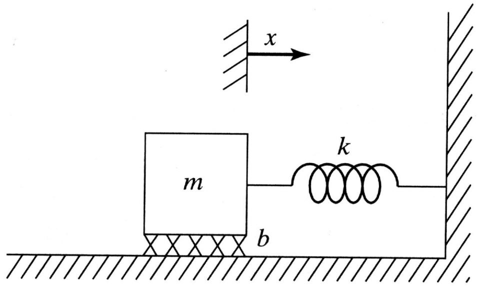


$$m\ddot{x} + b\dot{x} + kx = 0 \qquad (9.3)$$

考虑 物理模型 → 力分析

→ 数学模型

→

From the study of differential equations [1], we know that the form of the solution to an equation of the form of (9.3) depends on the roots of its **characteristic equation**,

$$ms^2 + bs + k = 0.$$

This equation has the roots

$$s_1 = -\frac{b}{2m} + \frac{\sqrt{b^2 - 4mk}}{2m},$$

$$s_2 = -\frac{b}{2m} - \frac{\sqrt{b^2 - 4mk}}{2m}.$$

System Responses (3 Solutions)

1. **Real and Unequal Roots**. This is the case when $b^2 > 4mk$; that is, friction dominates, and sluggish behavior results. This response is called **overdamped**.
2. **Complex Roots**. This is the case when $b^2 < 4mk$; that is, stiffness dominates, and oscillatory behavior results. This response is called **underdamped**.
3. **Real and Equal Roots**. This is the special case when $b^2 = 4mk$; that is, friction and stiffness are "balanced," yielding the fastest possible nonoscillatory response. This response is called **critically damped**.

The third case (critical damping) is generally a desirable situation: the system nulls out nonzero initial conditions and returns to its nominal position as rapidly as possible, yet without oscillatory behavior.

Real and unequal roots

It can easily be shown (by direct substitution into (9.3)) that the solution, $x(t)$, giving the motion of the block in the case of real, unequal roots has the form

$$x(t) = c_1 e^{s_1 t} + c_2 e^{s_2 t}, \qquad (9.6)$$

Complex Roots solution

Complex roots

For the case where the characteristic equation has complex roots of the form

$$s_1 = \lambda + \mu i,$$
$$s_2 = \lambda - \mu i, \tag{9.11}$$

it is still the case that the solution has the form

$$x(t) = c_1 e^{s_1 t} + c_2 e^{s_2 t}. \tag{9.12}$$

However, equation (9.12) is difficult to use directly, because it involves imaginary numbers explicitly. It can be shown (see Exercise 9.1) that **Euler's formula**,

$$e^{ix} = \cos x + i \sin x, \tag{9.13}$$

allows the solution (9.12) to be manipulated into the form

$$x(t) = c_1 e^{\lambda t} \cos(\mu t) + c_2 e^{\lambda t} \sin(\mu t). \tag{9.14}$$

As before, the coefficients c_1 and c_2 are constants that can be computed for any given set of initial conditions (i.e., initial position and velocity of the block). If we write the constants c_1 and c_2 in the form

$$c_1 = r \cos \delta,$$
$$c_2 = r \sin \delta, \tag{9.15}$$

Closed-Loop System

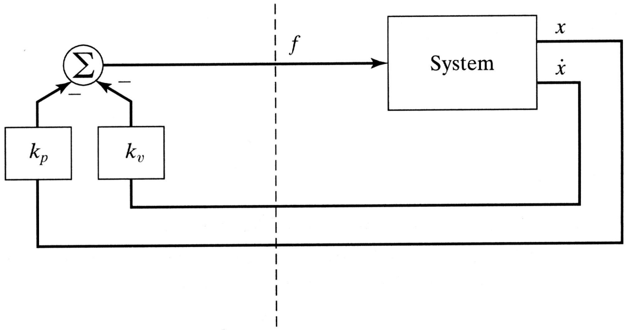


FIGURE 9.7: A closed-loop control system. The control computer (to the left of the dashed line) reads sensor input and writes actuator output commands.

of disturbance forces applied to the block. In a later section, we will construct a **trajectory-following** control system, which can cause the block to follow a desired position trajectory.

By equating the open-loop dynamics of (9.34) with the control law of (9.35), we can derive the closed-loop dynamics as

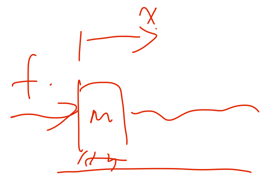
$$m\ddot{x} + b\dot{x} + kx = -k_p x - k_v \dot{x}, \tag{9.36}$$

or

$$m\ddot{x} + (b + k_v)\dot{x} + (k + k_p)x = 0, \tag{9.37}$$

or

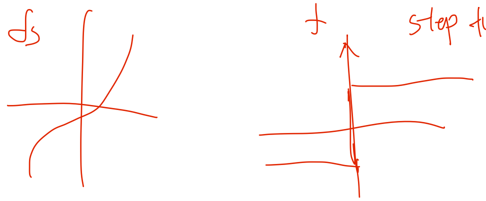
$$m\ddot{x} + b'\dot{x} + k'x = 0, \tag{9.38}$$



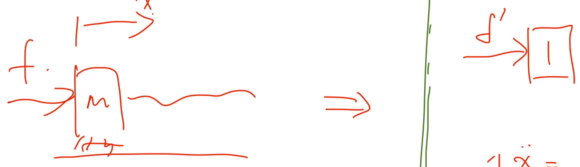
$m\ddot{x} + b\dot{x} + kx = f$

$= -k_p x - k_v \dot{x}$

choose $\to$ $b'^2 = 4mk'$



$m\ddot{x} + b\,\mathrm{sign}(\dot{x}) + kx^3 = f$



$m\ddot{x}+b\dot{x}+kx = \alpha f' + \beta \Rightarrow \begin{cases} \alpha = m \\ \beta = b\dot{x}+kx \\ f' = \ddot{x} \end{cases}$

$1\ddot{x} = f' = \ldots$

too complicated

The open-loop equation of motion for the system is

$$m\ddot{x} + b\dot{x} + kx = f. \tag{9.40}$$

The model-based portion of the control appears in a control law of the form

$$f = \alpha f' + \beta, \tag{9.41}$$

where α and β are functions or constants and are chosen so that, if f' is taken as the *new input* to the system, *the system appears to be a unit mass*. With this structure of the control law, the system equation (the result of combining (9.40) and (9.41)) is

$$m\ddot{x} + b\dot{x} + kx = \alpha f' + \beta. \tag{9.42}$$

Clearly, in order to make the system appear as a unit mass from the f' input, for this particular system we should choose α and β as follows:

$$\alpha = m,$$
$$\beta = b\dot{x} + kx. \tag{9.43}$$

Making these assignments and plugging them into (9.42), we have the system equation

$$\ddot{x} = f'. \tag{9.44}$$

Control-Law Partitioning (cont'd)

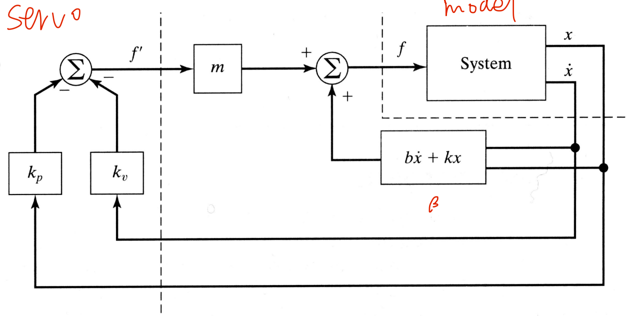
Ad:

Don't need to calculate → deal non-linear problem

$m\ddot{x} + \underbrace{\boxed{b\,\mathrm{sign}(\dot{x}) + kx^3}}_{\Downarrow\ \beta} = f$

servo

model



$f = \alpha f' + \beta$

$= m\cdot(-k_v\dot{x} - k_p x) + b\dot{x} + kx$

FIGURE 9.8: A closed-loop control system employing the partitioned control method.

This is the equation of motion for a unit mass. We now proceed as if (9.44) were the open-loop dynamics of a system to be controlled. We design a control law to compute f', just as we did before:

$$f' = -k_v\dot{x} - k_p x. \tag{9.45}$$

Combining this control law with (9.44) yields

$b^2 = 4mk.$

$k_v^2 = 4k_p$

$$\ddot{x} + k_v\dot{x} + k_p x = 0. \tag{9.46}$$

Under this methodology, the setting of the control gains is simple and is independent of the system parameters; that is,

$$k_v = 2\sqrt{k_p} \tag{9.47}$$

must hold for critical damping. Figure 9.8 shows a block diagram of the partitioned controller used to control the system of Fig. 9.6.

$s^2 + k_v s + k_p = 0.$

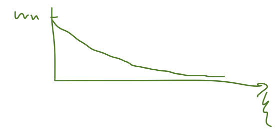
$s^2 + 2\xi w_n s + w_n^2 = 0$

$w_n = \sqrt{k_p} \le \frac{1}{2} w_{res}$

$w_n = \sqrt{\frac{k}{m}}.$

$w = \sqrt{1-\xi^2}\, w_n.$

In Exam: choose $w_n = \sqrt{k_p} = \frac{1}{2} w_{res}$



$k_v = 2\sqrt{k_p}$

$k_p \to \infty$. more stable

$w_n = \sqrt{\frac{k}{m}} = \sqrt{k_p}$

$m = 1$

$w_n = \sqrt{k_p} \le \frac{1}{2} w_{reso}$

$$f' = \ddot{x}_d + k_v \dot{e} + k_p e. \tag{9.50}$$

$1\ddot{x} = f' = \ddot{x}_d + k_v \dot{e} + k_p e$

We see that (9.50) is a good choice if we combine it with the equation of motion of a unit mass (9.44), which leads to

$$\ddot{x} = \ddot{x}_d + k_v \dot{e} + k_p e, \tag{9.51}$$

or

$$\ddot{e} + k_v \dot{e} + k_p e = 0. \tag{9.52}$$

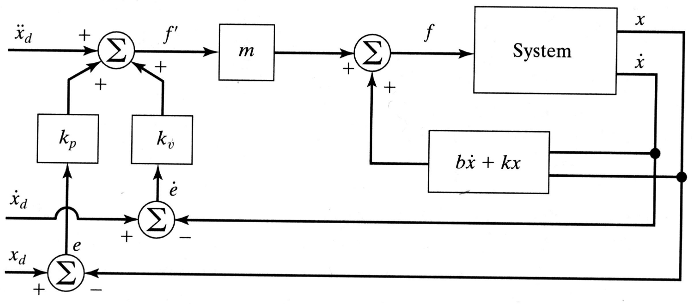
$e = x_d - x$

$\dot{e} = \dot{x}_d - \dot{x}$

$\ddot{e} = \ddot{x}_d - \ddot{x}$

This is a second-order differential equation for which we can choose the coefficients, so we can design any response we wish. (Often, critical damping is the choice made.) Such an equation is sometimes said to be written in **error space**, because it describes the evolution of errors relative to the desired trajectory. Figure 9.9 shows a block diagram of our trajectory-following controller.

If our model is perfect (i.e., our knowledge of m, b, and k), and if there is no noise and no initial error, the block will follow the desired trajectory exactly. If there is an initial error, it will be suppressed according to (9.52), and thereafter the system will follow the trajectory exactly.



when static

$f' = 1\cdot\ddot{x}$

$0 = \cancel{\ddot{e}} + \cancel{k_v\dot{e}} + k_p e$

f''

$f' + f_{dist} = f$

$f_{dist} = k_p e$

Disturbance Rejection

Steady-state error

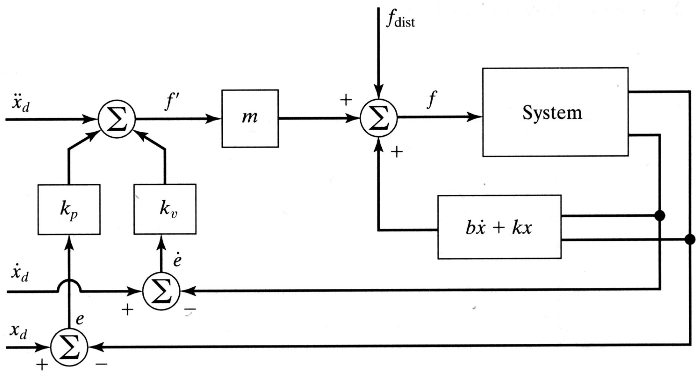
Let's consider the simplest kind of disturbance—namely, that f_{dist} is a constant. In this case, we can perform a **steady-state analysis** by analyzing the system at rest (i.e., the derivatives of all system variables are zero). Setting derivatives to zero in (9.53) yields the steady-state equation

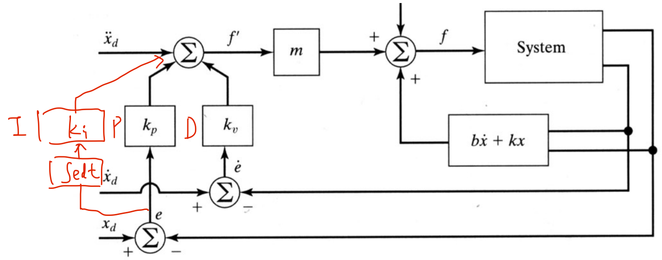
$$k_p e = f_{\text{dist}}, \tag{9.55}$$

or

$$e = f_{\text{dist}}/k_p. \tag{9.56}$$

The value of e given by (9.56) represents a **steady-state error**. Thus, it is clear that the higher the position gain k_p, the smaller will be the steady-state error.





PID-Control

Addition of an integral term

In order to eliminate steady-state error, a modified control law is sometimes used. The modification involves the addition of an *integral* term to the control law. The control law becomes

$$f' = \ddot{x}_d + k_v \ddot{e} + k_p e + k_i \int edt, \tag{9.57}$$

which results in the error equation

$$\ddot{e} + k_v \dot{e} + k_p e + k_i \int edt = f_{\text{dist}}. \tag{9.58}$$

The term is added so that the system will have no steady-state error in the presence of constant disturbances. If $e(t) = 0$ for $t < 0$, we can write (9.58) for $t > 0$ as

$$\dddot{e} + k_v \ddot{e} + k_p \dot{e} + k_i e = \dot{f}_{\text{dist}}, \; = 0 \tag{9.59}$$

which, in the steady state (for a constant disturbance), becomes

$$k_i e = 0, \tag{9.60}$$

even k_i is smal

so

enough time $k_i \int edt \to \infty$

$$e = 0. \tag{9.61}$$

Non-Linear Controller

The model-based portion of the control is $f = \alpha f' + \beta$, where now we use

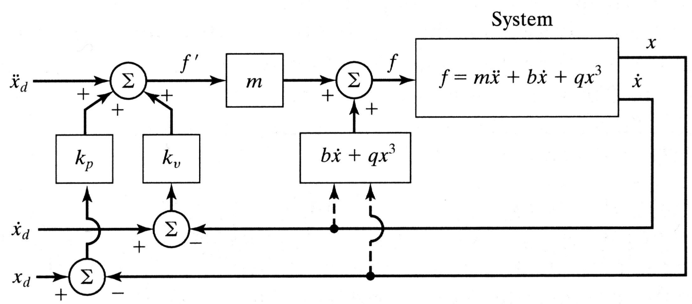
$$\alpha = m,$$

$$\beta = b\dot{x} + qx^3; \tag{10.2}$$

the servo portion is, as always

$$f' = \ddot{x}_d + k_v \dot{e} + k_p e, \tag{10.3}$$

where the values of the gains are calculated from some desired performance specification. Figure 10.2 shows a block diagram of this control system. The resulting closed-loop system maintains poles in fixed locations.



Control-Law for a Manipulator

The rigid-body dynamics have the form

$$\tau = \underbrace{M(\Theta)}_{\alpha}\ddot{\Theta} + \underbrace{V(\Theta,\dot{\Theta}) + G(\Theta)}_{\beta}, \tag{10.11}$$

where $M(\Theta)$ is the $n \times n$ inertia matrix of the manipulator, $V(\Theta,\dot{\Theta})$ is an $n \times 1$ vector of centrifugal and Coriolis terms, and $G(\Theta)$ is an $n \times 1$ vector of gravity terms. Each element of $M(\Theta)$ and $G(\Theta)$ is a complicated function that depends on Θ, the position of all the joints of the manipulator. Each element of $V(\Theta,\dot{\Theta})$ is a complicated function of both Θ and $\dot{\Theta}$.

Additionally, we could incorporate a model of friction (or other non-rigid-body effects). Assuming that our model of friction is a function of joint positions and velocities, we add the term $F(\Theta,\dot{\Theta})$ to (10.11), to yield the model

$$\tau = M(\Theta)\ddot{\Theta} + \underbrace{V(\Theta,\dot{\Theta}) + G(\Theta) + F(\Theta,\dot{\Theta})}_{\beta}. \tag{10.12}$$

The problem of controlling a complicated system like (10.12) can be handled by the partitioned controller scheme we have introduced in this chapter. In this case, we have

$$\tau = \alpha\tau' + \beta, \tag{10.13}$$

where τ is the $n \times 1$ vector of joint torques. We choose

$$\begin{aligned} \alpha &= M(\Theta), \\ \beta &= V(\Theta,\dot{\Theta}) + G(\Theta) + F(\Theta,\dot{\Theta}), \end{aligned} \tag{10.14}$$

$\tau = \alpha \tau' + \beta$
$= M(\Theta) \cdot \tau' + V(\Theta, \dot{\Theta}) + G(\Theta) + F(\Theta, \dot{\Theta})$

Model-Based Manipulator

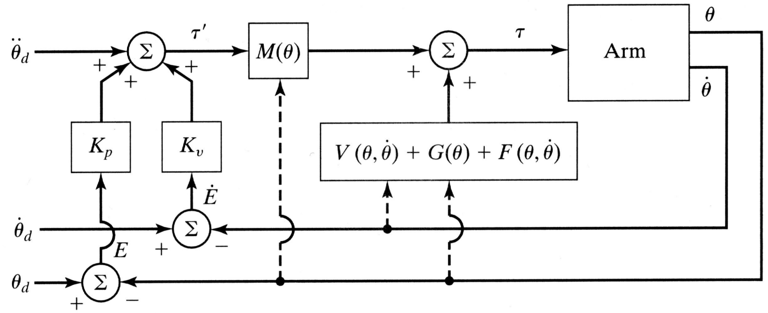


FIGURE 10.5: A model-based manipulator-control system.

with the servo law

$$\tau' = \ddot{\Theta}_d + K_v \dot{E} + K_p E, \tag{10.15}$$

where

$$E = \Theta_d - \Theta. \tag{10.16}$$

The resulting control system is shown in Fig. 10.5.

Using (10.12) through (10.15), it is quite easy to show that the closed-loop system is characterized by the error equation

$$\ddot{E} + K_v \dot{E} + K_p E = 0. \tag{10.17}$$

Note that this vector equation is decoupled: The matrices K_v and K_p are diagonal, so that (10.17) could just as well be written on a joint-by-joint basis as

$$\ddot{e}_i + k_{vi} \dot{e} + k_{pi} e = 0. \tag{10.18}$$

$$J^T \cdot \tau = F = \begin{pmatrix} F_x \\ \vdots \\ N_z \end{pmatrix} = M_x(\theta) \cdot \ddot{\theta} + V_x + G_x$$

$$\dot{x} = J \dot{\theta}$$

Only position

no v, a

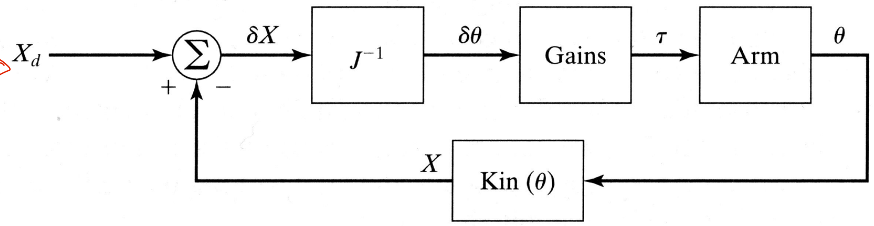
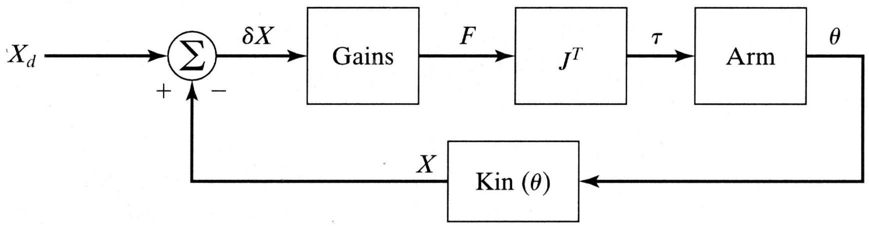


FIGURE 10.12: The inverse-Jacobian Cartesian-control scheme.



Cartesian Control using Newton-Euler Result

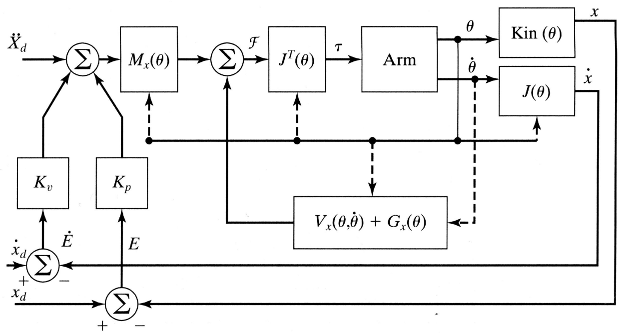
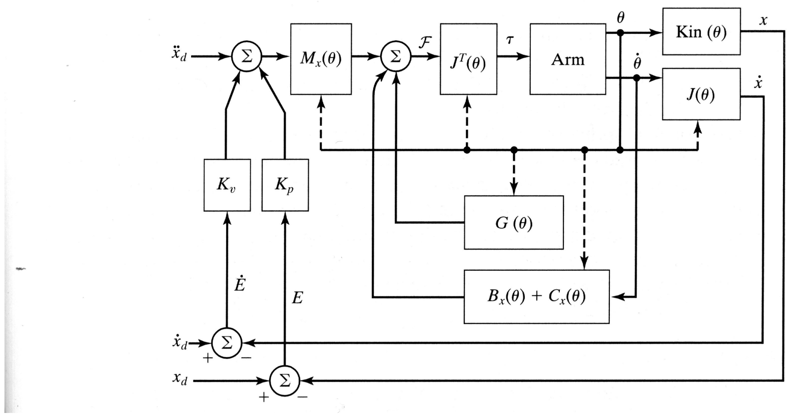
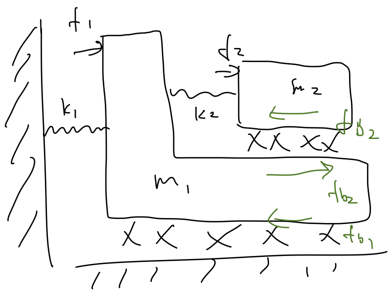


FIGURE 10.14: The Cartesian model-based control scheme.





$$f_2 = m_2 \ddot{x}_2 + b_2(\dot{x}_2 - \dot{x}_1) + k_2(x_2 - x_1)$$

$$f_1 = m_1 \ddot{x}_1 + b_1 \dot{x}_1 - b_2(\dot{x}_2 - \dot{x}_1) + k_1 x_1 - k_2(x_2 - x_1)$$

$$\begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = \underbrace{\begin{pmatrix} m_1 & 0 \\ 0 & m_2 \end{pmatrix}}_{\alpha} \begin{pmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{pmatrix} \underbrace{+ \begin{pmatrix} b_1 + b_2 & -b_2 \\ -b_2 & b_2 \end{pmatrix} \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} + \begin{pmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}}_{\beta}$$