Chapter 9
Force and Compliance Controls

A class of simple tasks may need only trajectory control where the robot end-effecter is moved merely along a prescribed time trajectory. However, a number of complex tasks, including assembly of parts, manipulation of tools, and walking on a terrain, entail the control of physical interactions and mechanical contacts with the environment. Achieving a task goal often requires the robot to comply with the environment, react to the force acting on the end-effecter, or adapt its motion to uncertainties of the environment. Strategies are needed for performing those tasks.

Force and compliance controls are fundamental task strategies for performing a class of tasks entailing the accommodation of mechanical interactions in the face of environmental uncertainties. In this chapter we will first present *hybrid position/force control*: a basic principle of strategic task planning for dealing with geometric constraints imposed by the task environment. An alternative approach to accommodating interactions will also be presented based on *compliance or stiffness control*. Characteristics of task compliances and force feedback laws will be analyzed and applied to various tasks.

9.1 Hybrid Position/Force Control

9.1.1 Principle

To begin with let us consider a daily task. Figure 9.1.1 illustrates a robot drawing a line with a pencil on a sheet of paper. Although we humans can perform this type of task without considering any detail of hand control, the robot needs specific control commands and an effective control strategy. To draw a letter, "A", for example, we first conceive a trajectory of the pencil tip, and command the hand to follow the conceived trajectory. At the same time we accommodate the pressure with which the pencil is contacting the sheet of paper. Let *o-xyz* be a coordinate system with the *z*-axis perpendicular to the sheet of paper. Along the *x* and *y* axes, we provide positional commands to the hand control system. Along the *z*-axis, on the other hand, we specify a force to apply. In other words, controlled variables are different between the horizontal and vertical directions. The controlled variable of the former is *x* and *y* coordinates, i.e. a position, while the latter controlled variable is a force in the *z* direction. Namely, two types of control loops are combined in the hand control system, as illustrated in Figure 9.1.2.

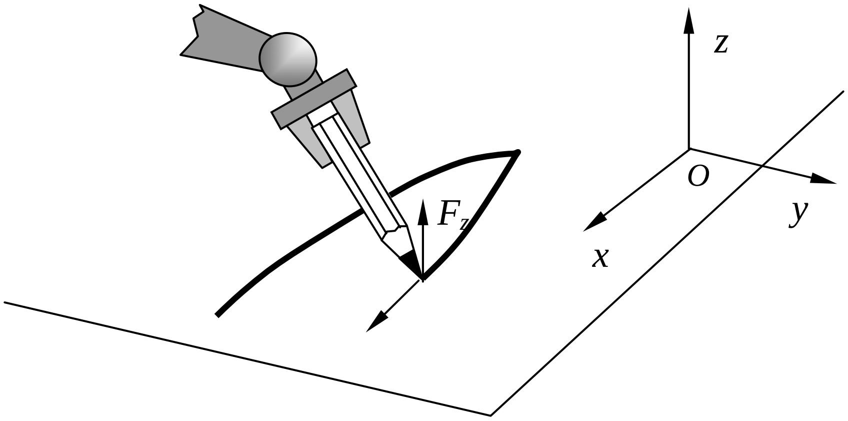

Figure 9.1.1 Robot drawing a line with a pencil on a sheet of paper

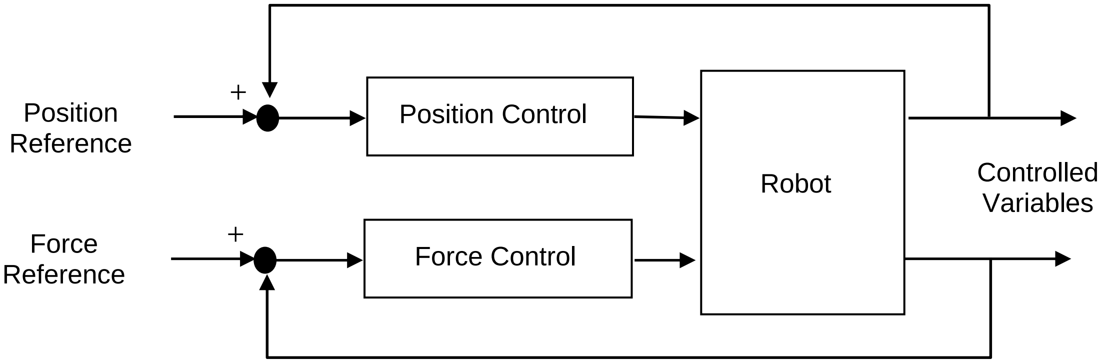

Figure 9.1.2 Position and force control loops

The above example is one of the simplest tasks illustrating the need for integrating different control loops in such a way that the control mode is consistent with the geometric constraint imposed to the robot system. As the geometric constraint becomes more complex and the task objective is more involved, an intuitive method may not suffice. In the following we will obtain a general principle that will help us find proper control modes consistent with both geometric constraints and task objectives. Let us consider the following six-dimensional task to derive a basic principle behind our heuristics and empiricism.

Example 9.1

Shown below is a task to pull up a peg from a hole. We assume that the peg can move in the vertical direction without friction when sliding in the hole. We also assume that the task process is quasi-static in that any inertial force is negligibly small. A coordinate system *O-xyz*, referred to as *C-frame*, is attached to the task space, as shown in the figure. The problem is to find a proper control mode for each of the axes: three translational and three rotational axes.

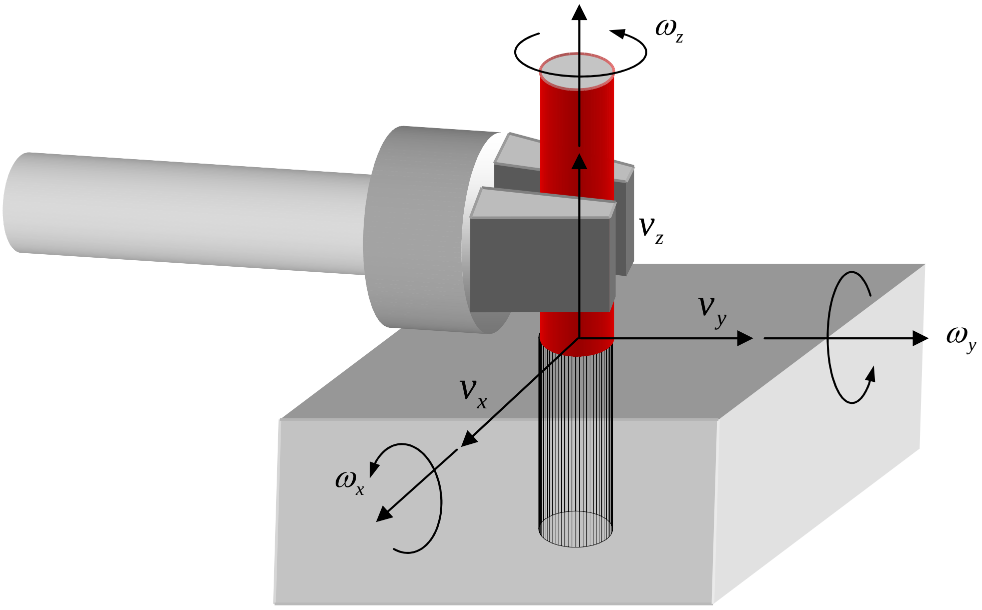

Figure 9.1.3 Pulling up a peg from a hole

The key question is how to assign a control mode, position control or force control, to each of the axes in the C-frame in such a way that the control action may not conflict with the geometric constraints and physics. M. Mason addressed this issue in his seminal work on hybrid position/force control. He called conditions dictated by physics *Natural Constraints*, and conditions determined by task goals and objectives *Artificial Constraints.* Table 9.1.1 summarizes these conditions.

From Figure 9.1.3 it is clear that the peg cannot be moved in the *x* and *y* directions due to the geometric constraint. Therefore, the velocities in these directions must be zero: $v_x = 0, v_y = 0$. Likewise, the peg cannot be rotated about the *x* and *y* axes. Therefore, the angular velocities are zero: $\omega_x = 0, \omega_y = 0$. These conditions constitute the natural constraints in the kinematic domain. The remaining directions are linear and angular *z* axes. Velocities along these two directions can be assigned arbitrarily, and may be controlled with position control mode. The reference inputs to these position control loops must be determined such that the task objectives may be satisfied. Since the task is to pull up the peg, an upward linear velocity must be given: $v_z = V > 0$. The orientation of the peg about the z-axis, on the other hand, doesn't have to be changed. Therefore, the angular velocity remains zero: $\omega_z = 0$. These reference inputs constitute the artificial constraints in the kinematic domain.

Table 9.1.1 Natural and artificial constraints of the peg-in-the-hole problem

	Kinematic	Static
Natural Constraints	$v_x = 0$ $v_y = 0$ $\omega_x = 0$ $\omega_y = 0$	$f_z = 0$ $\tau_z = 0$
Artificial Constraints	$v_z = V > 0$ $\omega_z = 0$	$f_x = 0$ $f_y = 0$ $\tau_x = 0$ $\tau_y = 0$

In the statics domain, forces and torques are specified in such a way that the quasi-static condition is satisfied. This means that the peg motion must not be accelerated with any unbalanced force, i.e. non-zero inertial force. Since we have assumed that the process is friction-less, no resistive force acts on the peg in the direction that is not constrained by geometry. Therefore, the linear force in the *z* direction must be zero: $f_z = 0$. The rotation about the *z* axis, too, is not constrained. Therefore, the torque about the *z* axis must be zero: $\tau_z = 0$. These conditions are dictated by physics, and are called the natural constraints in the statics domain. The remaining directions are geometrically constrained. In these directions, forces and torques can be assigned arbitrarily, and may be controlled with force control mode. The reference inputs to these control loops must be determined so as to meet the task objectives. In this task, it is not required to push the peg against the wall of the hole, nor twist it. Therefore, the reference inputs are set to zero: $f_x = 0,\ f_y = 0,\ \tau_x = 0,\ \tau_y = 0$. These constitute the artificial constraints in the statics domain.

In the above example, it is clear that the axes involved in the natural constraints and the artificial constraints are orthogonal to each other in both kinematic and static domains. Moreover, the axes involved in the natural kinematic constraints and the artificial static constraints are the same, and the ones listed in the natural static constraints and the artificial kinematic constraints are the same. These relationships are rather obvious in the above example where the direction of each C-frame axis is aligned with the direction along which each control mode, position or force, is assigned. If such a C-frame exists, these orthogonality properties are simply the consequence of the following assumptions and rule:

- Each C-frame axis must have only one control mode, either position or force,
- The process is quasi-static and friction less, and
- The robot motion must conform to geometric constraints.

In general, the axes of a C-frame are not necessarily the same as the direction of a separate control mode. Nevertheless, the orthogonality properties hold in general. We prove this next.

Let V^6 be a six-dimensional vector space, and $V_a \subset V^6$ be an admissible motion space, that is, the entire collection of admissible motions conforming to the geometric constraints involved in a given task. Let V_c be a constraint space that is the orthogonal complement to the admissible motion space:

$$V_c = V_a^{\perp} \tag{9.1.1}$$

Let $\mathbf{F} \in V^6$ be a six-dimensional endpoint force acting on the end-effecter, and $\Delta\mathbf{p} \in V^6$ be an infinitesimal displacement of the end-effecter. The work done on the end-effecter is given by

$$\Delta Work = \mathbf{F}^T \Delta\mathbf{p} \tag{9.1.2}$$

Decomposing each vector to the one in the admissible motion space and the one in the constraint space,

$$\begin{aligned} &\mathbf{F} = \mathbf{F}_a + \mathbf{F}_c; \qquad && \mathbf{F}_a \in V_a,\ \mathbf{F}_c \in V_c \\ &\Delta\mathbf{p} = \Delta\mathbf{p}_a + \Delta\mathbf{p}_c; \quad && \Delta\mathbf{p}_a \in V_a,\ \Delta\mathbf{p}_c \in V_c \end{aligned} \tag{9.1.3}$$

and substituting them to eq.(2) yield

$$\begin{aligned} \Delta Work &= (\mathbf{F}_a + \mathbf{F}_c)^T (\Delta\mathbf{p}_a + \Delta\mathbf{p}_c) = \mathbf{F}_a{}^T \Delta\mathbf{p}_a + \mathbf{F}_a{}^T \Delta\mathbf{p}_c + \mathbf{F}_c{}^T \Delta\mathbf{p}_a + \mathbf{F}_c{}^T \Delta\mathbf{p}_c \\ &= \mathbf{F}_a{}^T \Delta\mathbf{p}_a + \mathbf{F}_c{}^T \Delta\mathbf{p}_c \end{aligned} \tag{9.1.4}$$

since $\mathbf{F}_a \perp \Delta\mathbf{p}_c,\quad \mathbf{F}_c \perp \Delta\mathbf{p}_a$ by definition. For the infinitesimal displacement $\Delta\mathbf{p}$ to be a virtual displacement $\delta\mathbf{p}$, its component in the constraint space must be zero: $\Delta\mathbf{p}_c = 0$. Then, $\Delta\mathbf{p}_a = \delta\mathbf{p}$ becomes a virtual displacement, and eq.(4) reduces to virtual work. Since the system is in a static equilibrium, the virtual work must vanish for all virtual displacements $\delta\mathbf{p}_a$.

$$\delta Work = \mathbf{F}_a{}^T \delta\mathbf{p}_a = 0, \quad \forall \delta\mathbf{p}_a \tag{9.1.5}$$

This implies that any force and moment in the admissible motion space must be zero, i.e. the natural static constraints:

$$0 = \mathbf{F}_a \in V_a \tag{9.1.6}$$

Furthermore, the properties of artificial static constraints can be derived from eqs.(4) and (5). Since in eq.(4) $\Delta\mathbf{p}_c = 0$, the static equilibrium condition holds, although $\mathbf{F}_c \in V_c$ takes an arbitrary value. This implies that to meet a task goal we can assign arbitrary values to the force and moment in the constraint space, i.e. the artificial static constraints.

$$arbitrary : \mathbf{F}_c \in V_c \tag{9.1.7}$$

Converting infinitesimal displacements to velocities, $\dot{\mathbf{p}}_a, \dot{\mathbf{p}}_c$, we can obtain the natural and artificial kinematic constraints:

$$\begin{aligned} &arbitrary : \dot{\mathbf{p}}_a \in V_a, \\ &0 = \dot{\mathbf{p}}_c \in V_c \end{aligned} \tag{9.1.8}$$

Table 9.1.2 summarizes the above results.

Table 9.1.2 Mason’s Principle of Hybrid Position/Force Control

	Kinematic	Static
Natural Constraints	$0 = \dot{\mathbf{p}}_c \in V_c$	$0 = \mathbf{F}_a \in V_a$
Artificial Constraints	$arbitrary : \dot{\mathbf{p}}_a \in V_a$	$arbitrary : \mathbf{F}_c \in V_c$

The reader will appreciate Mason’s Principle when considering the following exercise problem, in which the partition between admissible motion space and constraint space cannot be described by a simple division of *C*-frame axes. Rather the admissible motion space lies along an axis where a translational axis and a rotational axis are coupled.

Exercise 9.2 (The same as PS)

9.1.2 Architecture of Hybrid Position/Force Control System

Based on Mason’s Principle, a hybrid position/force control system can be constructed in such a way that the robot control system may not have a conflict with the natural constraints of the task process, while performing the task towards the task goal. Figure 9.1.5 shows the block diagram of a hybrid position/force control system. The upper half of the diagram is position control loops, where artificial kinematic constraints are provided as reference inputs to the system and are compared with the actual position of the end-effecter. The lower half of the diagram is force control loops, where artificial static constraints are provided as reference inputs to the feedback loops and are compared with the actual force and moment at the end-effecter. Note that feedback signals are described in an appropriate *C*-frame attached to the end-effecter.

If the feedback signals are noise-less and the *C*-frame is perfectly aligned with the actual task process, the position signal of the feedback loop must lie in the admissible motion space, and the force being fed back must lie in the constraint space. However, the feedback signals are in general corrupted with sensor noise and the *C*-frame may be misaligned. Therefore, the position signal may contain some component in the constraint space, and some fraction of the force signal may be in the admissible motion space. These components are contradicting with the natural constraints, and therefore should not be fed back to the individual position and force controls. To filter out the contradicting components, the feedback errors are projected to their own subspaces, i.e. the positional error $\mathbf{e}_\mathrm{p}$ mapped to the admissible motion space V_a and the force feedback error $\mathbf{e}_\mathrm{f}$ mapped to the constraint space V_c. In the block diagram these filters are shown by projection matrices, $\mathbf{P}_\mathrm{a}$ and $\mathbf{P}_\mathrm{c}$:

$$\widehat{\mathbf{e}}_p = \mathbf{P}_a \mathbf{e}_p, \quad \widehat{\mathbf{e}}_f = \mathbf{P}_c \mathbf{e}_f \tag{9.1.9}$$

When the C-frame axes are aligned with the directions of the individual position and force control loops, the projection matrices are diagonal, consisting of only 1 and 0 in the diagonal components. In the case of the peg-in-the-hole problem, they are:

$$\mathbf{P}_a = diag(1\,1\,0\,1\,1\,0), \quad \mathbf{P}_c = diag(0\,0\,1\,0\,0\,1) \tag{9.1.10}$$

In case of Example 9.2 where the C-frame axes are not the directions of the individual position and force control loops, the projection matrices are not diagonal.

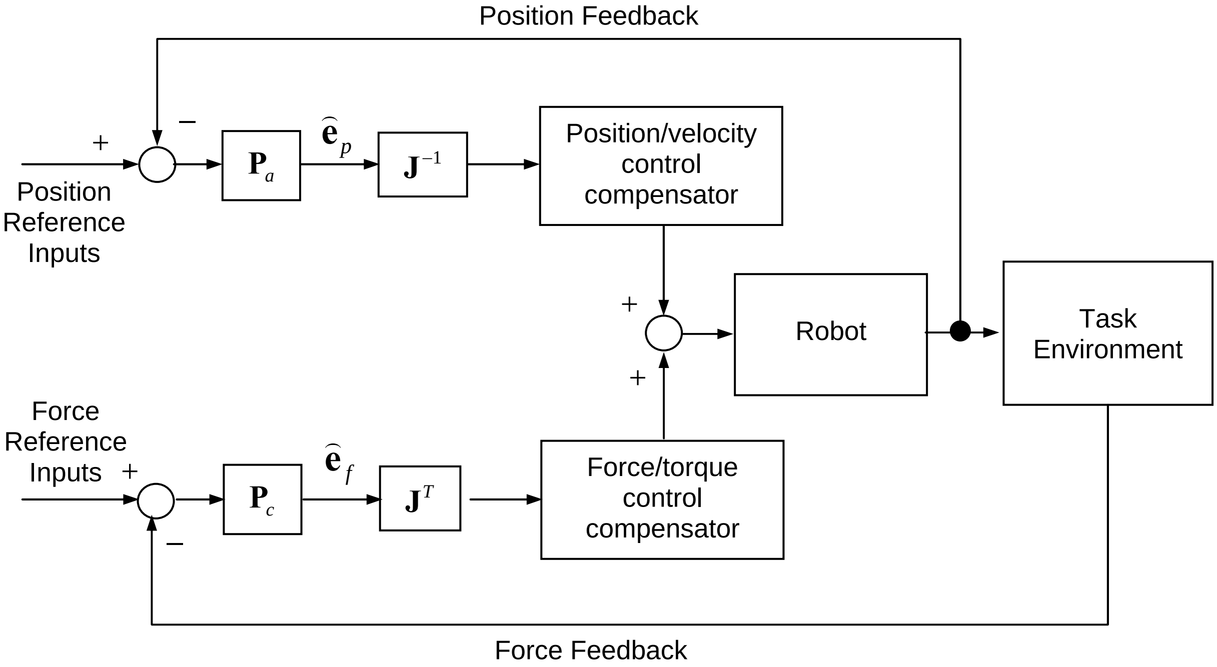

Figure 9.1.4 Block diagram of hybrid position/force control system

These feedback errors, $\hat{\mathbf{e}}_p$ and $\hat{\mathbf{e}}_f$, are in the C-frame, hence they must be converted to the joint space in order to generate control commands to the actuators. Assuming that the positional error vector is small and that the robot is not at a singular configuration, the position feedback error in joint coordinates is given by

$$\mathbf{e}_q = \mathbf{J}^{-1}\hat{\mathbf{e}}_p \tag{9.1.11}$$

where **J** is the Jacobian relating the end-effecter velocities in the C-frame to joint velocites. The force feedback error in the joint coordinates, on the other hand, is obtained based on the duality principle:

$$\mathbf{e}_\tau = \mathbf{J}^T\,\hat{\mathbf{e}}_f \tag{9.1.12}$$

These two error signals in the joint coordinates are combined after going through dynamic compensation in the individual joint controls.

9.2 Compliance Control

9.2.1 Task strategy

Use of both position and force information is a unique feature in the control of robots physically interacting with the environment. In hybrid position/force control, separation was made explicitly between position and force control loops through projections of feedback signals onto admissible motion space and constraint space. An alternative to this space separation architecture is to control a *relationship* between position and force in the task space. *Compliance Control* is a basic control law relating the displacement of the end-effecter to the force and moment acting on it. Rather than totally separating the task space into subspaces of either position or force control, compliance control reacts to the endpoint force such that a given functional relationship, typically a linear map, is held between the force and the displacement. Namely, a functional relationship to generate is given by

$$\Delta\mathbf{p} = \mathbf{C}\mathbf{F} \tag{9.2.1}$$

where **C** is an *m* x *m* Compliance Matrix, and $\Delta\mathbf{p}$ and $\mathbf{F}$ are endpoint displacement and force represented in an *m*-dimensional, task coordinate system. Note that the inverse to the compliance matrix is a stiffness matrix:

$$\mathbf{K} = \mathbf{C}^{-1} \tag{9.2.2}$$

if the inverse exists.

The components of the compliance matrix, or the stiffness matrix, are design parameters to be determined so as to meet task objectives and constraints. Opening a door, for example, can be performed with the compiance illustrated in Figure 9.2.1. The trajectory of the doorknob is geometrically constrained to the circle of radius R centered at the door hinge. The robot hand motion must comply to the constrained doorknob trajectory, although the trajectory is not exactly known. The robot must not break the doorknob, although the conceived trajectory is different from the actual trajectory. This task requirement can be met by assigining a small stiffness, i.e. a high compliance, to the radial direction perpendicular to the trajectory. As illustrated in the figure, such a small spring constant generates only a small restoring force in response to the discrepancy between the actual doorknob trajectory and the reference trajectory of the robot hand. Along the direction tangent to the doorknob trajectory, on the other hand, a large stiffness, or a small compliance, is assigned. This is to force the doorknob to move along the trajectory despite friction and other resistive forces. The stiffness matrix is therefore given by

$$\mathbf{K} = \begin{pmatrix} k_x & 0 \\ 0 & k_y \end{pmatrix}; \quad k_x << 1, k_y >> 1 \tag{9.2.3}$$

with reference to the task coordinate system *O-xy*. Using this stiffness with which the doorknow is held, the robot can open the door smoothly and dexterously, although the exact trajectory of the doorknob is not known.

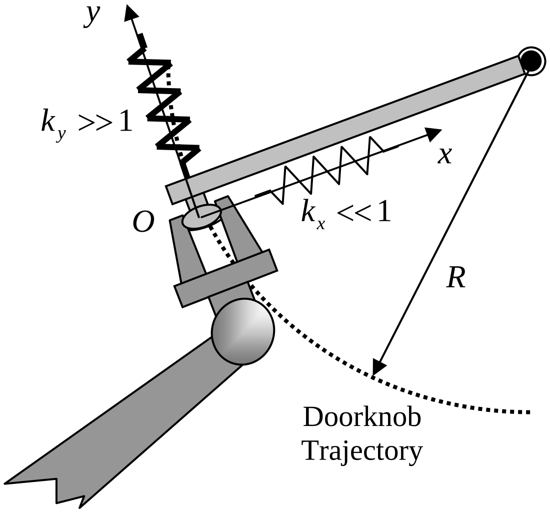

Figure 9.2.1 Door opening with compliance control

9.2.2 Compliance control synthesis

Now that a desired compliance is given, let us consider the method of generating the desired compliance. There are multiple ways of synthesizing a compliance control system. The simplest method is to accommodate the proportional gains of joint feedback controls so that desired restoring forces are generated in proportion to discrepancies between the actual and reference joint angles. As shown in Figure 9.2.2, a feedback control error e_i is generated when a disturbance force or torque acts on the joint. At steady state a ststic balance is made, as an actuator torque τ_i proportional to the control error e_i cancels out the disturbance torque. The proportionality constant is determined by the position feedback gain k_i, when friction is neglected. Therefore a desired stiffness or compliance can be obtained by tuning the position feedback gain.

Compliance synthesis is trivial for single joint control systems. For general *n* degree-of-freedom robots, however, multiple feedback loops must be coordinated. We now consider how to generate a desired *m x m* compliance or stiffness *matrix* specified at the endpoint by tuning joint feedback gains.

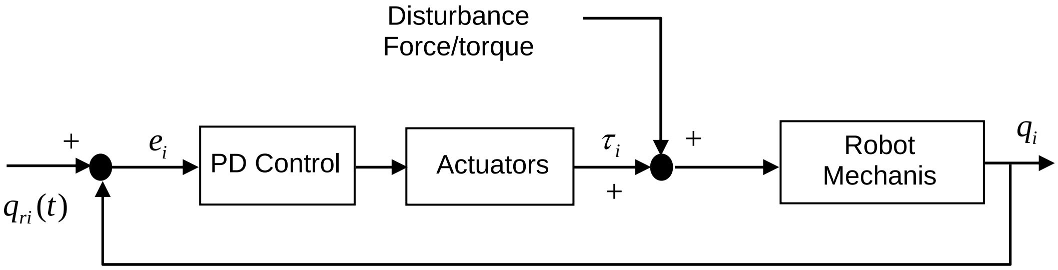

Figure 9.2.2 Single joint position feedback control system

Theorem Let **J** be the Jacobian relating endpoint velocity $\dot{\mathbf{p}} \in R^{m\,x1}$ to joint velocities $\dot{\mathbf{q}} \in R^{n\,x1}$, and $\boldsymbol{\tau} \in R^{n\,x1}$ be joint torques associated with joint coordinates **q**. Let $\Delta\mathbf{p} \in R^{m\,x1}$ be a *m* x 1 vector of the endpoint displacement measured from a nominal position $\overline{\mathbf{p}}$, and $\mathbf{F} \in R^{m\,x1}$ be the endpoint force associated with the endpoint displacement $\Delta\mathbf{p}$. Let $\mathbf{K_p}$ be a desired endpoint stiffness matrix defined as:

$$\mathbf{F} = \mathbf{K}_p \Delta\mathbf{p} \tag{9.2.4}$$

The necessary condition for joint feedback gain $\mathbf{K_q}$ to generate the endpoint stiffness $\mathbf{K_p}$ is given by

$$\mathbf{K}_q = \mathbf{J}^T \mathbf{K}_p \mathbf{J} \tag{9.2.5}$$

assuming no friction at the joints and linkage mechanisms.

Proof

Using the Jacobian and the duality principle as well as eq.(4),

$$\boldsymbol{\tau} = \mathbf{J}^T \mathbf{F} = \mathbf{J}^T \mathbf{K}_p \Delta\mathbf{p} = \mathbf{J}^T \mathbf{K}_p \mathbf{J} \cdot \Delta\mathbf{q} \tag{9.2.6}$$

Using eq.(5), the above relationship reduces to

$$\boldsymbol{\tau} = \mathbf{K}_q \Delta\mathbf{q} \tag{9.2.7}$$

This implies that $\mathbf{K}_q$ is the joint feedback gain matrix.

Example 9.2.1 Consider a two-link, planar robot arm with absolute joint angles and joint torques, as shown in Figure 9.2.3. Obtain the joint feedback gain matrix producing the endpoint stiffness $\mathbf{K}_p$:

$$\mathbf{K}_p = \begin{pmatrix} k_1 & 0 \\ 0 & k_2 \end{pmatrix} \tag{9.2.8}$$

Assuming that the link length is 1 for both links, the Jacobian is given by

$$\mathbf{J} = \begin{pmatrix} -s_1 & -s_2 \\ c_1 & c_2 \end{pmatrix} \qquad (9.2.9)$$

From eq.(5),

$$\mathbf{K}_q = \mathbf{J}^T \mathbf{K}_p \mathbf{J} = \begin{pmatrix} k_{q1} & k_{q3} \\ k_{q3} & k_{q2} \end{pmatrix} \qquad (9.2.10)$$

where

$$\begin{aligned} k_{q1} &= k_1 s_1^2 + k_2 c_1^2 \\ k_{q2} &= k_1 s_2^2 + k_2 c_2^2 \\ k_{q3} &= k_1 s_1 s_2 + k_2 c_1 c_2 \end{aligned} \qquad (9.2.11)$$

Note that the joint feedback gain matrix $\mathbf{K}_q$ is symmetric and that the matrix $\mathbf{K}_q$ degenerates when the robot is at a singular configuration. If it is non-singular, then

$$\Delta\mathbf{p} = \mathbf{J}\Delta\mathbf{q} = \mathbf{J}\mathbf{K}_q^{-1}\boldsymbol{\tau} = \mathbf{J}\mathbf{K}_q^{-1}\mathbf{J}^T\mathbf{F} = \mathbf{J}(\mathbf{J}^T\mathbf{K}_p\mathbf{J})^{-1}\mathbf{J}^T\mathbf{F} = \mathbf{K}_p^{-1}\mathbf{F} = \mathbf{C}\mathbf{F} \qquad (9.2.12)$$

Therefore, the obtained joint feedback gain provides the desired endpoint stiffness given by eq.(8), or the equivalent compliance.

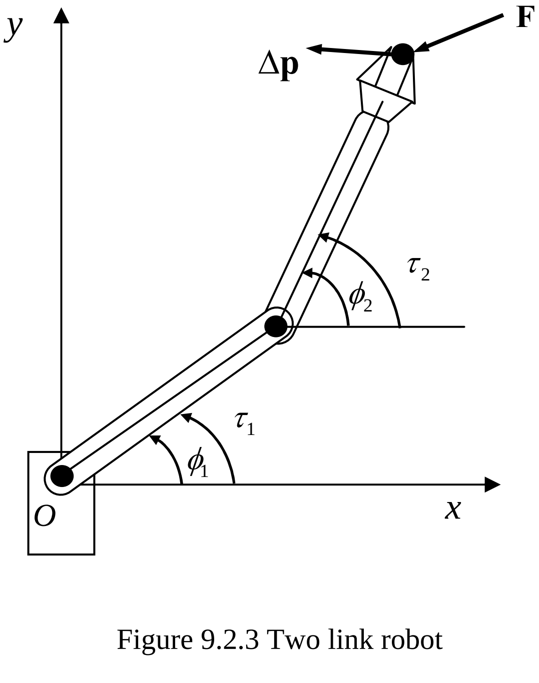

Figure 9.2.3 Two link robot