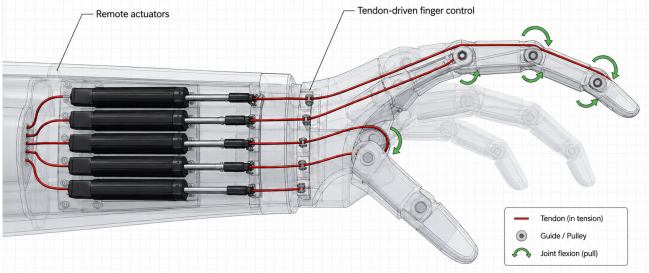
Chapter 6. Hand Dynamics and Control

Sections 6.4–6.5

Contents

Contents

Why tendon actuation?



If we put a motor directly at every revolute joint, the hand becomes bulky and heavy.

A common solution is to put the actuators away from the joints, for example in the palm or forearm, and transmit force through tendons or wires.

$$\boxed{\text{remote actuator} \quad \longrightarrow \quad \text{tendon transmission} \quad \longrightarrow \quad \text{joint motion}}$$

Central problem

Given forces applied at tendon ends, find the joint torques produced by those forces.

Figure 6.7(pp.293)

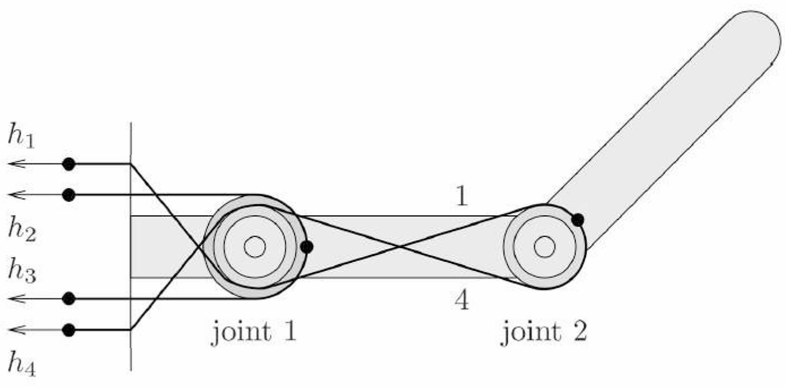


Figure 6.7(pp.293): A tendon-driven finger.

Extension function

Definition (Tendon extension function)

For tendon i, define

$$h_i : Q \to \mathbb{R}.$$

Here $h_i(\theta)$ is the path length of the i-th tendon when the joint configuration is θ, and $Q \subset \mathbb{R}^n$ is the joint configuration space.

Simple pulley case

If a tendon passes around pulleys, each joint contributes an arc-length term:

$$h_i(\theta) = l_i \pm r_{i1}\theta_1 \pm \cdots \pm r_{in}\theta_n.$$

Here l_i is the length at $\theta = 0$, and r_{ij} is the pulley radius at joint j.
The sign is positive if increasing θ_j makes the path longer, and negative if it makes the path shorter.

Figure 6.8 (pp.295)

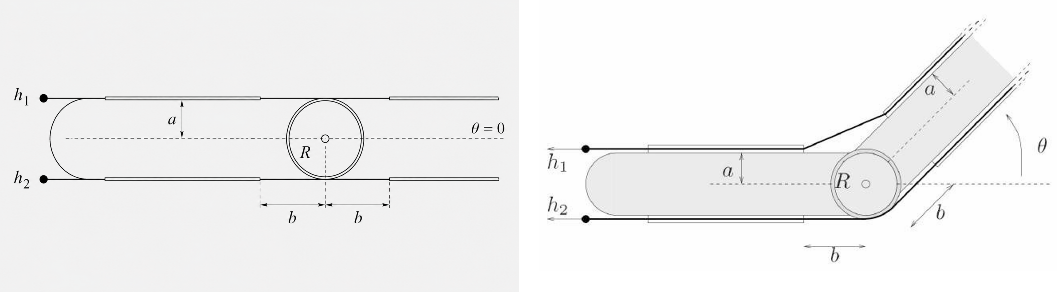


Figure 6.8: Example of tendon routing with nonlinear extension function.

Nonlinear routing example

For the joint in Figure 6.8, when $\theta > 0$,

$$h_1(\theta) = l_1 + 2\sqrt{a^2 + b^2}\cos\left(\tan^{-1}\left(\frac{a}{b}\right) + \frac{\theta}{2}\right) - 2b,$$

$$h_2(\theta) = l_2 + R\theta.$$

When $\theta < 0$, the roles of the upper and lower tendons are reversed.

Key point

Tendon length is not always linear in θ. If the routing geometry changes with the joint angle, then $h_i(\theta)$ can be a nonlinear function.

Coupling matrix

Let

$$u = h(\theta) \in \mathbb{R}^p$$

be the vector of geometric tendon extensions for p inelastic tendons.

Definition (Coupling matrix)

$$P(\theta) := \frac{\partial h^T}{\partial \theta}(\theta) \in \mathbb{R}^{n \times p}.$$

Here

$$h(\theta) = \begin{bmatrix} h_1(\theta) \\ \vdots \\ h_p(\theta) \end{bmatrix}, \qquad \frac{\partial h}{\partial \theta}(\theta) = \begin{bmatrix} \frac{\partial h_1}{\partial \theta_1} & \cdots & \frac{\partial h_1}{\partial \theta_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_p}{\partial \theta_1} & \cdots & \frac{\partial h_p}{\partial \theta_n} \end{bmatrix}.$$

Meaning

$P_{ji}(\theta)$ tells us how the i-th tendon length changes when the j-th joint angle changes.

Velocity relation

Along a motion $\theta = \theta(t)$,

$$u(t) = h(\theta(t)).$$

By the chain rule,

$$\dot{u}(t) = \frac{d}{dt}h(\theta(t)) = \frac{\partial h}{\partial \theta}(\theta(t))\dot{\theta}(t).$$

Since $P(\theta) = (\partial h/\partial \theta)^T$,

$$\boxed{\dot{u} = P(\theta)^T\dot{\theta}.}$$

In words

Joint velocity creates tendon-end velocity through the geometry of the tendon routing.

Virtual work balance

Assume there is no friction, no transmission loss, and no energy stored inside the transmission.
Then the tendon network is lossless.
For a compatible infinitesimal displacement,

$$du = P(\theta)^T d\theta.$$

The virtual work at the tendon side is

$$dW_{\text{tendon}} = f^T du.$$

The same work represented at the joint side is

$$dW_{\text{joint}} = \tau^T d\theta.$$

Because the transmission is lossless,

$$\boxed{f^T du = \tau^T d\theta.}$$

Along a motion, this becomes the power identity

$$\boxed{f^T \dot{u} = \tau^T \dot{\theta}.}$$

From tendon forces to joint torques

Use

$$\dot{u} = P(\theta)^T \dot{\theta}.$$

Then

$$f^T \dot{u} = f^T P(\theta)^T \dot{\theta} = (P(\theta) f)^T \dot{\theta}.$$

The power identity gives

$$(P(\theta) f)^T \dot{\theta} = \tau^T \dot{\theta}$$

for all allowable $\dot{\theta}$. Therefore,

$$\boxed{\tau = P(\theta) f.}$$

Meaning

Each column of $P(\theta)$ is the joint-torque vector produced by unit tension in one tendon.

Dynamics with inelastic tendon actuation

For a finger mechanism,

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + N(\theta, \dot{\theta}) = \tau.$$

Using tendon actuation, $\tau = P(\theta)f$, so

$$\boxed{M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + N(\theta, \dot{\theta}) = P(\theta)f.}$$

Simplifying assumption

We ignore actuator dynamics and tendon dynamics. Thus, when a tendon force f is commanded, it is treated as acting immediately through the coupling matrix $P(\theta)$.

Figure 6.9(pp.296)

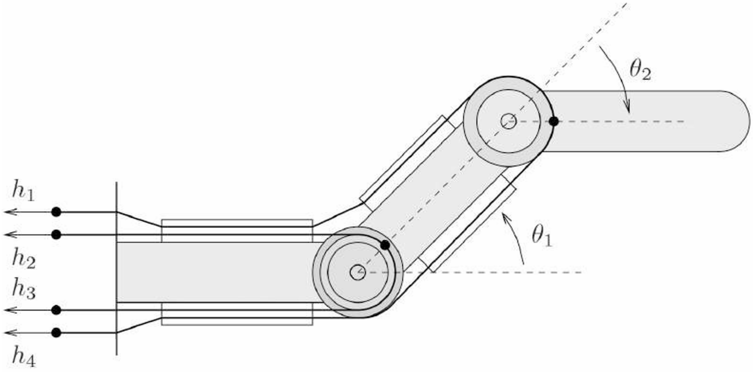


Figure 6.9(pp.296): Planar tendon-driven finger.

Example 6.4: two-link tendon-driven finger

This planar finger has two revolute joints and four tendons.
For the tendons attached to joint 1,

$$h_2 = l_2 - R_1\theta_1, \qquad h_3 = l_3 + R_1\theta_1.$$

For the outer link, when $\theta_1 > 0$,

$$h_1 = l_1 + 2\sqrt{a^2+b^2}\cos\left(\tan^{-1}\left(\frac{a}{b}\right) + \frac{\theta_1}{2}\right) - 2b - R_2\theta_2,$$

$$h_4 = l_4 + R_1\theta_1 + R_2\theta_2.$$

Example 6.4: coupling matrix

For $\theta_1 > 0$,

$$P(\theta) = \frac{\partial h^T}{\partial \theta} = \begin{bmatrix} -\sqrt{a^2+b^2}\sin\left(\tan^{-1}\left(\frac{a}{b}\right) + \frac{\theta_1}{2}\right) & -R_1 & R_1 & R_1 \\ -R_2 & 0 & 0 & R_2 \end{bmatrix}.$$

Observation

From $\tau = P(\theta)f$, tendons 2 and 3 affect only the first joint, while tendons 1 and 4 affect both joints. This coupling comes from the routing geometry.

Contents

Elastic tendons: modeling idea

The previous kinematic analysis also extends to elastic tendons.

Lumped-elasticity model

We assume that the tendons are completely free to slide along the fingers. Hence, all elasticity can be lumped into a single spring element at the base of each tendon.

Since the routing geometry is unchanged, tendon forces still produce joint torques through

$$\boxed{\tau = P(\theta) f.}$$

The difference from the inelastic case is how the tendon force is produced:

$$\boxed{\text{actuator-commanded displacement } \; e \quad \longrightarrow \quad \text{elastic stretch} \quad \longrightarrow \quad \text{tendon force } \; f.}$$

Basic quantities

For tendon i:

- e_i: actuator-commanded displacement from the zero-tension reference state.
- $h_i(\theta)$: tendon path length at the current joint configuration.
- $h_i(\theta) - h_i(0)$: tendon-length change caused by the joint motion.
- f_i: tendon tension magnitude.
- $k_i > 0$: tendon stiffness.

Reference convention

At $\theta = 0$ and $e_i = 0$, the tendon is assumed to have zero tension.

Figure 6.10(pp.297)

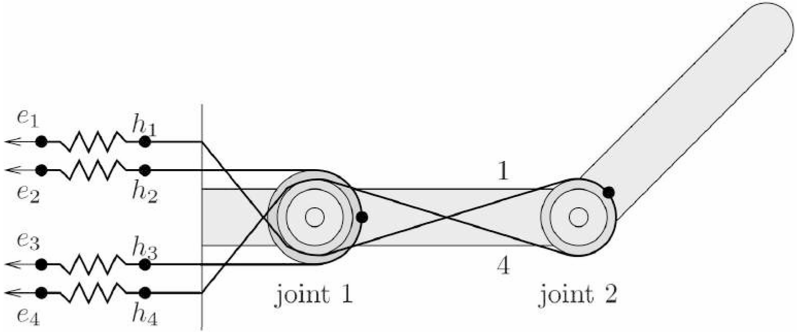


Figure 6.10(pp.297): Planar finger with position-controlled elastic tendons.

Displacement decomposition

The actuator displacement is split into two parts:

$$e_i = \underbrace{\delta_i}_{\text{spring stretch}} + \underbrace{\big(h_i(\theta) - h_i(0)\big)}_{\text{joint-induced tendon displacement}} .$$

Thus,

$$\delta_i = e_i - \big(h_i(\theta) - h_i(0)\big).$$

- e_i is the actuator-side pull.
- $h_i(\theta) - h_i(0)$ is the signed displacement caused by joint motion.
- The remaining part δ_i is the actual stretch of the tendon spring.

By Hooke's law,

$$f_i = k_i \delta_i = k_i\big(e_i + h_i(0) - h_i(\theta)\big).$$

Therefore,

$$\boxed{f = K\big(e + h(0) - h(\theta)\big)}, \qquad K = \operatorname{diag}(k_1, \ldots, k_p).$$

Elastic dynamics and stiffness term

Start from the joint-space dynamics with tendon-induced generalized torque:

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + N(\theta,\dot{\theta}) = \tau_{\text{tendon}} = P(\theta)f.$$

and substitute

$$f = K(e + h(0) - h(\theta)).$$

Then

$$M\ddot{\theta} + C\dot{\theta} + N = \underbrace{P(\theta)Ke}_{\text{actuator input torque}} + \underbrace{\left[-P(\theta)K(h(\theta) - h(0))\right]}_{\text{passive restoring torque}}.$$

Equivalently,

$$\boxed{M\ddot{\theta} + C\dot{\theta} + N + \underbrace{P(\theta)K(h(\theta) - h(0))}_{\text{passive stiffness term}} = \underbrace{P(\theta)Ke}_{\text{actuator input torque}}.}$$

The blue term is part of the passive elastic dynamics, not the commanded actuator input.

Stiffness and input coupling

Define

$$S(\theta) := P(\theta)K(h(\theta) - h(0)), \qquad Q(\theta) := P(\theta)K.$$

Then

$$\boxed{M\ddot{\theta} + C\dot{\theta} + N + S(\theta) = Q(\theta)e.}$$

Meaning

$S(\theta)$ is the passive stiffness produced by the elastic tendon network. $Q(\theta)$ maps actuator displacement commands into joint torque input.

Figure 6.10(pp.297)

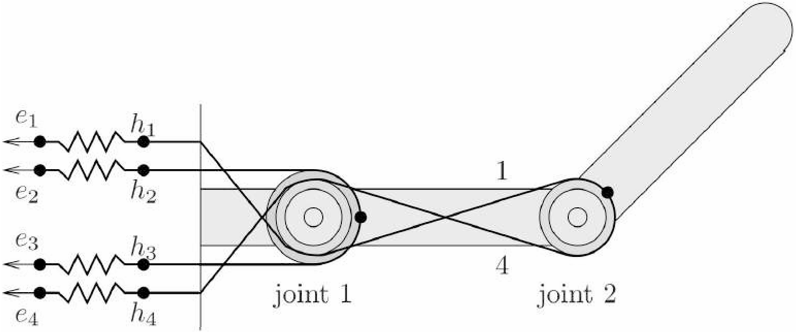


Figure 6.10(pp.297): Planar finger with position-controlled elastic tendons.

Example 6.5: elastic tendon coupling

For the Figure 6.10 network,

$$\begin{aligned} h_1 &= l_1 + r_{11}\theta_1 - r_{12}\theta_2, & h_2 &= l_2 - r_{21}\theta_1, \\ h_3 &= l_3 + r_{31}\theta_1, & h_4 &= l_4 - r_{41}\theta_1 + r_{42}\theta_2. \end{aligned}$$

Hence

$$P(\theta) = \frac{\partial h^T}{\partial \theta} = \begin{bmatrix} r_{11} & -r_{21} & r_{31} & -r_{41} \\ -r_{12} & 0 & 0 & r_{42} \end{bmatrix}.$$

Because all h_i are linear functions of θ, $P(\theta)$ is constant in this example.

Example 6.5: stiffness and input coupling

With

$$K = \begin{bmatrix} k_1 & 0 & 0 & 0 \\ 0 & k_2 & 0 & 0 \\ 0 & 0 & k_3 & 0 \\ 0 & 0 & 0 & k_4 \end{bmatrix}, \quad h(\theta) - h(0) = \begin{bmatrix} r_{11}\theta_1 - r_{12}\theta_2 \\ -r_{21}\theta_1 \\ r_{31}\theta_1 \\ -r_{41}\theta_1 + r_{42}\theta_2 \end{bmatrix} = P^T\theta,$$

the stiffness term is

$$S(\theta) = P(\theta)K(h(\theta) - h(0)) = \begin{bmatrix} k_1 r_{11}^2 + k_2 r_{21}^2 + k_3 r_{31}^2 + k_4 r_{41}^2 & -k_1 r_{11} r_{12} - k_4 r_{41} r_{42} \\ -k_1 r_{11} r_{12} - k_4 r_{41} r_{42} & k_1 r_{12}^2 + k_4 r_{42}^2 \end{bmatrix} \theta.$$

The actuator-to-torque coupling is

$$Q(\theta) = P(\theta)K = \begin{bmatrix} k_1 r_{11} & -k_2 r_{21} & k_3 r_{31} & -k_4 r_{41} \\ -k_1 r_{12} & 0 & 0 & k_4 r_{42} \end{bmatrix}.$$

Contents

Positive span of available torques

Tendons can pull, but they cannot push. Therefore,

$$f_i > 0 \quad \text{for all } i.$$

So the set of achievable torques is

$$\mathcal{T}(\theta) = \{P(\theta)f : f \in \mathbb{R}^p,\ f_i > 0\}.$$

This is the positive span of the columns of $P(\theta)$.

Force-closure tendon network

Definition (Force-closure)

A tendon network is force-closure at θ if every desired joint torque can be produced with positive tendon tensions. That is, for every $\tau \in \mathbb{R}^n$, there is $f \in \mathbb{R}^p$ such that

$$P(\theta)f = \tau, \quad f > 0 \ \ (:= f_i > 0 \quad \forall i = 1, \dots, p.) \tag{6.34}$$

Equivalently,

$$\mathrm{pos}\{P_1(\theta), \dots, P_p(\theta)\} = \mathbb{R}^n,$$

where $P_i(\theta)$ is the i-th column of $P(\theta)$.

Force-closure criterion

Proposition

A tendon network is force-closure if and only if

1. $P(\theta)$ *is surjective, and*
2. *there exists* $f_N > 0$ *such that* $P(\theta)f_N = 0$.

Proof.

If the network is force-closure, then every torque is possible, so $P(\theta)$ is surjective. In particular, the zero torque can be produced by some positive tension vector $f_N > 0$, so $P(\theta)f_N = 0$.
Conversely, take any desired τ. Since $P(\theta)$ is surjective, choose f_0 with $P(\theta)f_0 = \tau$. Since $f_N > 0$, choose $\alpha > 0$ large enough so that $f_0 + \alpha f_N > 0$. Then

$$P(\theta)(f_0 + \alpha f_N) = \tau.$$

□

How to choose tendon forces

Suppose we want to produce a desired joint torque τ. Then we need to solve

$$P(\theta)f = \tau.$$

If $P(\theta) : \mathbb{R}^p \to \mathbb{R}^n$ is surjective, one particular solution is

$$f = P^+(\theta)\tau,$$

where

$$P^+(\theta) = P(\theta)^T \left(P(\theta)P(\theta)^T\right)^{-1} \in \mathbb{R}^{p\times n} \quad \text{and} \quad P(\theta)P^+(\theta) = I_n.$$

But this solution may not have all positive components. So we add an internal tension vector $f_N \in \mathcal{N}(P(\theta))$:

$$\boxed{f = P^+(\theta)\tau + f_N.}$$

Since

$$P(\theta)f_N = 0,$$

the added term does not change the torque.

Position-controlled elastic tendon actuation

For elastic tendons, we solve

$$P(\theta)Ke = \tau, \qquad e_i + h_i(0) - h_i(\theta) > 0. \tag{6.35}$$

The inequality means that each tendon remains in positive tension.
Since K is invertible, force-closure of the tendon network implies that there exists a positive null-space extension $e_N > 0$ such that

$$P(\theta)Ke_N = 0.$$

We can choose the actuator command as

$$\boxed{e = (P(\theta)K)^{+}\tau + e_N.}$$

Contents

Control of robot hands: basic setting

We consider a constrained multifingered hand system.

- $\theta \in \mathbb{R}^n$: finger joint variables.
- $x \in \mathbb{R}^p$: object or workspace variables.
- $q = (\theta, x)$: full configuration.

The control problem has two parts:

1. Track a desired object/workspace trajectory $x_d(t)$.
2. Maintain a desired internal force for stable contact.

Meaning of J and G

J is the hand Jacobian, and G is the grasp map. Together, GJ^{-T} maps joint torques to the resulting workspace force.

Reduced workspace dynamics

In the reduced workspace coordinates, the constrained dynamics are written as

$$\tilde{M}(x)\ddot{x} + \tilde{C}(x,\dot{x})\dot{x} + \tilde{N}(x,\dot{x}) = F = GJ^{-T}\tau.$$

Here F is the generalized force acting on the object/workspace coordinate x.

Control strategy

First design F as if it were a direct input for tracking $x_d(t)$. Then choose joint torques τ satisfying

$$GJ^{-T}\tau = F.$$

Equivalently, the actual input is τ, but the controller is first designed in the workspace through the virtual input F.

$$\boxed{\text{joint torque} \quad \tau \xrightarrow{GJ^{-T}} \text{workspace force} \quad F \Longrightarrow \text{motion of} \quad x}$$

Workspace tracking by computed torque

Let

$$e_x := x - x_d.$$

Treat F as a virtual input and choose

$$\boxed{F = \tilde{M}(x)(\ddot{x}_d - K_v \dot{e}_x - K_p e_x) + \tilde{C}(x, \dot{x})\dot{x} + \tilde{N}(x, \dot{x}).}$$

Substituting this into

$$\tilde{M}\ddot{x} + \tilde{C}\dot{x} + \tilde{N} = F$$

gives

$$\ddot{x} = \ddot{x}_d - K_v \dot{e}_x - K_p e_x.$$

Hence

$$\boxed{\ddot{e}_x + K_v \dot{e}_x + K_p e_x = 0.}$$

Meaning

With suitable positive gains, the workspace tracking error converges to zero.

From workspace force to joint torque

After computing the desired workspace force F, choose τ such that

$$GJ^{-T}\tau = F.$$

Let

$$B(q) := GJ^{-T} \in \mathbb{R}^{p\times n}.$$

Assume

$$n \geq p, \qquad \operatorname{rank} B(q) = p.$$

Then $B(q)$ is surjective, so at least one solution exists.
A general choice is

$$\boxed{\tau = J^T(G^+F + f_N),} \tag{6.36}$$

where

$$G^+ = G^T(GG^T)^{-1}, \qquad f_N \in \mathcal{N}(G).$$

F_n is called the contact internal force because it does not affect the workspace motion.

Role of internal forces

The general torque realizing the workspace force F is

$$\boxed{\tau = J^T G^+ F + J^T f_N, \qquad f_N \in \mathcal{N}(G).}$$

The first term controls the object motion:

$$G J^{-T} (J^T G^+ F) = F.$$

The second term does not affect the object motion:

$$G J^{-T} (J^T f_N) = G f_N = 0.$$

Interpretation

f_N is used to regulate contact forces, for example to keep the grasp stable.

Contents

Why use a hierarchy?

A multifingered hand is difficult to control as one huge system.

- It may have many actuators.
- It may have many contact and kinematic constraints.
- The controller must run fast in real time.

Main difficulty

If we model the whole hand as one large dynamical system, then computing the control input can become too expensive for real-time control.

Two basic operations

The main idea is to control a complex robot hand by breaking it into simpler parts.

1. Attach: combine robots by constraints

If several robots are mechanically connected, we treat them as one composite robot.

$$\text{fingers} + \text{object} \quad \longrightarrow \quad \text{hand-object system}$$

Example: when fingers grasp an object, contact constraints link the finger motion and the object motion.

2. Control: add a controller to a robot

A controller changes how a robot behaves, but the result is still treated as a robot object.

$$\text{robot} \quad \longrightarrow \quad \text{controlled robot}$$

Conceptual hierarchy

The control structure can be built recursively:

finger robot	$\longrightarrow$	controlled finger		
	$\longrightarrow$	attached hand-object system	$\longrightarrow$	controlled hand-object system

- Low level: control each finger locally, e.g., by PD feedback.
- Middle level: attach several fingers and the object through contact constraints.
- High level: control the task variable, such as the object trajectory $x_d(t)$.

Reference

Richard M. Murray, Zexiang Li, and S. Shankar Sastry.
A Mathematical Introduction to Robotic Manipulation.
CRC Press, 1994.
ISBN: 0849379814, 9780849379819.