 It should be noted that the normal force $f_1$ and the tangential force $f_2$ in Figure 16.1 are drawn with respect to the task-space coordinate system.
16.3 Stiffness Control

Most early robot manipulators used in industrial automation were required to perform positional tasks (e.g., spray painting). As a result, the robot manipulators were designed to be very rigid, and the control strategies of choice were simple position controllers that achieved satisfactory positioning accuracy. However, in applications such as sanding and grinding, a rigid (or "stiff") manipulator does not lend itself favorably to the force control objective. If, however, the "stiffness" of the manipulator can be controlled [3], the force control objective can be more easily accomplished. This section presents the stiffness control formulation for an $n$-DOF robot manipulator and is followed by an example for a 2-DOF Cartesian robot.

16.3.1 Controller Design

The stiffness control objective for an $n$–link is formulated in this section. First, the force exerted by the manipulator on the environment is defined as

$$ f \equiv K_e (x - x_e) \tag{16.15} $$

where $K_e \in \mathbb{R}^{n \times n}$ is a diagonal, positive semidefinite, constant matrix used to denote the environmental stiffness with $x_e \in \mathbb{R}^n$ being the task-space vector denoting static location of the environment.

A proportional-derivative type of controller can be designed as follows to achieve the multi-dimensional stiffness control objective:

$$ \tau = J^T(q)(-K_v \dot{x} + K_p \ddot{x}) + N(q, \dot{q}) \tag{16.16} $$

where the task-space tracking error is defined as

$$ \ddot{x} = x_d - x \tag{16.17} $$

with $x_d$ denoting the desired constant end-effector position. After formulating the closed-loop error system and performing a comprehensive stability analysis [4], it can be shown that for the $i$th diagonal element of matrices $K_e$ and $K_p$:

$$ \lim_{t \to \infty} f_i \approx K_{pi} (x_{di} - x_{ei}) \quad \text{in the constrained directions} \tag{16.18} $$

$$ \lim_{t \to \infty} x_i = x_{di} \quad \text{in the unconstrained directions} $$

The result in Equation (16.18) can be interpreted as follows. In the unconstrained task-space direction, the desired steady-state setpoint is reached and the position control objective is achieved. In the constrained task-space direction (with the assumption that $K_{ei} \gg K_{pi}$ for force control), the force control objective is achieved and $K_{pi}$ may be interpreted as the manipulator stiffness in the task-space direction.

Example 16.2 Stiffness Control of a Cartesian Manipulator

The following example presents the design of a stiffness control strategy for the 2-DOF Cartesian manipulator shown in Figure 16.1. The control objective is to move the end effector to a desired final position $v_d$ while exerting a desired normal force $f_{d1}$. In this example, the surface friction $f_2$ and the joint friction are assumed to be negligible, and the normal force $f_1$ is assumed to satisfy the following relationship:

$$ f_1 = k_e (u - u_e) \tag{16.19} $$

---

1A diagonal element of the matrix $K_e$ is zero if the manipulator is not constrained in the corresponding task-space direction.
To accomplish the control objective, the stiffness control algorithm is designed as follows:

$$\tau = J^T(q)(-K_v \dot{x} + K_p \ddot{x}) + G(q)$$  \hspace{1cm} (16.20)

where

$$\ddot{x} = \begin{bmatrix} u_d - u \\ v_d - v \end{bmatrix}$$  \hspace{1cm} (16.21)

$u_d$ denotes the desired normal position, the quantities $\tau$, $J(q)$, and $G(q)$ retain their definitions of Example 1, and the gain matrices can simply be selected as $K_v = k_v I$ and $K_p = k_p I$ with $k_v$ and $k_p$ being positive constants. It is important to select $k_p$ such that $k_p \ll k_e$ as required by the stiffness control formulation. To satisfy the control objective, the desired normal position $u_d$ needs to be specified. To that end, by using the first condition in Equation (16.18), we can solve for $u_d$ from the following equation:

$$f_{d1} = k_p(u_d - u_e)$$  \hspace{1cm} (16.22)

### 16.4 Hybrid Position/Force Control

An important disadvantage of the stiffness controller discussed in the previous section is that the desired manipulator position and the desired force exerted on the environment are constants. That is, the controller is restricted to a constant setpoint regulation objective. In applications such as polishing and deburring, the end effector must track a prescribed path while tracking some desired force trajectory on the environment. In such a scenario, the stiffness controller will not achieve satisfactory results.

To that end, another approach that simultaneously achieves position and force tracking objectives is used. The hybrid position/force controller developed in [5, 6] decouples the position and force control problems into subtasks via a task-space formulation. This formulation is critical in determining the directions in which force or position should be controlled. Once these subtasks are identified, separate position and force controllers can be developed. In this section, we first present a hybrid position/force controller for an $n$-link manipulator and then discuss this control strategy with respect to the 2-DOF Cartesian manipulator shown in Figure 16.1.

#### 16.4.1 Controller Design

The following hybrid position/force controller first uses feedback-linearization to globally linearize the robot manipulator dynamics and then employs linear controllers to track the desired position and force trajectories. To that end, by using the task-space transformation of Equation (16.3) to decompose the normal and tangential surface motions and after some mathematical manipulation, the robot dynamics can be expressed as

$$M(q)J^{-1}(q)(\ddot{x} - \dot{J}(q)\dot{q}) + V_m(q, \dot{q})\dot{q} + N(q, \dot{q}) + J^T(q) f = \tau$$  \hspace{1cm} (16.23)

Based on the structure of Equation (16.23) and the control objectives, a feedback-linearizing controller for the above system can be constructed as follows:

$$\tau = M(q)J^{-1}(q)(\ddot{\hat{y}} - \dot{J}(q)\dot{q}) + V_m(q, \dot{q})\dot{q} + N(q, \dot{q}) + J^T(q) f$$  \hspace{1cm} (16.24)

where $\ddot{\hat{y}} \in \Re^n$ denotes the linear position and force control strategies. Note that from Equation (16.23) and Equation (16.24), we have

$$\ddot{x} = \ddot{\hat{y}}$$  \hspace{1cm} (16.25)
Given that the system dynamics in Equation (16.25) have been decoupled in the task-space into tangential (position) and normal (force) components denoted by subscripts $T$ and $N$, respectively, we can design separate position and force control algorithms.

### 16.4.1.1 Tangential Position Control Component

The tangential task-space components of $\ddot{y}$ can be represented as follows:

$$\ddot{x}_T = \ddot{y}_T$$  \hspace{1cm} (16.26)

where $\ddot{y}_T$ denotes the $i$th linear tangential task-space position controller. The corresponding tangential task-space tracking error is defined as follows:

$$\ddot{x}_T = x_{Tdi} - x_T$$  \hspace{1cm} (16.27)

where $x_{Tdi}$ represents the desired position trajectory. Based on the control objective and the structure of Equation (16.27), we can design the following linear algorithm:

$$\ddot{y}_T = \ddot{x}_{Tdi} + k_{Tv}\dot{x}_T + k_{Tp}\ddot{x}_T$$  \hspace{1cm} (16.28)

where $k_{Tv}$ and $k_{Tp}$ are the $i$th positive control gains. After substituting this controller defined in Equation (16.28) into Equation (16.26), we obtain the following closed-loop dynamics:

$$\ddot{x}_T + k_{Tv}\dot{x}_T + k_{Tp}\ddot{x}_T = 0$$  \hspace{1cm} (16.29)

Given that $k_{Tv}$ and $k_{Tp}$ are positive, an asymptotic position tracking result is obtained:

$$\lim_{t \to \infty} \ddot{x}_T = 0$$  \hspace{1cm} (16.30)

### 16.4.1.2 Normal Force Control Component

The normal task-space components of $\ddot{y}$ can be represented as follows:

$$\ddot{x}_N = \ddot{y}_N$$  \hspace{1cm} (16.31)

where $\ddot{y}_N$ denotes the $j$th linear normal task-space force controller. The corresponding normal task-space tracking error is defined as follows:

$$f_N = k_e(x_N - x_e)$$  \hspace{1cm} (16.32)

where the environment is modeled as a spring, $k_e$ denotes the $j$th component of the environmental stiffness, and $x_e$ represents the static location of the environment in the normal direction. The force dynamics can be formulated by taking the second time derivative of Equation (16.32) as follows:

$$\ddot{x}_N = \frac{1}{k_e} f_N = \ddot{y}_N$$  \hspace{1cm} (16.33)

which relates the acceleration in the normal direction to the second time derivative of the force. To facilitate the construction of a linear controller, we define the following force tracking error:

$$\dot{f}_N = f_{Nd} - f_N$$  \hspace{1cm} (16.34)

where $f_{Nd}$ denotes the $j$th of the desired force trajectory in the normal direction. A linear controller that achieves the force tracking control objective is defined as

$$\ddot{y}_N = \frac{1}{k_e} (\dot{f}_{Nd} + k_{Nv}\dot{f}_N + k_{Np}\ddot{f}_N)$$  \hspace{1cm} (16.35)
where $k_{Nj}$ and $k_{Npj}$ are $j$th positive control gains. After substituting the controller into Equation (16.33), we obtain the following closed-loop dynamics

$$\ddot{f}_{Nj} + k_{Nj}\dot{f}_{Nj} + k_{Npj}f_{Nj} = 0$$

(16.36)

that achieves asymptotic force tracking as shown below:

$$\lim_{t \to \infty} \ddot{f}_{Nj} = 0$$

(16.37)

**Example 16.3 Hybrid Position/Force Control along a Slanted Surface**

This example discusses the hybrid position/force control strategy for the 2-DOF Cartesian manipulator shown in Figure 16.2. The control objective is to move the end effector with a desired surface position trajectory $v_d(t)$ while exerting a desired normal force trajectory denoted by $f_d(t)$. Note that in the case of the stiffness controller, the desired position and desired force were constant setpoints. With the assumption of negligible joint friction, we also assume that the normal force $f_1$ satisfies the following relationship:

$$f_1 = k_e(u - u_c)$$

(16.38)

To accomplish the control objective, the hybrid position/force controller is defined as follows:

$$\tau = M J^{-1}\dot{y} + G + J^T f$$

(16.39)

where $\dot{y} \in \mathbb{R}^2$ represents the linear position and force controllers and the quantities $\tau, J(q)$, and $G(q)$ have been defined in Example 1. This controller decouples the robot dynamics in the task-space as follows:

$$\ddot{x} = [\ddot{u} \ddot{v}] = [\dot{x}_{N1} \dot{x}_{T1}] = [\dot{y}_{N1} \dot{y}_{T1}] = \ddot{y}$$

(16.40)

**FIGURE 16.2** Hybrid position/force controller [4].
where \( u \) represents the normal component of the task-space and \( v \) represents the tangential component of the task-space. The corresponding linear position and force controller is designed as follows:

\[
\ddot{y}_T = \dot{x}_{T1} + k_{T1} \dot{x}_T + k_{Tp1} \ddot{x}_T
\]

and

\[
\ddot{y}_T = \frac{1}{k_{e1}} (f_{Nd1} + k_{Nv1} \dot{f}_{N1} + k_{Np1} \ddot{f}_{N1})
\]

### 16.5 Hybrid Impedance Control

In many applications, such as the circular saw cutting through a block of metal example discussed in the introduction, it is necessary to regulate the dynamic behavior between the force exerted on the environment and the manipulator motion. As opposed to the hybrid position/force control that facilitates the separate design of the position and force controllers, the hybrid impedance control exploits the Ohm's law type of relationship between force and motion; hence, the use of the “impedance” nomenclature.

The model of the environmental interactions is critical to any force control strategy [7]. In previous sections, the environment was modeled as a simple spring. However, in many applications, it can easily be seen that such a simplistic model may not capture many significant environmental interactions. To that end, to classify various types of environments, the following linear transfer function is defined:

\[
f(s) = Z_e(s) \dot{x}(s)
\]

where the variable \( s \) denotes the Laplace transform variable, \( f \) is the force exerted on the environment, \( Z_e(s) \) denotes the environmental impedance, and \( \dot{x} \) represents the velocity of the manipulator at the point of contact with the environment.

#### 16.5.1 Types of Impedance

The term \( Z_e(s) \) is referred to as the impedance because Equation (16.43) represents an Ohm's law type of relationship between motion and force. Similar to circuit theory, these environmental impedances can be separated into various categories, three of which are defined below.

**Definition 16.1** Impedance is inertial if and only if \( |Z_e(0)| = 0 \).

Figure 16.3 (a) depicts a robot manipulator moving a payload of mass \( m \) with velocity \( \dot{q} \). The interaction force is defined as

\[ f = m \ddot{q} \]

As a result, we can construct the inertial environmental impedance as follows:

\[ Z_e(s) = ms \]

**Definition 16.2** Impedance is resistive if and only if \( |Z_e(0)| = k \) where \( 0 < k < \infty \).

Figure 16.3 (b) depicts a robot manipulator moving through a liquid medium with velocity \( \dot{q} \). In this example of a resistive environment, the liquid medium applies a damping force with a damping coefficient \( b \). The interaction force is defined as

\[ f = b \dot{q} \]

which yields a resistive environmental impedance of

\[ Z_e(s) = b \]
Definition 16.3  Impedance is capacitive if and only if $|Z_e(0)| = \infty$.

A capacitive environment is depicted in Figure 16.3 (c), wherein a robot manipulator pushes against an object of mass $m$ with a velocity $\dot{q}$. A damper-spring component is assumed with a damping coefficient $b$ and a spring constant $k$. Thus, the interaction force is defined as

$$f = m\ddot{q} + b\dot{q} + kq$$

with a capacitive environmental impedance defined as

$$Z_e(s) = ms + b + \frac{k}{s}$$

From the discussion in the previous section, we have seen that the impedance is defined by a force-velocity relationship. The impedance control formulations discussed hereon will assume that the environmental impedance is inertial, resistive, or capacitive. The procedure for designing an impedance controller for any given environmental impedance is as follows. The manipulator impedance $Z_m(s)$ is selected after the environmental impedance has been modeled [8]. The selection criterion for the manipulator impedance is based on the dynamic performance of the manipulator. Specifically, the manipulator impedance is selected so as to result in zero steady-state error for a step input command of force or velocity, and this can be achieved if the manipulator impedance is the dual of the environmental impedance.

16.5.2  Duality Principle

16.5.2.1  Velocity Step-Input

For position control [8], the relationship between the velocity and force is modeled by

$$f(s) = Z_m(s)(\dot{x}_d(s) - \dot{x}(s))$$  \hspace{1cm} (16.44)

where $\dot{x}_d$ denotes the input velocity of the manipulator at the point of contact with the environment and $Z_m(s)$ represents the manipulator impedance, which is selected to achieve zero steady-state error to a step
input by utilizing the relationship between $\dot{x}_d$ and $\dot{x}$. The steady-state velocity error can be defined as

$$E_v = \lim_{s \to 0} s(\dot{x}_d(s) - \dot{x}(s))$$  \hspace{1cm} (16.45)$$

where $\dot{x}_d(s) = 1/s$ for a step velocity input. From Figure 16.4, we can reduce the above expression to

$$E_v = \lim_{s \to 0} \frac{Z_v(s)}{Z_v(s) + Z_m(s)}$$  \hspace{1cm} (16.46)$$

For $E_v$ to be zero in the above equation, $Z_v(s)$ must be noncapacitive (i.e., $|Z_v(0)| < \infty$) and $Z_m(s)$ must be a noninertial impedance (i.e., $|Z_m(0)| > 0$).

The zero steady-state error can be achieved for a velocity step-input if:

- Inertial environments are position controlled with noninertial manipulator impedances
- Resistive environments are position controlled with capacitive manipulator impedances

and the duality principle nomenclature arises from the fact that inertial environmental impedances can be position-controlled with capacitive manipulator impedances.

### 16.5.2.2 Force Step-Input

To establish the duality principle for force step-input, the dynamic relationship between force and velocity is modeled by

$$\dot{x}(s) = Z_m^{-1}(s)(f_d(s) - f(s))$$  \hspace{1cm} (16.47)$$

where $f_d$ is used to represent the input force exerted at the contact point with the environment. As outlined previously, the manipulator impedance $Z_m(s)$ is selected to achieve zero stead-state error to a step-input by utilizing the dynamic relationship between $f_d$ and $f$. To establish the duality concept, we define the steady-state force error as follows:

$$E_f = \lim_{s \to 0} s(f_d(s) - f(s))$$  \hspace{1cm} (16.48)$$

where $f_d(s) = 1/s$ for a step force input. From Figure 16.5, we can reduce the above expression to

$$E_f = \lim_{s \to 0} \frac{Z_m(s)}{Z_m(s) + Z_v(s)}$$  \hspace{1cm} (16.49)$$
For $E_a$ to be zero in the above equation, $Z_e(s)$ must be noninertial (i.e., $|Z_e(0)| > 0$) and $Z_m(s)$ must be a noncapacitive impedance (i.e., $|Z_m(0)| < \infty$).

The zero steady-state error can be achieved for a force step-input if:

- Capacitive environments are position controlled with noncapacitive manipulator impedances
- Resistive environments are position controlled with inertial manipulator impedances

and the duality principle nomenclature arises from the fact that capacitive environmental impedances can be force-controlled with inertial manipulator impedances.

### 16.5.3 Controller Design

As in the previous section, we can show that the torque controller of the form

$$\tau = M(q)J^{-1}(q)(\ddot{y} - \dot{f}(q)\dot{q}) + V_m(q, \dot{q})\dot{q} + N(q, \dot{q}) + J^T(q)f$$  \hfill (16.50)$$
yields the linear set of equations:

$$\ddot{x} = \ddot{y}$$  \hfill (16.51)$$
where $\ddot{y} \in \mathbb{R}^n$ denotes the impedance position and force control strategies.

To separate the force control design from the position control design, we define the equations to be position controlled in the task-space directions as follows:

$$\ddot{x}_{pi} = \ddot{y}_{pi}$$  \hfill (16.52)$$
where $\ddot{y}_{pi}$ denotes the $i$th position-controlled task-space direction variable.

Assuming zero initial conditions, the Laplace transform of the previous equation is given by

$$s\dot{x}_{pi}(s) = \ddot{y}_{pi}(s)$$  \hfill (16.53)$$
From the position control model defined in (16.44),

$$s\dot{x}_{pi}(s) = s\left(\ddot{y}_{pi}(s) - Z_{pmi}^{-1}(s)f_{pi}(s)\right)$$  \hfill (16.54)$$
where $Z_{pmi}$ is the $i$th position-controlled manipulator impedance. After equating Equation (16.53) and Equation (16.54), we obtain the following $i$th position controller:

$$\ddot{y}_{pi} = L^{-1}\left\{s\ddot{y}_{pi}(s) - Z_{pmi}^{-1}(s)f_{pi}(s)\right\}$$  \hfill (16.55)$$
where $L^{-1}$ denotes the inverse Laplace transform operation.

We now define the equations to be force controlled in the task-space directions as follows:

$$\ddot{x}_{ji} = \ddot{y}_{ji}$$  \hfill (16.56)$$
where $\ddot{y}_{ji}$ represents the $j$th force-controlled task-space direction variable.

Assuming zero initial conditions, the Laplace transform of the previous equation is given by

$$s\dot{x}_{ji}(s) = \ddot{y}_{ji}(s)$$  \hfill (16.57)$$
From the force control model in Equation (16.47), we can construct the following:

$$s\dot{x}_{ji}(s) = sZ_{fji}^{-1}(s)(f_{ji}(s) - f_j(s))$$  \hfill (16.58)$$
where $Z_{fji}$ represents the $j$th force controlled manipulator impedance. After equating Equation (16.57) and Equation (16.58), we obtain the following $j$th force controller:

$$\ddot{y}_{ji} = L^{-1}\left\{sZ_{fji}^{-1}(s)(f_{ji}(s) - f_j(s))\right\}$$  \hfill (16.59)$$
Example 16.4 Hybrid Impedance Control along a Slanted Surface

This example discusses the hybrid impedance control strategy for the 2-DOF Cartesian manipulator shown in Figure 16.6. The joint and surface frictions are neglected, and we assume that the normal force $f_1$ and the tangential force $f_2$, respectively, satisfy the following relationships:

$$f_1 = h_u \ddot{u} + b_v \dot{u} + k_u u$$  \hspace{1cm} (16.60)

and

$$f_2 = d_v \dot{v}$$  \hspace{1cm} (16.61)

where $h_u$, $b_v$, $k_u$, and $d_v$ are positive scalar constants. From Equation (16.50), the hybrid impedance controller is constructed as follows:

$$\tau = MJ^{-1} \ddot{y} + G + f^T f$$  \hspace{1cm} (16.62)

where $\ddot{y} \in \mathbb{R}^2$ represents the separate position and force controllers, and the quantities $\tau, J(q), G(q)$ are defined in Example 1. This torque controller decouples the robot dynamics in the task-space as follows:

$$\ddot{x} = \begin{bmatrix} \ddot{u} \\ \ddot{v} \end{bmatrix} = \begin{bmatrix} \dot{y}_f \\ \dot{y}_p \end{bmatrix} = \ddot{y}$$  \hspace{1cm} (16.63)

based on which a determination can be made as to which task-space directions should be force or position controlled.

After applying Definition 16.2, we determine that the environmental impedance in the task-space direction of $v$ is resistive in nature. Therefore, after invoking the duality principle, we can select a position controller that utilizes the capacitive manipulator impedance as follows:

$$Z_{pml}(s) = h_ms + b_m + \frac{k_m}{s}$$  \hspace{1cm} (16.64)

where $h_m, b_m, k_m$ are all positive scalar constants. Given that the task-space variable $v$ is position controlled, we can choose a position controller that satisfies the following relationship:

$$Z_{pml}(s) = h_ms + b_m + \frac{k_m}{s}$$

Copyright © 2005 by CRC Press LLC
controlled, we can formulate the following notations:

\[ \ddot{x}_{p1} = \ddot{v} = \dddot{y}_{p1} \]  
(16.65)

and

\[ f_{p1} = f_2 \]  
(16.66)

After using Equation (16.64) and the definition for \( \dddot{y}_{p1} \) in Equation (16.55), it follows that

\[ \dddot{y}_{p1} = \dddot{x}_{p1} + \frac{b_m}{h_m} (\dot{x}_{p1} - \dot{x}_{p1}) + \frac{k_m}{h_m} (x_{p1} - x_{p1}) - \frac{1}{h_m} f_{p1} \]  
(16.67)

After applying Definition 3 to the task-space direction given by \( u \), we can state that the environmental impedance is capacitive. Again, by invoking the duality principle, we can select a force controller that utilizes the inertial manipulator impedance as follows:

\[ Z_{f\text{in}}(s) = d_m s \]  
(16.68)

where \( d_m \) is a positive scalar constant. Given that the task-space variable \( u \) is to be force controlled, we can formulate the following notations:

\[ \dddot{x}_{f1} = \dddot{u} = \dddot{y}_{f1} \]  
(16.69)

and

\[ f_{f1} = f_1 \]  
(16.70)

After using Equation (16.68) and the definition for \( \dddot{y}_{f1} \) in Equation (16.59), it follows that

\[ \dddot{y}_{f1} = \frac{1}{d_m} (f_{f1} - f_{f1}) \]  
(16.71)

The overall impedance strategy is obtained by combining \( \dddot{y}_{p1} \) and \( \dddot{y}_{f1} \) defined in Equation (16.67) and Equation (16.71), respectively, which is as follows:

\[ \dddot{y} = \begin{bmatrix} \dddot{y}_{f1} \\ \dddot{y}_{p1} \end{bmatrix} \]  
(16.72)

### 16.6 Reduced Order Position/Force Control

The end effector of a rigid manipulator that is constrained by a rigid environment cannot move freely through the environment and has its degrees of freedom reduced. As a result, at least one degree of freedom with respect to position is lost. This occurs at the point of contact with the environment (that is, when the environmental constraint is engaged) when contact (interaction) forces develop. Thus, the reduction of positional degrees of freedom while developing interaction forces motivates the design of position/force controllers that incorporate this phenomenon.

#### 16.6.1 Holonomic Constraints

For the reduced order control strategy discussed in this section, an assumption about the environment being holonomic and frictionless is made. Specifically, we assume that a constraint function \( \psi(q) \in \mathbb{R}^p \) exists in joint-space coordinates that satisfies the following condition that establishes holonomic environmental constraints:

\[ \psi(q) = 0 \]  
(16.73)
where the dimension of the constrained function is assumed to be smaller than the number of joints in the manipulator (i.e., \( p < n \)). The structure of a constraint function is related to the robot kinematics and the environmental configuration.

The constrained robot dynamics for an \( n \)-link robot manipulator with holonomic and frictionless constraints are described by

\[
M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + F(q) + G(q) + A^T(q)\lambda = \tau
\]

where \( \lambda \in \mathbb{R}^p \) represents the generalized force multipliers associated with the constraints and the constrained Jacobian matrix \( A(q) \in \mathbb{R}^{p \times n} \) is defined by

\[
A(q) = \frac{\partial \bar{\psi}(q)}{\partial q}
\]

### 16.6.2 Reduced Order Model

An \( n \)-joint robot manipulator constrained by a rigid environment has \( n - p \) degrees of freedom. However, the joint-space model of Equation (16.74) contain \( n \) position variables (i.e., \( q \in \mathbb{R}^n \)) which when combined with the \( p \) force variables (i.e., \( \lambda \in \mathbb{R}^p \)) result in \( n + p \) controlled states. As a result, the controlled states are greater than the number of inputs. To rectify this situation, a variable transformation can be used to reduce the control variables from \( n + p \) to \( n \).

The constrained robot model can be reduced by assuming that a function \( g(x) \in \mathbb{R}^n \) exists that relates the constrained space vector \( x \in \mathbb{R}^{n-p} \) to the joint-space vector \( q \in \mathbb{R}^p \) and is given by

\[
q = g(x)
\]

where \( g(x) \) is selected to satisfy

\[
\left[ \frac{\partial g(x)}{\partial x} \right]^T A^T(q) \bigg|_{q=g(x)} = 0
\]

and the Jacobian matrix \( \Sigma(x) \in \mathbb{R}^{n \times (n-p)} \) is defined as

\[
\Sigma(x) = \frac{\partial g(x)}{\partial x}
\]

The reader is referred to the work presented by McClamroch and Wang in [9] for a detailed analysis that establishes the above arguments. The reduced order model is constructed as follows [4, 9]:

\[
M(x)\dot{x} + N(x, \dot{x}) + A^T(x)\lambda = \tau
\]

where

\[
N(x, \dot{x}) = (V_m(x, \dot{x})\Sigma(x) + M(x)\dot{\Sigma}(x))\dot{x} + F(x, \dot{x}) + G(x)
\]

### 16.6.3 Reduced Order Controller Design

To design a position/force controller based on the reduced order model, we first define the constrained position tracking error and the force multiplier tracking error, respectively, as follows:

\[
\dot{x} = x_d - x
\]

and

\[
\dot{\lambda} = \lambda_d - \lambda
\]
It is important to note that the desired constrained space position trajectory $x_d$ and its first two time derivatives are bounded and that the desired force multiplier trajectory $\lambda_d$ is known and bounded.

Based on the structure of the error system and the control objective, the feedback linearizing reduced order position/force controller [9] is defined as follows:

$$
\tau = M(x)\Sigma(x)(\dot{x}_d + K_v\dot{x} + K_p\ddot{x}) + N(x, \dot{x}) + A^T(x)(\lambda_d + K_f\dot{\lambda})
$$

(16.83)

where $K_v, K_p \in \mathbb{R}^{(n-p) \times (n-p)}$, and $K_f \in \mathbb{R}^{p \times p}$ are diagonal and positive definite matrices. After constructing the closed-loop error system, the following asymptotic stability result for the constrained position tracking error and the constrained force tracking error can be obtained

$$
\lim_{t \to \infty} \ddot{\tilde{x}}, \dot{\tilde{x}}, \tilde{x} = 0
$$

(16.84)

and

$$
\lim_{t \to \infty} \tilde{\lambda} = 0
$$

(16.85)

**Example 16.5 Reduced Order Position/Force Control along a Slanted Surface**

This example discusses the reduced order control strategy for the 2-DOF Cartesian manipulator shown in Figure 16.7, where the joint and surface friction is neglected. As given in Example 1, the constrained function for this system is given by

$$
\dot{\psi}(q) = q_1 - q_2 - 3 = 0
$$

(16.86)

Based on the structure of Equation (16.79) and Equation (16.80), the manipulator dynamics on the constrained surface can be formulated as follows:

$$
M\ddot{q} + G + A^T\lambda = \tau
$$

(16.87)

![FIGURE 16.7 Reduced order position/force controller [4].](image-url)
where \( M, q, G, \) and \( \tau \) are defined in Example 1 and

\[
A^T = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \tag{16.88}
\]

For this problem, we assume that \( x = q_1 \), and as per Equation (16.76), we must find a function \( g(x) \) that satisfies \( q = g(x) \). For holonomic constraints, we can verify that following kinematic relationships hold:

\[
\begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} x \\ x - 3 \end{bmatrix} \tag{16.89}
\]

From the definition of the Jacobian in Equation (16.78), we obtain

\[
\Sigma = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{16.90}
\]

and the reduced order position/force controller is defined as follows:

\[
\tau = M \Sigma (\dddot{x}_d + K_v \dot{x} + K_p \dot{x}) + G + A^T (\dot{\lambda}_d + K_f \dot{\lambda}) \tag{16.91}
\]

where \( x_d \) represents the desired trajectory of \( q_1 \) and \( \lambda_d \) represents the desired force multiplier.

For this problem, we can scrutinize the relationship between \( \lambda \) and the normal force exerted on the surface by equating the expressions for the forces in this example and that in Example 1 as follows:

\[
A^T \lambda = f^T f \tag{16.92}
\]

where \( f \) and \( f \) are defined in Example 1. Given that the surface friction has been neglected, from the above equation, it follows that:

\[
\lambda = \frac{f_1}{\sqrt{2}} \tag{16.93}
\]

where \( f_1 \) is the normal force exerted on the surface. Similarly, from kinematic relationships, we observe that

\[
q_1 = \frac{\sqrt{2}}{2} v + \frac{3}{2} \tag{16.94}
\]

where \( v \) denotes the end-effector position measured along the surface. The significance of the expressions in Equation (16.93) and Equation (16.94) lies in the fact that they would be used for trajectory generation because the position and force control objectives are formulated in terms of \( f_1 \) and \( v \). Specifically, \( \lambda_d \) is obtained from the desired normal force \( f_{d1} \) in Equation (16.93) and \( q_{d1} \) is obtained from the desired end-effector surface position \( v_{d1} \) in Equation (16.94).
16.7 Background and Further Reading

Over the last three decades, many researchers have developed a variety of force control algorithms for robot manipulators. Raibert and Craig in [6] originally proposed the hybrid position/force control approach. Khatib and Burdik [10] incorporated dynamic coupling effects between the robot joints. In [9], McClamroch and Wang developed a reduced-order model by applying nonlinear transformation to the constrained robot dynamics, which facilitated the separate design of position and force control strategies. For a comprehensive review and comparison of [9] with other position/force control strategies, the reader is referred to Grabbe et al. [11].

More advanced adaptive control strategies (not discussed in this chapter) have also been developed for position and force control, such as in [12], where Carelli and Kelly designed adaptive controllers that ensured asymptotic position tracking but only bounded force tracking error (the reader is referred to [13] and [14] for similar work). Adaptive full-state feedback controllers designed in Arimoto et al. [15], Yao and Tomizuka [16], and Zhen and Goldenberg [17] achieved a stronger stability result of asymptotic position and force tracking. Later de Queiroz et al. [18] designed partial-state feedback adaptive position/force controllers that compensated for parametric uncertainty in the manipulator dynamics while eliminating the need for link velocity measurements. In [19], Lanzon and Richards designed robust trajectory/force controllers for non-redundant rigid manipulators using sliding-mode and adaptive control techniques.

Furthermore, position/force control strategies are also supporting other critical technology initiatives. For example, in the field of medical robotics, the UC Berkeley/UCSF Telesurgical Workstation controller [20] is designed to incorporate the motion and force requirements of complex tasks such as tying a knot and suturing (e.g., a sufficiently large force is required to manipulate the suture through tissue). Another interesting application is the use of position/force control strategies in visual servo applications of robot manipulators [21].

In summary, we have presented several position/force control strategies for rigid robot manipulators. It is important to note that while this chapter presents the development of position/force controllers in chronological order, many other sophisticated control strategies have been developed and implemented (some of which have been cited in this section). With the advances in computer processing technology and real-time control, researchers in the robotics and control engineering fields will be able to design strategies that capture and process more information about the environment and compensate for uncertainties in the manipulator model. This would certainly result in significant performance enhancements and as a result, the use of manipulators in force control applications will be more common.

References


