Chapter 9
Applications of Unconstrained Optimization

9.1 Introduction

Optimization problems occur in many disciplines, for example, in engineering, physical sciences, social sciences, and commerce. In this chapter, we demonstrate the usefulness of the unconstrained optimization algorithms studied in this book by applying them to a number of problems in engineering. Applications of various constrained optimization algorithms will be presented in Chap. 16.

Optimization is particularly useful in the various branches of engineering like electrical, mechanical, chemical, and aeronautical engineering. The applications we consider here and in Chap. 16 are in the areas of digital signal processing, pattern recognition, automatic control, robotics, and telecommunications. For each selected application, sufficient background material is provided to assist the reader to understand the application. The steps involved are the problem formulation phase which converts the problem at hand into an unconstrained optimization problem, and the solution phase which involves selecting and applying an appropriate optimization algorithm.

In Sec. 9.2, we examine a problem of point-pattern matching in an unconstrained optimization framework. To this end, the concept of similarity transformation is introduced to quantify the meaning of 'best pattern matching'. In addition, it is shown that the optimal pattern from a database that best matches a given point pattern can be obtained by minimizing a convex quadratic function. In Sec. 9.3, we consider a problem known as the inverse kinematics of robotic manipulators which entails a system of nonlinear equations. The problem is first converted into an unconstrained minimization problem and then various methods studied earlier are applied and the results obtained are compared in terms of solution accuracy and computational efficiency. Throughout the discussion, the advantages of using an optimization-based solution method relative to a conventional closed-form method are stressed. In Sec. 9.4, we obtain weighted least-squares and minimax designs of finite-duration impulse-response (FIR) digital filters using unconstrained optimization.

9.2 Point-Pattern Matching

9.2.1 Motivation

A problem that arises in pattern recognition is the so-called *point-pattern matching problem*. In this problem, a pattern such as a printed or handwritten character, numeral, symbol, or even the outline of a manufactured part can be described by a set of points, say,

$$\mathcal{P} = \{\mathbf{p}_1,\ \mathbf{p}_2,\ \ldots,\ \mathbf{p}_n\} \tag{9.1}$$

where

$$\mathbf{p}_i = \begin{bmatrix} p_{i1} \\ p_{i2} \end{bmatrix}$$

is a vector in terms of the coordinates of the ith sample point. If the number of points in $\mathcal{P}$, n, is sufficiently large, then $\mathcal{P}$ in Eq. (9.19.1) describes the object accurately and $\mathcal{P}$ is referred to as a *point pattern* of the object. The same object viewed from a different distance and/or a different angle will obviously correspond to a different point pattern, $\tilde{\mathcal{P}}$, and it is of interest to examine whether or not two given patterns are matched to within a scaled rotation and a translation.

In a more general setting, we consider the following pattern-matching problem: We have a database that contains N standard point patterns $\{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_N\}$ where each $\mathcal{P}_i$ has the form of Eq. (9.19.1) and we need to find a pattern from the database that best matches a given point pattern $\mathcal{Q} = \{\mathbf{q}_1,\ \mathbf{q}_2,\ \ldots,\ \mathbf{q}_n\}$. In order to solve this problem, two issues need to be addressed. First, we need to establish a measure to quantify the meaning of 'best matching'. Second, we need to develop a solution method to find an optimal pattern $\mathcal{P}^*$ from the database that best matches pattern $\mathcal{Q}$ based on the chosen measure.

9.2.2 Similarity transformation

Two point patterns $\mathcal{P}$ and $\tilde{\mathcal{P}}$ are said to be *similar* if one pattern can be obtained by applying a scaled rotation plus a translation to the other. If pattern $\mathcal{P}$ is given by Eq. (9.19.1) and

$$\tilde{\mathcal{P}} = \{\tilde{\mathbf{p}}_1,\ \tilde{\mathbf{p}}_2,\ \ldots,\ \tilde{\mathbf{p}}_n\} \quad \text{with} \qquad \tilde{\mathbf{p}}_i = [\tilde{p}_{i1}\ \tilde{p}_{i2}]^T$$

then $\mathcal{P}$ and $\tilde{\mathcal{P}}$ are similar if and only if there exist a rotation angle θ, a scaling factor η, and a translation vector $\mathbf{r} = [r_1\ r_2]^T$ such that the relation

$$\tilde{\mathbf{p}}_i = \eta \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \mathbf{p}_i + \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \tag{9.2}$$

holds for $i = 1, 2, \ldots, n$. A transformation that maps pattern $\mathcal{P}$ to pattern $\mathcal{Q}$ is said to be a *similarity transformation*. From Eq. (()9.29.2), we see that a similarity transformation is characterized by the parameter column vector $[\eta\ \theta\ r_1\ r_2]^T$. Note that the similarity transformation is a nonlinear function of parameters η and θ. This nonlinearity can lead to a considerable increase in the amount of computation required by the optimization process. This problem can be fixed by applying the variable substitution

$$a = \eta\cos\theta, \quad b = \eta\sin\theta$$

to Eq. (9.29.2) to obtain

$$\tilde{\mathbf{p}}_i = \begin{bmatrix} a & -b \\ b & a \end{bmatrix}\mathbf{p}_i + \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \tag{9.3}$$

Thus the parameter vector becomes $\mathbf{x} = [a\ b\ r_1\ r_2]^T$. Evidently, the similarity transformation now depends *linearly* on the parameters.

9.2.3 Problem formulation

In a real-life problem, a perfect match between a given point pattern $\mathcal{Q}$ and a point pattern in the database is unlikely, and the best we can do is identify the closest pattern to $\mathcal{Q}$ to within a similarity transformation.

Let

$$\mathcal{Q} = \{\mathbf{q}_1,\ \mathbf{q}_2,\ \ldots,\ \mathbf{q}_n\}$$

be a given pattern and assume that

$$\tilde{\mathcal{P}}(\mathbf{x}) = \{\tilde{\mathbf{p}}_1,\ \tilde{\mathbf{p}}_2,\ \ldots,\ \tilde{\mathbf{p}}_n\}$$

is a transformed version of pattern

$$\mathcal{P} = \{\mathbf{p}_1,\ \mathbf{p}_2,\ \ldots,\ \mathbf{p}_n\}$$

Let these patterns be represented by the matrices

$$\mathbf{Q} = [\mathbf{q}_1\ \mathbf{q}_2\ \cdots\ \mathbf{q}_n],\ \tilde{\mathbf{P}}(\mathbf{x}) = [\tilde{\mathbf{p}}_1\ \tilde{\mathbf{p}}_2\ \cdots\ \tilde{\mathbf{p}}_n],\ \text{and}\ \mathbf{P} = [\mathbf{p}_1\ \mathbf{p}_2\ \cdots\ \mathbf{p}_n]$$

respectively. A transformed pattern $\tilde{\mathcal{P}}$ that matches $\mathcal{Q}$ can be obtained by solving the unconstrained optimization problem

$$\underset{\mathbf{x}}{\text{minimize}} \quad \|\tilde{\mathbf{P}}(\mathbf{x}) - \mathbf{Q}\|_F^2 \tag{9.4}$$

where $\|\cdot\|_F$ denotes the Frobenius norm (see Sec. A.8.2). The solution of the above minimization problem corresponds to finding the best transformation that would minimize the difference between patterns $\tilde{\mathcal{P}}$ and $\mathcal{Q}$ in the Frobenius sense. Since

$$\|\tilde{\mathbf{P}}(\mathbf{x}) - \mathbf{Q}\|_F^2 = \sum_{i=1}^{n} \|\tilde{\mathbf{p}}_i(x) - \mathbf{q}_i\|^2$$

the best transformation in the least-squares sense is obtained.

Now if $\mathbf{x}^*$ is the minimizer of the problem in Eq. (9.49.4), then the error

$$e(\tilde{\mathcal{P}}, Q) = \|\tilde{\mathbf{P}}(\mathbf{x}^*) - \mathbf{Q}\|_F \tag{9.5}$$

is a measure of the dissimilarity between patterns $\tilde{\mathcal{P}}$ and Q. Obviously, $e(\tilde{\mathcal{P}}, Q)$ should be as small as possible and a zero value would correspond to a perfect match.

9.2.4 Solution of the problem in Eq. (9.49.4)

On using Eq. (9.39.3), Eq. (9.59.5) gives

$$\begin{aligned}
\left\|\tilde{\mathbf{P}}(\mathbf{x}) - \mathbf{Q}\right\|_F^2 &= \sum_{i=1}^{n} \left\|\tilde{\mathbf{p}}_i(x) - \mathbf{q}_i\right\|^2 \\
&= \sum_{i=1}^{n} \left\| \begin{bmatrix} ap_{i1} - bp_{i2} + r_1 \\ bp_{i1} + ap_{i2} + r_2 \end{bmatrix} - \mathbf{q}_i \right\|^2 \\
&= \sum_{i=1}^{n} \left\| \begin{bmatrix} p_{i1} & -p_{i2} & 1 & 0 \\ p_{i2} & p_{i1} & 0 & 1 \end{bmatrix} \mathbf{x} - \mathbf{q}_i \right\|^2 \\
&= \mathbf{x}^T \mathbf{H} \mathbf{x} - 2\mathbf{x}^T \mathbf{b} + \kappa
\end{aligned} \tag{9.6a}$$

where

$$\mathbf{H} = \begin{bmatrix} \sum_{i=1}^{n} \mathbf{R}_i^T \mathbf{R}_i & \sum_{i=1}^{n} \mathbf{R}_i^T \\ \sum_{i=1}^{n} \mathbf{R}_i & n\mathbf{I}_2 \end{bmatrix}, \quad \mathbf{R}_i = \begin{bmatrix} p_{i1} & -p_{i2} \\ p_{i2} & p_{i1} \end{bmatrix} \tag{9.6b}$$

$$\mathbf{b} = \sum_{i=1}^{n} [\mathbf{R}_i \; \mathbf{I}_2]^T \mathbf{q}_i \tag{9.6c}$$

$$\kappa = \sum_{i=1}^{n} \left\|\mathbf{q}_i\right\|^2 \tag{9.6d}$$

(see Prob. 9.1(*a*)). It can be readily verified that the Hessian $\mathbf{H}$ in Eq. (9.6b9.6b) is positive definite (see Prob. 9.1(*b*)) and hence it follows from Chap. 2 that the objective function in Eq. (9.49.4) is globally strictly convex and, therefore, has a unique global minimizer. Using Eq. (9.6a9.6a), the gradient of the objective function can be obtained as

$$\mathbf{g}(\mathbf{x}) = 2\mathbf{H}\mathbf{x} - 2\mathbf{b}$$

The unique global minimizer can be obtained in closed form by letting

$$\mathbf{g}(\mathbf{x}) = 2\mathbf{H}\mathbf{x} - 2\mathbf{b} = \mathbf{0}$$

and hence

$$\mathbf{x}^* = \mathbf{H}^{-1}\mathbf{b} \tag{9.7}$$

Since $\mathbf{H}$ is a positive definite matrix of size 4×4, its inverse exists and is easy to evaluate (see Prob. 9.1(c)).

9.2.5 Alternative measure of dissimilarity

As can be seen in Eq. (9.6a9.6a), the Frobenius norm of a matrix can be related to the L_2 norm of its column vectors. If we define two new vectors $\tilde{\mathbf{p}}(\mathbf{x})$ and $\mathbf{q}$ as

$$\tilde{\mathbf{p}}(\mathbf{x}) = \begin{bmatrix} \tilde{\mathbf{p}}_1(\mathbf{x}) \\ \tilde{\mathbf{p}}_2(\mathbf{x}) \\ \vdots \\ \tilde{\mathbf{p}}_n(\mathbf{x}) \end{bmatrix} \quad \text{and} \quad \mathbf{q} = \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \vdots \\ \mathbf{q}_n \end{bmatrix}$$

then Eq. (9.6) Eqs. (9.6a)–(9.6d) implies that

$$\|\tilde{\mathbf{P}}(\mathbf{x}) - \mathbf{Q}\|_F^2 = \|\tilde{\mathbf{p}}(\mathbf{x}) - \mathbf{q}\|^2$$

Hence the dissimilarity measure defined in Eq. (9.59.5) can be expressed as

$$e(\tilde{\mathcal{P}}, Q) = \|\tilde{\mathbf{p}}(\mathbf{x}) - \mathbf{q}\|$$

An alternative of the above dissimilarity measure can be defined in terms of the L_{2p} norm

$$e_{2p}(\tilde{\mathcal{P}}, Q) = \|\tilde{\mathbf{p}}(\mathbf{x}) - \mathbf{q}\|_{2p}$$

As p increases, $e_{2p}(\tilde{\mathcal{P}}, Q)$ approaches the L_∞ norm of $\tilde{\mathbf{p}}(\mathbf{x}) - \mathbf{q}$ which is numerically equal to the maximum of the function. Therefore, solving the problem

$$\underset{\mathbf{x}}{\text{minimize}} \quad e_{2p}(\tilde{\mathcal{P}}, Q) = \|\tilde{\mathbf{p}}(\mathbf{x}) - \mathbf{q}\|_{2p} \tag{9.8}$$

with a sufficiently large p amounts to minimizing the maximum error between symbols $\tilde{\mathcal{P}}$ and Q. If we let

$$\mathbf{r}_{i1} = [p_{i1} \quad -p_{i2} \quad 1 \quad 0]^T$$
$$\mathbf{r}_{i2} = [p_{i2} \quad p_{i1} \quad 0 \quad 1]^T$$
$$\mathbf{q}_i = \begin{bmatrix} q_{i1} \\ q_{i2} \end{bmatrix}$$

then the objective function in Eq. (9.89.8) can be expressed as

$$e_{2p}(\mathbf{x}) = \left\{ \sum_{i=1}^{n} [(\mathbf{r}_{i1}^T \mathbf{x} - q_{i1})^{2p} + (\mathbf{r}_{i2}^T \mathbf{x} - q_{i2})^{2p}] \right\}^{1/2p} \tag{9.9a}$$

The gradient and Hessian of $e_{2p}(\mathbf{x})$ can be evaluated as

$$\nabla e_{2p}(\mathbf{x}) = \frac{1}{e_{2p}^{2p-1}(\mathbf{x})} \sum_{i=1}^{n} [(\mathbf{r}_{i1}^T \mathbf{x} - q_{i1})^{2p-1} + (\mathbf{r}_{i2}^T \mathbf{x} - q_{i2})^{2p-1}] \tag{9.9b}$$

and

$$\nabla^2 e_{2p}(\mathbf{x}) = \frac{(2p-1)}{e_{2p}^{2p-1}(\mathbf{x})} \sum_{i=1}^{n} [(\mathbf{r}_{i1}^T \mathbf{x} - q_{i1})^{2p-2} \mathbf{r}_{i1} \mathbf{r}_{i1}^T + (\mathbf{r}_{i2}^T \mathbf{x} - q_{i2})^{2p-2} \mathbf{r}_{i2} \mathbf{r}_{i2}^T]$$
$$- \frac{(2p-1)}{e_{2p}(\mathbf{x})} \nabla e_{2p}(\mathbf{x}) \nabla^T e_{2p}(\mathbf{x}) \tag{9.9c}$$

respectively (see Prob. 9.3(*a*)). It can be shown that the Hessian $\nabla^2 e_{2p}(\mathbf{x})$ in Eq. (9.9c9.9c) is positive semidefinite for any $\mathbf{x} \in R^4$ and, therefore, the objective function $e_{2p}(\mathbf{x})$ is globally convex (see Prob. 9.3(*b*)).

Since the Hessian of $e_{2p}(\mathbf{x})$ is a 4×4 positive semidefinite matrix and is available in closed form, the Newton algorithm (Algorithm 5.3) with the Hessian matrix $\mathbf{H}_k$ modified according to Eq. (5.13**??**) is an appropriate algorithm for the solution of the problem in Eq. (9.89.8). If the power $2p$ involved in the optimization problem is a power of 2, i.e., $2p = 2^K$, then the problem at hand can be solved by first solving the problem for the case $p = 1$ using Eq. (9.79.7). The minimizer so obtained can then be used as the initial point to minimize the objective function for $p = 2$. This procedure is then repeated for $p = 4, 8, 16, \ldots$ until two successive optimizations give the same maximum error to within a prescribed tolerance.

9.2.6 Handwritten character recognition

For illustration purposes, we consider the problem of recognizing a handwritten character using a database comprising the ten 'standard' characters shown in Fig. 9.19.1. Each character in the database can be represented by a point pattern of the form in Eq. (9.19.1) with $n = 196$, and the patterns for *a*, *c*, *e*, … can be denoted as $\mathcal{P}_a$, $\mathcal{P}_c$, $\mathcal{P}_e$, … where the subscript represents the associated character. Fig. 9.29.2 shows a set of sample points that form pattern $\mathcal{P}_a$ in the database. The character to

be recognized is plotted in Fig. 9.39.3. It looks like a rotated *e*, it is of larger size relative to the corresponding character in the database, and it is largely located in the third quadrant. To apply the method discussed, the character in Fig. 9.39.3 is represented by a point pattern $\mathcal{Q}$ with $n = 196$.

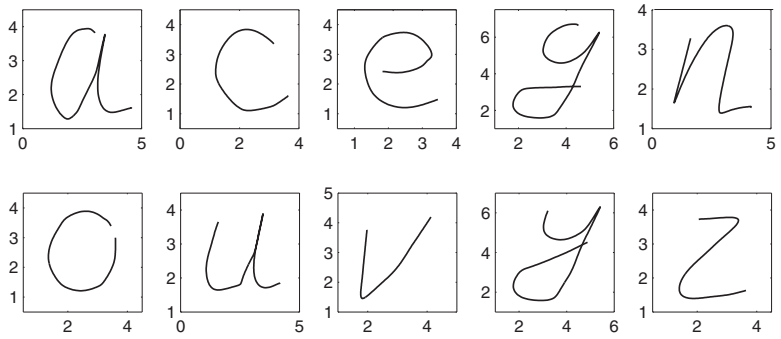

Fig. 9.1: Ten standard characters in the database.

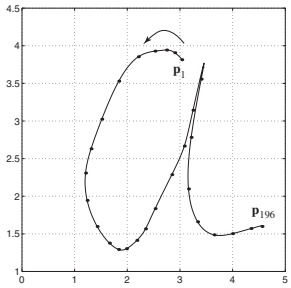

Fig. 9.2: figure
Sample points in pattern $\mathcal{P}_a$.

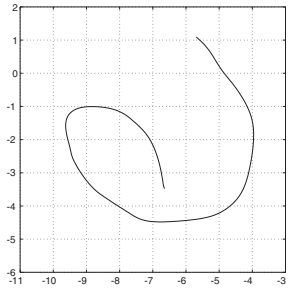

Fig. 9.3: figure
A character to be recognized.

The dissimilarity between each pattern $\mathcal{P}_{\text{character}}$ in the database and pattern $\mathcal{Q}$ is measured in terms of $e(\mathcal{P}_{\text{character}}, \mathcal{Q})$ in Eq. (9.59.5) and $e_{2p}(\mathcal{P}_{\text{character}}, \mathcal{Q})$ in Eq. (9.89.8) with $2p = 128$. Note that the minimization of $e(\mathcal{P}_{\text{character}}, \mathcal{Q})$ can be viewed as a special case of the problem in Eq. (9.89.8) with $p = 1$, and its solution can be obtained using Eq. (9.79.7). For the minimization of $e_{128}(\mathcal{P}_{\text{character}}, \mathcal{Q})$, a sequential implementation of the Newton method as described in Sec. 9.2.5 was used to obtain the solution. The results obtained are summarized in Table 9.1 where $\mathbf{x}_2^*$ and $\mathbf{x}_{128}^*$ denote the minimizers of $e_2(\mathcal{P}_{\text{character}}, \mathcal{Q})$ and $e_{128}(\mathcal{P}_{\text{character}}, \mathcal{Q})$, respectively. From the table, it is evident that the character in Fig. 9.39.3 is most similar to character *e*.

See [1] [**?**] for an in-depth investigation of dissimilarity and affine invariant distances between two-dimensional point patterns.

9.3 Inverse Kinematics for Robotic Manipulators

9.3.1 Position and orientation of a manipulator

Typically an industrial robot, also known as a robotic manipulator, comprises a chain of mechanical links with one end fixed relative to the ground and the other end, known as the *end-effector*, free to move. Motion is made possible in a manipulator by moving the joint of each link about its axis with an electric or hydraulic actuator.

Table 9.1 Comparison of dissimilarity measures

Character	$\mathbf{x}_2^*$	$e(\mathcal{P},\mathcal{Q})$	$\mathbf{x}_{128}^*$	$e_{128}(\mathcal{P},\mathcal{Q})$
a	$\begin{bmatrix} 0.8606 \\ 0.0401 \\ -8.9877 \\ -4.4466 \end{bmatrix}$	30.7391	$\begin{bmatrix} 0.4453 \\ 0.3764 \\ -6.8812 \\ -4.0345 \end{bmatrix}$	2.7287
c	$\begin{bmatrix} 0.8113 \\ 1.3432 \\ -5.5632 \\ -7.0455 \end{bmatrix}$	19.9092	$\begin{bmatrix} -0.0773 \\ 1.0372 \\ -4.4867 \\ -4.4968 \end{bmatrix}$	2.0072
e	$\begin{bmatrix} -1.1334 \\ 1.9610 \\ 0.6778 \\ -3.9186 \end{bmatrix}$	5.2524	$\begin{bmatrix} -1.0895 \\ 2.0307 \\ 0.6513 \\ -4.1631 \end{bmatrix}$	0.4541
g	$\begin{bmatrix} -0.2723 \\ 0.5526 \\ -3.5780 \\ -3.1246 \end{bmatrix}$	30.4058	$\begin{bmatrix} -0.0481 \\ 0.8923 \\ -2.7970 \\ -5.3467 \end{bmatrix}$	2.5690
n	$\begin{bmatrix} 0.0670 \\ 0.5845 \\ -5.6081 \\ -3.8721 \end{bmatrix}$	33.0044	$\begin{bmatrix} -0.0745 \\ 0.6606 \\ -5.2831 \\ -3.9995 \end{bmatrix}$	2.5260
o	$\begin{bmatrix} 1.0718 \\ 1.3542 \\ -6.0667 \\ -8.3572 \end{bmatrix}$	16.8900	$\begin{bmatrix} -0.2202 \\ 1.2786 \\ -3.1545 \\ -4.9915 \end{bmatrix}$	2.1602
u	$\begin{bmatrix} 0.3425 \\ 0.3289 \\ -8.5193 \\ -2.2115 \end{bmatrix}$	33.6184	$\begin{bmatrix} 0.0600 \\ 0.0410 \\ -6.8523 \\ -2.0225 \end{bmatrix}$	2.8700
v	$\begin{bmatrix} 1.7989 \\ -0.2632 \\ -12.0215 \\ -6.2948 \end{bmatrix}$	20.5439	$\begin{bmatrix} 1.1678 \\ 0.0574 \\ -9.6540 \\ -5.9841 \end{bmatrix}$	2.0183
y	$\begin{bmatrix} -0.1165 \\ 0.6660 \\ -3.8249 \\ -4.1959 \end{bmatrix}$	30.1985	$\begin{bmatrix} -0.0064 \\ 0.6129 \\ -4.2815 \\ -4.1598 \end{bmatrix}$	2.3597
z	$\begin{bmatrix} 0.1962 \\ 1.7153 \\ -3.2896 \\ -6.9094 \end{bmatrix}$	21.4815	$\begin{bmatrix} 0.0792 \\ 1.1726 \\ -4.4356 \\ -4.8665 \end{bmatrix}$	2.0220

One of the basic problems in robotics is the description of the position and orientation of the end-effector in terms of the joint variables. There are two types of joints: rotational joints for rotating the associated robot link, and translational joints for pushing and pulling the associated robot link along a straight line. However,

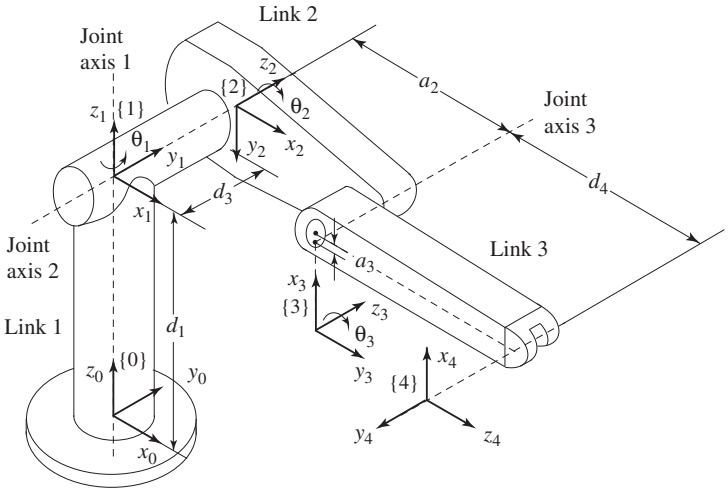

Fig. 9.4: A three-link robotic manipulator.

joints in industrial robots are almost always rotational. Fig. 9.49.4 shows a three-joint industrial robot, where the three joints can be used to rotate links 1, 2, and 3. In this case, the end-effector is located at the end of link 3, whose position and orientation can be conveniently described relative to a fixed coordinate system which is often referred to as a *frame* in robotics. As shown in Fig. 9.49.4, frame {0} is attached to the robot base and is fixed relative to the ground. Next, frames {1}, {2}, and {3} are attached to joint axes 1, 2, and 3, respectively, and are subject to the following rules:

- The z axis of frame $\{i\}$ is along the joint axis i for $i = 1,\ 2,\ 3$.
- The x axis of frame $\{i\}$ is perpendicular to the z axes of frames $\{i\}$ and $\{i+1\}$ for $i = 1,\ 2,\ 3$.
- The y axis of frame $\{i\}$ is determined such that frame $\{i\}$ is a standard right-hand coordinate system.
- Frame {4} is attached to the end of link 3 in such a way that the axes of frames {3} and {4} are in parallel and the distance between the z axes of these two frames is zero.

Having assigned the frames, the relation between two consecutive frames can be characterized by the so-called *Denavit-Hartenberg (D-H) parameters* [2] [**?**] which are defined in the following table:

a_i: distance from the z_i axis to the z_{i+1} axis measured along the x_i axis
α_i: angle between the z_i axis and the z_{i+1} axis measured about the x_i axis
d_i: distance from the x_{i-1} axis to the x_i axis measured along the z_i axis
θ_i: angle between the x_{i-1} axis and the x_i axis measured about the z_i axis

As can be observed in Fig. 9.49.4, parameters $d_1,\ a_2$, and d_4 in this case represent the lengths of links 1, 2, and 3, respectively, d_3 represents the offset between link 1

and link 2, and a_3 represents the offset between link 2 and link 3. In addition, the above frame assignment also determines the angles $\alpha_0 = 0°$, $\alpha_1 = -90°$, $\alpha_2 = 0°$, and $\alpha_3 = -90°$. Table 9.2 summarizes the D-H parameters of the three-joint robot in Fig. 9.49.4 where the only variable parameters are θ_1, θ_2, and θ_3 which represent the rotation angles of joints 1, 2, and 3, respectively.

Table 9.2 D-H parameters of 3-link robot

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	$0°$	0	d_1	θ_1
2	$-90°$	0	0	θ_2
3	$0°$	a_2	d_3	θ_3
4	$-90°$	a_3	d_4	$0°$

Since the D-H parameters a_{i-1}, α_{i-1}, d_i, and θ_i characterize the relation between frames $\{i-1\}$ and $\{i\}$, they can be used to describe the position and orientation of frame $\{i\}$ in relation to those of frame $\{i-1\}$. To this end, we define the so-called *homogeneous transformation* in terms of the 4×4 matrix

$$ {}^{i-1}_{i}\mathbf{T} = \left[\begin{array}{c:c} {}^{i-1}_{i}\mathbf{R} & {}^{i-1}\mathbf{p}_{i\text{ORG}} \\ \hdashline 0\ 0\ 0 & 1 \end{array}\right]_{4\times 4} \tag{9.10} $$

where vector ${}^{i-1}\mathbf{p}_{i\text{ORG}}$ denotes the position of the origin of frame $\{i\}$ with respect to frame $\{i-1\}$, and matrix ${}^{i-1}_{i}\mathbf{R}$ is an orthogonal matrix whose columns denote the x-, y-, and z-coordinate vectors of frame $\{i\}$ with respect to frame $\{i-1\}$. With the D-H parameters a_{i-1}, α_{i-1}, d_i, and θ_i known, the homogeneous transformation in Eq. (9.109.10) can be expressed as [2][**?**]

$$ {}^{i-1}_{i}\mathbf{T} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & \alpha_{i-1} & \alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9.11} $$

where $s\theta$ and $c\theta$ denote $\sin\theta$ and $\cos\theta$, respectively. The significance of the above formula is that it can be used to evaluate the position and orientation of the end-effector as

$$ {}^{0}_{N}\mathbf{T} = {}^{0}_{1}\mathbf{T}\,{}^{1}_{2}\mathbf{T}\cdots{}^{N-1}_{N}\mathbf{T} \tag{9.12} $$

where each ${}^{i-1}_{i}\mathbf{T}$ on the right-hand side can be obtained using Eq. (9.119.11). The formula in Eq. (9.129.12) is often referred to as the *equation of forward kinematics*.

Example 9.1 Derive closed-form formulas for the position and orientation of the robot tip in Fig. 9.49.4 in terms of joint angles θ_1, θ_2, and θ_3.

Solution Using Table 9.2 and Eq. (9.119.11), the homogeneous transformations ${}^{i-1}_{i}\mathbf{T}$ for $i = 1,\ 2,\ 3$, and 4 are obtained as

$$
{}^{0}_{1}\mathbf{T} = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad
{}^{1}_{2}\mathbf{T} = \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{2}_{3}\mathbf{T} = \begin{bmatrix} c_3 & -s_3 & 0 & a_2 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad
{}^{3}_{4}\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & a_3 \\ 0 & 0 & 1 & d_4 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

With $N = 4$, Eq. (9.129.12) gives

$$
\begin{aligned}
{}^{0}_{4}\mathbf{T} &= {}^{0}_{1}\mathbf{T}\,{}^{1}_{2}\mathbf{T}\,{}^{2}_{3}\mathbf{T}\,{}^{3}_{4}\mathbf{T} \\
&= \begin{bmatrix} c_1c_{23} & s_1 & -c_1s_{23} & c_1(a_2c_2 + a_3c_{23} - d_4s_{23}) - d_3s_1 \\ s_1c_{23} & -c_1 & -s_1s_{23} & s_1(a_2c_2 + a_3c_{23} - d_4s_{23}) + d_3c_1 \\ -s_{23} & 0 & -c_{23} & d_1 - a_2s_2 - a_3s_{23} - d_4c_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

where $c_1 = \cos\theta_1,\ s_1 = \sin\theta_1,\ c_{23} = \cos(\theta_2 + \theta_3)$, and $s_{23} = \sin(\theta_2 + \theta_3)$. Therefore, the position of the robot tip with respect to frame {0} is given by

$${}^{0}\mathbf{p}_{4\text{ORG}} = \begin{bmatrix} c_1(a_2c_2 + a_3c_{23} - d_4s_{23}) - d_3s_1 \\ s_1(a_2c_2 + a_3c_{23} - d_4s_{23}) + d_3c_1 \\ d_1 - a_2s_2 - a_3s_{23} - d_4c_{23} \end{bmatrix} \tag{9.13}$$

and the orientation of the robot tip with respect to frame {0} is characterized by the orthogonal matrix

$${}^{0}_{4}\mathbf{R} = \begin{bmatrix} c_1c_{23} & s_1 & -c_1s_{23} \\ s_1c_{23} & -c_1 & -s_1s_{23} \\ -s_{23} & 0 & -c_{23} \end{bmatrix} \tag{9.14}$$

■

9.3.2 Inverse kinematics problem

The joint angles of manipulator links are usually measured using sensors such as optical encoders that are attached to the link actuators. As discussed in Sec. 9.3.1, when the joint angles $\theta_1,\ \theta_2,\ \ldots,\ \theta_n$ are known, the position and orientation of the end-effector can be evaluated using Eq. (9.129.12). A related and often more important problem is the *inverse kinematics problem* which is as follows: find the joint angles θ_i for $1 \le i \le n$ with which the manipulator's end-effector would achieve a *prescribed* position and orientation. The significance of the inverse kinematics

lies in the fact that the tasks to be accomplished by a robot are usually in terms of trajectories in the Cartesian space that the robot's end-effector must follow. Under these circumstances, the position and orientation for the end-effector are known and the problem is to find the correct values of the joint angles that would move the robot's end-effector to the desired position and orientation.

Mathematically, the inverse kinematics problem can be described as the problem of finding the values θ_i for $1 \le i \le n$ that would satisfy Eq. (9.129.12) for a given ${}^0_N\mathbf{T}$. Since Eq. (9.129.12) is highly nonlinear, the problem of finding its solutions is not a trivial one [2][**?**]. For example, if a prescribed position of the end-effector for the three-link manipulator in Fig. 9.49.4 is given by ${}^0\mathbf{p}_{4\text{ORG}} = [p_x \; p_y \; p_z]^T$, then Eq. (9.139.13) gives

$$\begin{aligned} c_1(a_2c_2 + a_3c_{23} - d_4s_{23}) - d_3s_1 &= p_x \\ s_1(a_2c_2 + a_3c_{23} - d_4s_{23}) + d_3c_1 &= p_y \\ d_1 - a_2s_2 - a_3s_{23} - d_4c_{23} &= p_z \end{aligned} \tag{9.15}$$

In the next section, we illustrate an optimization approach for the solution of the inverse kinematics problem on the basis of Eq. (9.159.15).

9.3.3 Solution of inverse kinematics problem

If we let

$$\mathbf{x} = [\theta_1 \;\; \theta_2 \;\; \theta_3]^T \tag{9.16a}$$

$$f_1(\mathbf{x}) = c_1(a_2c_2 + a_3c_{23} - d_4s_{23}) - d_3s_1 - p_x \tag{9.16b}$$

$$f_2(\mathbf{x}) = s_1(a_2c_2 + a_3c_{23} - d_4s_{23}) + d_3c_1 - p_y \tag{9.16c}$$

$$f_3(\mathbf{x}) = d_1 - a_2s_2 - a_3s_{23} - d_4c_{23} - p_z \tag{9.16d}$$

then Eq. (9.159.15) is equivalent to

$$f_1(\mathbf{x}) = 0 \tag{9.17a}$$

$$f_2(\mathbf{x}) = 0 \tag{9.17b}$$

$$f_3(\mathbf{x}) = 0 \tag{9.17c}$$

To solve this system of nonlinear equations, we construct the objective function

$$F(\mathbf{x}) = f_1^2(\mathbf{x}) + f_2^2(\mathbf{x}) + f_3^2(\mathbf{x})$$

and notice that vector $\mathbf{x}^*$ solves Eq. (9.17) Eqs. (9.17a)–(9.17c) if and only if $F(\mathbf{x}^*) = 0$. Since function $F(\mathbf{x})$ is nonnegative, finding a solution point $\mathbf{x}$ for Eq. (9.17)Eqs. (9.17a)–(9.17c) amounts to finding a minimizer $\mathbf{x}^*$ at which $F(\mathbf{x}^*) = 0$. In other words, we can convert the inverse kinematics problem at hand into the unconstrained minimization problem

$$\text{minimize} \quad F(\mathbf{x}) = \sum_{k=1}^{3} f_k^2(\mathbf{x}) \tag{9.18}$$

An advantage of this approach over conventional methods for inverse kinematics problems [2][**?**] is that when the desired position $[p_x \ p_y \ p_z]^T$ is *not* within the manipulator's reach, the conventional methods will fail to work and a conclusion that no solution exists will be drawn. With the optimization approach, however, minimizing function $F(\mathbf{x})$ will still yield a minimizer, say, $\mathbf{x}^* = [\theta_1^* \ \theta_2^* \ \theta_3^*]^T$, although the objective function $F(\mathbf{x})$ would not become zero at $\mathbf{x}^*$. In effect, an *approximate* solution of the problem would be obtained, which could be entirely satisfactory in most engineering applications. We shall illustrate this point further in Example 9.2 by means of computer simulations.

To apply the minimization algorithms studied earlier, we let

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ f_3(\mathbf{x}) \end{bmatrix}$$

and compute the gradient of $F(\mathbf{x})$ as

$$\mathbf{g}(\mathbf{x}) = 2\mathbf{J}^T(\mathbf{x})\mathbf{f}(\mathbf{x}) \tag{9.19}$$

where the Jacobian matrix $\mathbf{J}(\mathbf{x})$ is given by

$$\begin{aligned} \mathbf{J}(\mathbf{x}) &= [\nabla f_1(\mathbf{x}) \ \nabla f_2(\mathbf{x}) \ \nabla f_3(\mathbf{x})]^T \\ &= \begin{bmatrix} -q_3 s_1 - d_3 c_1 & q_4 c_1 & q_2 c_1 \\ q_3 c_1 - d_3 s_1 & q_4 s_1 & q_2 s_1 \\ 0 & -q_3 & -q_1 \end{bmatrix} \end{aligned} \tag{9.20}$$

with $q_1 = a_2 c_{23} - d_4 s_{23}$, $q_2 = -a_3 s_{23} - d_4 c_{23}$, $q_3 = a_2 c_2 + q_1$, and $q_4 = -a_2 s_2 + q_2$. The Hessian of $F(\mathbf{x})$ is given by

$$\mathbf{H}(\mathbf{x}) = 2\mathbf{J}^T(\mathbf{x})\mathbf{J}(\mathbf{x}) + 2\sum_{k=1}^{3} f_k(\mathbf{x})\nabla^2 f_k(\mathbf{x}) \tag{9.21}$$

where $\nabla^2 f_k(\mathbf{x})$ is the Hessian of $f_k(\mathbf{x})$ (see Prob. 9.4).

Example 9.2 In the three-link manipulator depicted in Fig. 9.49.4, $d_1 = 66.04$ cm, $d_3 = 14.91$ cm, $d_4 = 43.31$ cm, $a_2 = 43.18$ cm, and $a_3 = 2.03$ cm. By applying a steepest-descent (SD), Newton (N), Gauss-Newton (GN), Fletcher-Reeves (FR) algorithm and then a quasi-Newton (QN) algorithm based on the Broyden-Fletcher-Goldfarb-Shanno updating formula in Eq. (7.57**??**), determine the joint angles $\theta_i(t)$ for $i = 1, 2, 3$ and $-\pi \le t \le \pi$ such that the manipulator's end-effector tracks the desired trajectory $\mathbf{p}_d(t) = [p_x(t) \ p_y(t) \ p_z(t)]^T$ where

$$p_x(t) = 30\cos t, \quad p_y(t) = 100\sin t, \quad p_z(t) = 10t + 66.04$$

for $-\pi \leq t \leq \pi$ as illustrated in Fig. 9.59.5.

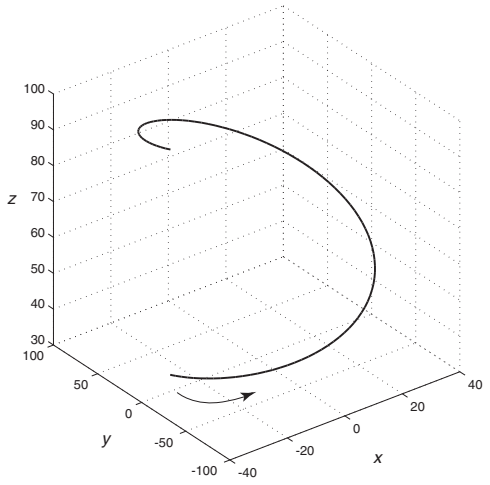

Fig. 9.5: Desired Cartesian trajectory for Example 9.2.

Solution The problem was solved by applying Algorithms 5.1, 5.5, and 6.3 as the steepest-descent, Gauss-Newton, and Fletcher-Reeves algorithm, respectively, using the inexact line search in Steps 1 to 6 of Algorithm 7.3 in each case. The Newton algorithm used was essentially Algorithm 5.3 incorporating the Hessian-matrix modification in Eq. (5.13**??**) as detailed below:

Algorithm 9.1 Newton algorithm
Step 1
Input $\mathbf{x}_0$ and initialize the tolerance ε.
Set $k = 0$.
Step 2
Compute $\mathbf{g}_k$ and $\mathbf{H}_k$.
Step 3
Compute the eigenvalues of $\mathbf{H}_k$ (see Sec. A.5).
Determine the smallest eigenvalue of $\mathbf{H}_k$, λ_{min}.
Modify matrix $\mathbf{H}_k$ to

$$\hat{\mathbf{H}}_k = \begin{cases} \mathbf{H}_k & \text{if } \lambda_{min} > 0 \\ \mathbf{H}_k + \gamma \mathbf{I}_n & \text{if } \lambda_{min} \leq 0 \end{cases}$$

where

$$\gamma = -1.05\lambda_{min} + 0.1$$

Step 4
Compute $\hat{\mathbf{H}}_k^{-1}$ and $\mathbf{d}_k = -\hat{\mathbf{H}}_k^{-1}\mathbf{g}_k$
Step 5
Find α_k, the value of α that minimizes $f(\mathbf{x}_k + \alpha \mathbf{d}_k)$, using the inexact line search in Steps 1 to 6 of Algorithm 7.3.
Step 6
Set $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$.
Compute $f_{k+1} = f(\mathbf{x}_{k+1})$.
Step 7
If $\|\alpha_k \mathbf{d}_k\| < \varepsilon$, then do:
Output $\mathbf{x}^* = \mathbf{x}_{k+1}$ and $f(\mathbf{x}^*) = f_{k+1}$, and stop.
Otherwise, set $k = k + 1$ and repeat from Step 2.

The quasi-Newton algorithm used was essentially Algorithm 7.3 with a slightly modified version of Step 8b as follows:

Step 8b′
Compute $D = \boldsymbol{\delta}_k^T \boldsymbol{\gamma}_k$. If $D \leq 0$, then set $\mathbf{S}_{k+1} = \mathbf{I}_n$, otherwise, compute $\mathbf{S}_{k+1}$ using Eq. (7.57**??**).

At $t = t_k$, the desired trajectory can be described in terms of its Cartesian coordinates as

$$\mathbf{p}_d(t_k) = \begin{bmatrix} p_x(t_k) \\ p_y(t_k) \\ p_z(t_k) \end{bmatrix} = \begin{bmatrix} 30\cos t_k \\ 100\sin t_k \\ 10t_k + 66.04 \end{bmatrix}$$

where $-\pi \leq t_k \leq \pi$. Assuming 100 uniformly spaced sample points, the solution of the system of equations in Eq. (9.17) Eqs. (9.17a)–(9.17c) can obtained by solving the minimization problem in Eq. (9.189.18) for $k = 1, 2, \ldots, 100$, i.e., for $t_k = -\pi, \ldots, \pi$, using the specified D-H parameters. Since the gradient and Hessian of $F(\mathbf{x})$ are available (see Eqs. (9.199.19) and (9.219.21)), the problem can be solved using each of the five optimization algorithms specified in the description of the problem to obtain a minimizer $\mathbf{x}^*(t_k)$ in each case. If the objective function $F(\mathbf{x})$ turns out to be zero at $\mathbf{x}^*(t_k)$, then $\mathbf{x}^*(t_k)$ satisfies Eq. (9.17) Eqs. (9.17a)–(9.17c), and the joint angles specified by $\mathbf{x}^*(t_k)$ lead the manipulator's end-effector to the desired position precisely. On the other hand, if $F[\mathbf{x}^*(t_k)]$ is nonzero, then $\mathbf{x}^*(t_k)$ is taken as an approximate solution of the inverse kinematics problem at instant t_k.

Once the minimizer $\mathbf{x}^*(t_k)$ is obtained, the above steps can be repeated at $t = t_{k+1}$ to obtain solution point $\mathbf{x}^*(t_{k+1})$. Since t_{k+1} differs from t_k only by a small amount and the profile of optimal joint angles is presumably continuous, $\mathbf{x}^*(t_{k+1})$ is expected

to be in the vicinity of $\mathbf{x}^*(t_k)$. Therefore, the previous solution $\mathbf{x}^*(t_k)$ can be used as a reasonable initial point for the next optimization.[1]

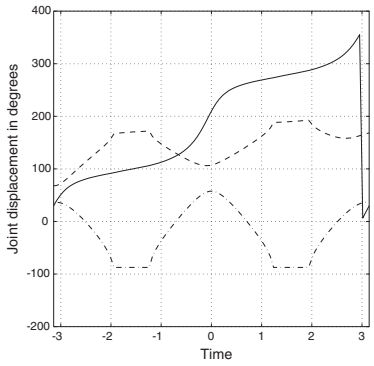

Fig. 9.6: figure
Optimal joint angles $\theta_1^*(t)$ (solid line), $\theta_2^*(t)$ (dashed line), and $\theta_3^*(t)$ (dot-dashed line).

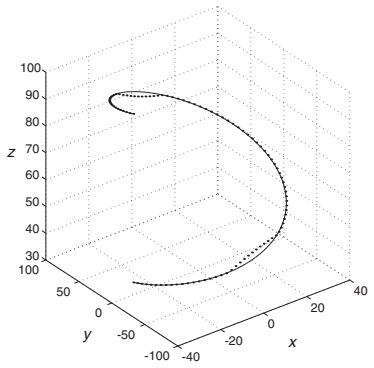

Fig. 9.7: figure
End-effector's profile (dotted line) and the desired trajectory (solid line).

The five optimization algorithms were applied to the problem at hand and were all found to work although with different performance in terms of solution accuracy and computational complexity. The solution obtained using the BFGS algorithm, $\mathbf{x}^*(t_k) = [\theta_1^*(t_k)\ \theta_2^*(t_k)\ \theta_3^*(t_k)]^T$ for $1 \le k \le 100$, is plotted in Fig. 9.69.6; the tracking profile of the end-effector is plotted as the dotted curve in Fig. 9.79.7 and is compared with the desired trajectory which is plotted as the solid curve. It turns out that the desired positions $\mathbf{p}_d(t_k)$ for $20 \le k \le 31$ and $70 \le k \le 81$ are beyond the manipulator's reach. As a result, we see in Fig. 9.79.7 that there are two small portions of the tracking profile that deviate from the desired trajectory, but even in this case, the corresponding $\mathbf{x}^*(t_k)$ still offers a reasonable approximate solution. The remaining part of the tracking profile coincides with the desired trajectory almost perfectly which simply means that for the desired positions within the manipulator's work space, $\mathbf{x}^*(t_k)$ offers a nearly exact solution.

The performance of the five algorithms in terms of the normalized CPU time and iterations per sample point and the error at sample points within and outside the work space is summarized in Table 9.3. The data supplied are in the form of averages with respect to 100 runs of the algorithms using random initializations. As can be seen, the average errors within the manipulator's work space for the solutions $\mathbf{x}^*(t_k)$ obtained using the steepest-descent and Fletcher-Reeves algorithms are much larger than those obtained using the Newton, Gauss-Newton, and BFGS algorithms, although the solutions obtained are still acceptable considering the relatively large

[1] Choosing the initial point on the basis of *any* knowledge about the solution instead of a random initial point can lead to a large reduction in the amount of computation in most optimization problems.

size of the desired trajectory. Andreas The best results in terms of efficiency as well as accuracy are obtained by using the Newton and BFGS algorithms.

Table 9.3 Performance comparisons for Example 9.2

Algorithm	Normalized CPU time	Average number of iterations per sample point	Average error within work space	Average error outside work space
SD	1.0	47	0.0488	4.3750
N	0.0553	4	4.0161×10^{-8}	4.3737
GN	0.1060	5	1.5578×10^{-4}	4.3739
FR	0.4516	14	0.1778	4.3743
BFGS	0.1244	6	2.8421×10^{-5}	4.3737

■

9.4 Design of Digital Filters

In this section, we will apply unconstrained optimization for the design of FIR digital filters. Different designs are possible depending on the type of FIR filter required and the formulation of the objective function. The theory and design principles of digital filters are quite extensive [3][**?**] and are beyond the scope of this book. To facilitate the understanding of the application of unconstrained optimization to the design of digital filters, we present a brief review of the highlights of the theory, properties, and characterization of digital filters in Appendix B, which should prove quite adequate in the present context.

The one design aspect of digital filters that can be handled quite efficiently with optimization is the approximation problem whereby the parameters of the filter have to be chosen to achieve a specified type of frequency response. Below, we examine two different designs (see Sec. B.9). In one design, we formulate a weighted least-squares objective function, i.e., one based on the square of the L_2 norm, for the design of linear-phase FIR filters and in another we obtain a minimax objective function, i.e., one based on the L_∞ norm.

The L_p norm of a vector where $p \geq 1$ is defined in Sec. A.8.1. Similarly, the L_p norm of a function $F(\omega)$ of a continuous variable ω can be defined with respect to the interval $[a, b]$ as

$$||F(\omega)||_p = \left(\int_a^b |F(\omega)|^p \, d\omega \right)^{1/p} \tag{9.22}$$

where $p \geq 1$ and if

$$\int_a^b |F(\omega)|^p \, d\omega \leq K < \infty$$

the L_p norm of $F(\omega)$ exists. If $F(\omega)$ is bounded with respect to the interval $[a, b]$, i.e., $|F(\omega)| \leq M$ for $\omega \in [a,\ b]$ where M is finite, then the L_∞ norm of $F(\omega)$ is defined

as

$$\|F(\omega)\|_\infty = \max_{a\le\omega\le b} |F(\omega)| \tag{9.23a}$$

and as in the case of the L_∞ norm of a vector, it can be verified that

$$\lim_{p\to\infty} \|F(\omega)\|_p = \|F(\omega)\|_\infty \tag{9.23b}$$

(see Sec. B.9.1).

9.4.1 Weighted least-squares design of FIR filters

As shown in Sec. B.5.1, an FIR filter is completely specified by its transfer function which assumes the form

$$H(z) = \sum_{n=0}^{N} h_n z^{-n} \tag{9.24}$$

where the coefficients h_n for $n = 0,\ 1,\ \ldots,\ n$ represent the impulse response of the filter.

9.4.1.1 Specified frequency response

Assuming a normalized sampling frequency of 2π, which corresponds to a normalized sampling period $T = 1$ s, the frequency response of an FIR filter is obtained as $H(e^{j\omega})$ by letting $z = e^{j\omega}$ in the transfer function (see Sec. B.8). In practice, the frequency response is required to approach some desired frequency response, $H_d(\omega)$, to within a specified error. Hence an FIR filter can be designed by formulating an objective function based on the difference between the actual and desired frequency responses (see Sec. B.9.3). Except in some highly specialized applications, the transfer function coefficients (or impulse response values) of a digital filters are real and, consequently, knowledge of the frequency response of the filter with respect to the positive half of the baseband fully characterizes the filter (see Sec. B.8). Under these circumstances, a weighted least-squares objective function that can be used to design FIR filters can be constructed as

$$e(\mathbf{x}) = \int_0^\pi W(\omega)|H(e^{j\omega}) - H_d(\omega)|^2\, d\omega \tag{9.25}$$

where $\mathbf{x} = [h_0\ h_1\ \cdots\ h_N]^T$ is an $N+1$-dimensional variable vector representing the transfer function coefficients, ω is a normalized frequency variable which is assumed to be in the range 0 to π rad/s, and $W(\omega)$ is a predefined weighting function. The design is accomplished by finding the vector $\mathbf{x}^*$ that minimizes $e(\mathbf{x})$, and this can be efficiently done by means of unconstrained optimization.

Weighting is used to emphasize or deemphasize the objective function with respect to one or more ranges of ω. Without weighting, an optimization algorithm would tend to minimize the objective function uniformly with respect to ω. Thus if the objective function is multiplied by a weighting constant larger than unity for values of ω in a certain critical range but is left unchanged for all other frequencies, a reduced value of the objective function will be achieved with respect to the critical frequency range. This is due to the fact that the weighted objective function will tend to be minimized uniformly and thus the actual unweighted objective function will tend to be scaled down in proportion to the inverse of the weighting constant in the critical range of ω relative to its value at other frequencies. Similarly, if a weighting constant of value less than unity is used for a certain uncritical frequency range, an increased value of the objective will be the outcome with respect to the uncritical frequency range. Weighting is very important in practice because through the use of suitable scaling, the designer is often able to design a more economical filter for the required specifications. In the above example, the independent variable is frequency. In other applications, it could be time or some other independent parameter.

An important step in an optimization-based design is to express the objective function in terms of variable vector $\mathbf{x}$ *explicitly*. This facilitates the evaluation of the gradient and Hessian of the objective function. To this end, if we let

$$\mathbf{c}(\omega) = [1 \ \cos\omega \ \cdots \ \cos N\omega]^T \tag{9.26a}$$

$$\mathbf{s}(\omega) = [0 \ \sin\omega \ \cdots \ \sin N\omega]^T \tag{9.26b}$$

the frequency response of the filter can be expressed as

$$H(e^{j\omega}) = \sum_{n=0}^{N} h_n \cos n\omega - j \sum_{n=0}^{N} h_n \sin n\omega = \mathbf{x}^T \mathbf{c}(\omega) - j\mathbf{x}^T \mathbf{s}(\omega) \tag{9.27}$$

If we let

$$H_d(\omega) = H_r(\omega) - jH_i(\omega) \tag{9.28}$$

where $H_r(\omega)$ and $-H_i(\omega)$ are the real and imaginary parts of $H_d(\omega)$, respectively, then Eqs. (9.279.27) and (9.289.28) give

$$\begin{aligned} |H(e^{j\omega}) - H_d(\omega)|^2 &= [\mathbf{x}^T \mathbf{c}(\omega) - H_r(\omega)]^2 + [\mathbf{x}^T \mathbf{s}(\omega) - H_i(\omega)]^2 \\ &= \mathbf{x}^T [\mathbf{c}(\omega)\mathbf{c}^T(\omega) + \mathbf{s}(\omega)\mathbf{s}^T(\omega)]\mathbf{x} \\ &\quad -2\mathbf{x}^T [\mathbf{c}(\omega)H_r(\omega) + \mathbf{s}(\omega)H_i(\omega)] + |H_d(\omega)|^2 \end{aligned}$$

Therefore, the objective function in Eq. (9.259.25) can be expressed as a quadratic function with respect to $\mathbf{x}$ of the form

$$e(\mathbf{x}) = \mathbf{x}^T \mathbf{Q}\mathbf{x} - 2\mathbf{x}^T \mathbf{b} + \kappa \tag{9.29}$$

where κ is a constant[2] and

[2] Symbol κ will be used to represent a constant throughout this chapter.

$$\mathbf{Q} = \int_0^{\pi} W(\omega)[\mathbf{c}(\omega)\mathbf{c}^T(\omega) + \mathbf{s}(\omega)\mathbf{s}^T(\omega)]\, d\omega \tag{9.30}$$

$$\mathbf{b} = \int_0^{\pi} W(\omega)[H_r(\omega)\mathbf{c}(\omega) + H_i(\omega)\mathbf{s}(\omega)]\, d\omega \tag{9.31}$$

Matrix $\mathbf{Q}$ in Eq. (9.309.30) is positive definite (see Prob. 9.5). Hence the objective function $e(\mathbf{x})$ in Eq. (9.299.29) is globally strictly convex and has a unique global minimizer $\mathbf{x}^*$ given by

$$\mathbf{x}^* = \mathbf{Q}^{-1}\mathbf{b} \tag{9.32}$$

For the design of high-order FIR filters, the matrix $\mathbf{Q}$ in Eq. (9.309.30) is of a large size and the methods described in Sec. 6.4 can be used to find the minimizer without obtaining the inverse of matrix $\mathbf{Q}$.

9.4.1.2 Linear phase response

The frequency response of an FIR digital filter of order N (or length $N+1$) with linear phase response is given by

$$H(e^{j\omega}) = e^{-j\omega N/2}A(\omega) \tag{9.33}$$

Assuming an even-order filter, function $A(\omega)$ in Eq. (9.339.33) can be expressed as

$$A(\omega) = \sum_{n=0}^{N/2} a_n \cos n\omega \tag{9.34a}$$

$$a_n = \begin{cases} h_{N/2} & \text{for } n = 0 \\ 2h_{N/2-n} & \text{for } n \neq 0 \end{cases} \tag{9.34b}$$

(see Sec. B.9.2) and if the desired frequency response is assumed to be of the form

$$H_d(\omega) = e^{-j\omega N/2}A_d(\omega)$$

then the least-squares objective function

$$e_l(\mathbf{x}) = \int_0^{\pi} W(\omega)[A(\omega) - A_d(\omega)]^2\, d\omega \tag{9.35a}$$

can be constructed where the variable vector is given by

$$\mathbf{x} = [a_0\ a_1\ \cdots\ a_{N/2}]^T \tag{9.35b}$$

If we now let

$$\mathbf{c}_l(\omega) = [1\ \cos\omega\ \cdots\ \cos N\omega/2]^T \tag{9.36a}$$

$A(\omega)$ can be written in terms of the inner product $\mathbf{x}^T\mathbf{c}_l(\omega)$ and the objective function $e_l(\mathbf{x})$ in Eq. (9.35a9.35a) can be expressed as

$$e_l(\mathbf{x}) = \mathbf{x}^T \mathbf{Q}_l \mathbf{x} - 2\mathbf{x}^T \mathbf{b}_l + \kappa \tag{9.36b}$$

where κ is a constant, as before, with

$$\mathbf{Q}_l = \int_0^{\pi} W(\omega)\mathbf{c}_l(\omega)\mathbf{c}_l^T(\omega)\, d\omega \tag{9.37a}$$

$$\mathbf{b}_l = \int_0^{\pi} W(\omega)A_d(\omega)\mathbf{c}_l(\omega)\, d\omega \tag{9.37b}$$

Like matrix $\mathbf{Q}$ in Eq. (9.309.30), matrix $\mathbf{Q}_l$ in Eq. (9.37a9.37a) is positive definite; hence, like the objective function $e(\mathbf{x})$ in Eq. (9.299.29), $e_l(\mathbf{x})$ in Eq. (9.36b9.36b) is globally strictly convex and its unique global minimizer is given in closed form by

$$\mathbf{x}_l^* = \mathbf{Q}_l^{-1}\mathbf{b}_l \tag{9.38}$$

For filters of order less than 200, matrix $\mathbf{Q}_l$ in Eq. (9.389.38) is of size less than 100, and the formula in Eq. (9.389.38) requires a moderate amount of computation. For higher-order filters, the closed-form solution given in Eq. (9.389.38) becomes computationally very demanding and methods that do not require the computation of the inverse of matrix $\mathbf{Q}_l$ such as those studied in Sec. 6.4 would be preferred.

Example 9.3

(*a*) Applying the above method, formulate the design of an even-order linear-phase lowpass FIR filter assuming the desired amplitude response

$$A_d(\omega) = \begin{cases} 1 & \text{for } 0 \le \omega \le \omega_p \\ 0 & \text{for } \omega_a \le \omega \le \pi \end{cases} \tag{9.39}$$

where ω_p and ω_a are the passband and stopband edges, respectively (see Sec. B.9.1). Assume a normalized sampling frequency of 2π rad/s.

(*b*) Using the formulation in part (*a*), design FIR filters with $\omega_p = 0.45\pi$ and $\omega_a = 0.5\pi$ for filter orders of 20, 40, 60, and 80.

Solution (*a*) A suitable weighting function $W(\omega)$ for this problem is

$$W(\omega) = \begin{cases} 1 & \text{for } 0 \le \omega \le \omega_p \\ \gamma & \text{for } \omega_a \le \omega \le \pi \\ 0 & \text{elsewhere} \end{cases} \tag{9.40}$$

The value of γ can be chosen to emphasize or deemphasize the error function in the stopband relative to that in the passband. Since $W(\omega)$ is piecewise constant, the matrix $\mathbf{Q}_l$ in Eq. (9.37a9.37a) can be written as

$$\mathbf{Q}_l = \mathbf{Q}_{l1} + \mathbf{Q}_{l2}$$

where

$$\mathbf{Q}_{l1} = \int_0^{\omega_p} \mathbf{c}_l(\omega)\mathbf{c}_l^T(\omega)\, d\omega = \{q_{ij}^{(1)}\} \qquad \text{for } 1 \le i,\ j \le \frac{N+2}{2} \tag{9.41a}$$

and

$$\mathbf{Q}_{l2} = \gamma \int_{\omega_a}^{\pi} \mathbf{c}_l(\omega)\mathbf{c}_l^T(\omega)\, d\omega = \{q_{ij}^{(2)}\} \qquad \text{for } 1 \le i,\ j \le \frac{N+2}{2} \tag{9.41b}$$

with

$$q_{ij}^{(1)} = \begin{cases} \dfrac{\omega_p}{2} + \dfrac{\sin[2(i-1)\omega_p]}{4(i-1)} & \text{for } i = j \\ \dfrac{\sin[(i-j)\omega_p]}{2(i-j)} + \dfrac{\sin[(i+j-2)\omega_p]}{2(i+j-2)} & \text{for } i \ne j \end{cases} \tag{9.42a}$$

and

$$q_{ij}^{(2)} = \begin{cases} \gamma\left[\dfrac{(\pi-\omega_a)}{2} - \dfrac{\sin[2(i-1)\omega_a]}{4(i-1)}\right] & \text{for } i = j \\ -\dfrac{\gamma}{2}\left[\dfrac{\sin[(i-j)\omega_a]}{(i-j)} + \dfrac{\sin[(i+j-2)\omega_a]}{(i+j-2)}\right] & \text{for } i \ne j \end{cases} \tag{9.42b}$$

Note that for $i = j = 1$, the expressions in Eq. (9.42) Eqs. (9.42a) and (9.42b) are evaluated by taking the limit as $i \to 1$, which implies that

$$q_{11}^{(1)} = \omega_p \quad \text{and} \quad q_{11}^{(2)} = \gamma(\pi - \omega_a) \tag{9.42c}$$

Vector $\mathbf{b}_l$ in Eq. (9.37b9.37b) is calculated as

$$\mathbf{b}_l = \int_0^{\omega_p} \mathbf{c}_l(\omega)\, d\omega = \{b_n\} \tag{9.43a}$$

with

$$b_n = \frac{\sin[(n-1)\omega_p]}{(n-1)} \qquad \text{for } 1 \le n \le \frac{N+2}{2} \tag{9.43b}$$

As before, for $n = 1$, the expression in Eq. (9.43b9.43b) is evaluated by taking the limit as $n \to 1$, which gives

$$b_1 = \omega_p \tag{9.43c}$$

(*b*) Optimal weighted least-squares designs for the various values of N were obtained by computing the minimizer $\mathbf{x}_l^*$ given by Eq. (9.389.38) and then evaluating the filter coefficients $\{h_i\}$ using Eq. (9.34b9.34b). The weighting constant γ was assumed to be 25. The amplitude responses of the FIR filters obtained are plotted in Fig. 9.89.8. ■

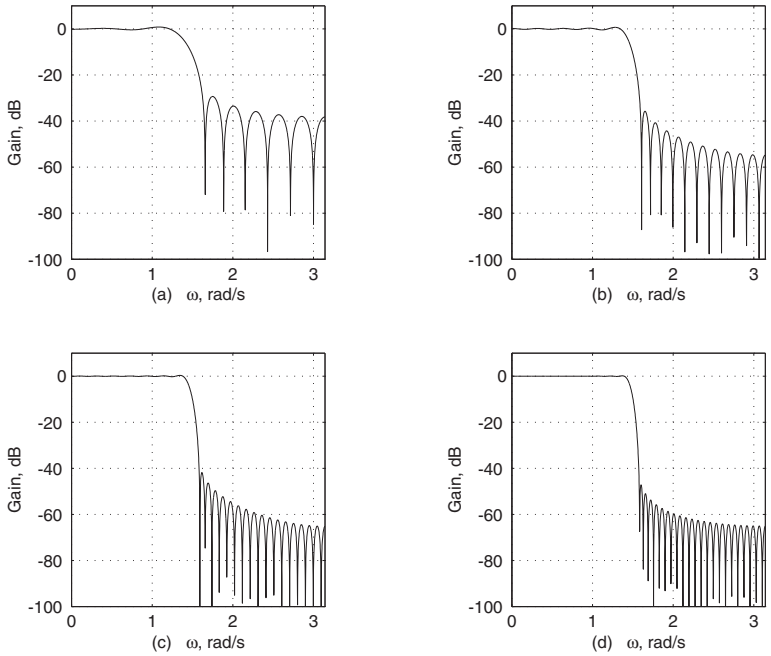

Fig. 9.8: Amplitude responses of the filters in Example 9.3: (a) $N = 20$, (b) $N = 40$, (c) $N = 60$, (d) $N = 80$.

9.4.2 Minimax design of FIR filters

The Parks-McClellan algorithm and its variants have been the most efficient tools for the minimax design of FIR digital filters [3]–[5] [**?**][**?**][**?**]. However, these algorithms apply only to the class of linear-phase FIR filters. The group delay introduced by these filters is constant and independent of frequency in the entire baseband (see Sec. B.8) but it can be quite large. In practice, a variable group delay in stopbands is of little concern and by allowing the phase response to be nonlinear in stopbands, FIR filters can be designed with constant group delay with respect to the passbands, which is significantly reduced relative to that achieved with filters that have a constant group delay throughout the entire baseband.

This section presents a least-pth approach to the design of low-delay FIR filters. For FIR filters, the weighted L_p error function with an even integer p can be shown to be globally convex.[3] This property, in conjunction with the availability of the gradient and Hessian of the objective function in closed form, enables us to develop an unconstrained optimization method for the design problem at hand.

[3] Note that this property does not apply to infinite-duration impulse response (IIR) filters [3][**?**].

9.4.2.1 Objective function

Given a desired frequency response $H_d(\omega)$ for an FIR filter, we want to determine the coefficients $\{h_n\}$ in the transfer function

$$H(z) = \sum_{n=0}^{N} h_n z^{-n} \tag{9.44}$$

such that the weighted L_{2p} approximation error

$$f(\mathbf{h}) = \left[\int_0^{\pi} W(\omega) |H(e^{j\omega}) - H_d(\omega)|^{2p} \, d\omega \right]^{1/2p} \tag{9.45}$$

is minimized, where $W(\omega) \geq 0$ is a weighting function, p is a positive integer, and $\mathbf{h} = [h_0 \; h_1 \; \cdots \; h_N]^T$.

If we let

$$\begin{aligned} H_d(\omega) &= H_{dr}(\omega) - jH_{di}(\omega) \\ \mathbf{c}(\omega) &= [1 \; \cos\omega \; \cdots \; \cos N\omega]^T \\ \mathbf{s}(\omega) &= [0 \; \sin\omega \; \cdots \; \sin N\omega]^T \end{aligned}$$

then Eq. (9.459.45) becomes

$$f(\mathbf{h}) = \left\{ \int_0^{\pi} W[(\mathbf{h}^T\mathbf{c} - H_{dr})^2 + (\mathbf{h}^T\mathbf{s} - H_{di})^2]^p \, d\omega \right\}^{1/2p} \tag{9.46}$$

where for simplicity the frequency dependence of W, $\mathbf{c}$, $\mathbf{s}$, H_{dr}, and H_{di} has been omitted. Now if we let

$$e_2(\omega) = [\mathbf{h}^T\mathbf{c}(\omega) - H_{dr}(\omega)]^2 + [\mathbf{h}^T\mathbf{s}(\omega) - H_{di}(\omega)]^2 \tag{9.47}$$

then the objective function can be expressed as

$$f(\mathbf{h}) = \left[\int_0^{\pi} W(\omega) e_2^p(\omega) \, d\omega \right]^{1/2p} \tag{9.48}$$

9.4.2.2 Gradient and Hessian of $f(\mathbf{h})$

Using Eq. (9.489.48), the gradient and Hessian of objective function $f(\mathbf{h})$ can be readily obtained as

$$\nabla f(\mathbf{h}) = f^{1-2p}(\mathbf{h}) \int_0^{\pi} W(\omega) e_2^{p-1}(\omega) \mathbf{q}(\omega) \, d\omega \tag{9.49a}$$

where

$$\mathbf{q}(\omega) = [\mathbf{h}^T\mathbf{c}(\omega) - H_{dr}(\omega)]\mathbf{c}(\omega) + [\mathbf{h}^T\mathbf{s}(\omega) - H_{di}(\omega)]\mathbf{s}(\omega) \tag{9.49b}$$

and

$$\nabla^2 f(\mathbf{h}) = \mathbf{H}_1 + \mathbf{H}_2 - \mathbf{H}_3 \tag{9.49c}$$

where

$$\mathbf{H}_1 = 2(p-1) f^{1-2p}(\mathbf{h}) \int_0^{\pi} W(\omega) e_2^{p-2}(\omega) \mathbf{q}(\omega) \mathbf{q}^T(\omega)\, d\omega \tag{9.49d}$$

$$\mathbf{H}_2 = f^{1-2p}(\mathbf{h}) \int_0^{\pi} W(\omega) e_2^{p-1}(\omega) [\mathbf{c}(\omega)\mathbf{c}^T(\omega) + \mathbf{s}(\omega)\mathbf{s}^T(\omega)]\, d\omega \tag{9.49e}$$

$$\mathbf{H}_3 = (2p-1) f^{-1}(\mathbf{h}) \nabla f(\mathbf{h}) \nabla^T f(\mathbf{h}) \tag{9.49f}$$

respectively.

Of central importance to the present algorithm is the property that for each and every positive integer p, the weighted L_{2p} objective function defined in Eq. (9.459.45) is convex in the entire parameter space $\mathcal{R}^{N+1}$. This property can be proved by showing that the Hessian $\nabla^2 f(\mathbf{h})$ is positive semidefinite for all $\mathbf{h} \in R^{N+1}$ (see Prob. 9.9).

9.4.2.3 Design algorithm

It is now quite clear that an FIR filter whose frequency response approximates a rather arbitrary frequency response $H_d(\omega)$ to within a given tolerance in the minimax sense can be obtained by minimizing $f(\mathbf{h})$ in Eq. (9.459.45) with a sufficiently large p. It follows from the above discussion that for a given p, $f(\mathbf{h})$ has a unique global minimizer. Therefore, any descent minimization algorithm, e.g., the steepest-descent, Newton, and quasi-Newton methods studied in previous chapters, can, in principle, be used to obtain the minimax design regardless of the initial design chosen. The amount of computation required to obtain the design is largely determined by the choice of optimization method as well as the initial point assumed.

A reasonable initial point can be deduced by using the L_2-optimal design obtained by minimizing $f(\mathbf{h})$ in Eq. (9.459.45) with $p = 1$. We can write

$$f(\mathbf{h}) = (\mathbf{h}^T \mathbf{Q} \mathbf{h} - 2\mathbf{h}^T \mathbf{p} + \kappa)^{1/2} \tag{9.50a}$$

where

$$\mathbf{Q} = \int_0^{\pi} W(\omega) [\mathbf{c}(\omega)\mathbf{c}^T(\omega) + \mathbf{s}(\omega)\mathbf{s}^T(\omega)]\, d\omega \tag{9.50b}$$

$$\mathbf{p} = \int_0^{\pi} W(\omega) [H_{dr}(\omega)\mathbf{c}(\omega) + H_{di}(\omega)\mathbf{s}(\omega)]\, d\omega \tag{9.50c}$$

Since $\mathbf{Q}$ is positive definite, the global minimizer of $f(\mathbf{h})$ in Eq. (9.50a9.50a) can be obtained as the solution of the linear equation

$$\mathbf{Q}\mathbf{h} = \mathbf{p} \tag{9.51}$$

We note that **Q** in Eq. (9.519.51) is a symmetric Toeplitz matrix[4] for which fast algorithms are available to compute the solution of Eq. (9.519.51) [6] [**?**].

The minimization of convex objective function $f(\mathbf{h})$ can be accomplished in a number of ways. Since the gradient and Hessian of $f(\mathbf{h})$ are available in closed-form and $\nabla^2 f(\mathbf{h})$ is positive semidefinite, the Newton method and the family of quasi-Newton methods are among the most appropriate.

From Eqs. (9.48) and (9.49) Eq. (9.48) and Eqs. (9.49a)–(9.49f), we note that $f(\mathbf{h})$, $\nabla f(\mathbf{h})$, and $\nabla^2 f(\mathbf{h})$ all involve integration which can be carried out using numerical methods. In computing $\nabla^2 f(\mathbf{h})$, the error introduced in the numerical integration can cause the Hessian to lose its positive definiteness but the problem can be easily fixed by modifying $\nabla^2 f(\mathbf{h})$ to $\nabla^2 f(\mathbf{h}) + \varepsilon \mathbf{I}$ where ε is a small positive scalar.

9.4.2.4 Direct and sequential optimizations

With a power p, weighting function $W(\omega)$, and an initial $\mathbf{h}$, say, $\mathbf{h}_0$, chosen, the design can be obtained directly or indirectly.

In a direct optimization, one of the unconstrained optimization methods is applied to minimize the L_{2p} objective function in Eq. (9.489.48) directly. Based on rather extensive trials, it was found that to achieve a near-minimax design, the value of p should be larger than 20 and for high-order FIR filters a value comparable to the filter order N should be used.

In sequential optimization, an L_{2p} optimization is first carried out with $p = 1$. The minimizer thus obtained, $\mathbf{h}^*$, is then used as the initial point in another optimization with $p = 2$. The same procedure is repeated for $p = 4, 8, 16, \ldots$ until the reduction in the objective function between two successive optimizations is less than a prescribed tolerance.

Example 9.4 Using the above direct and sequential approaches first with a Newton and then with a quasi-Newton algorithm, design a lowpass FIR filter of order $N = 54$ that would have approximately constant passband group delay of 23 s. Assume idealized passband and stopband gains of 1 and 0, respectively; a normalized sampling frequency $\omega_s = 2\pi$; passband edge $\omega_p = 0.45\pi$ and stopband edge $\omega_a = 0.55\pi$; $W(\omega) = 1$ in both the passband and stopband, and $W(\omega) = 0$ elsewhere.

Solution The design was carried out using the direct approach with $p = 128$ and the sequential approach with $p = 2, 4, 8, \ldots, 128$ by minimizing the objective function in Eq. (9.489.48) with the Newton algorithm and a quasi-Newton algorithm with the BFGS updating formula in Eq. (7.57**??**). The Newton algorithm used was essentially the same as Algorithm 9.1 (see solution of Example 9.2) except that Step 3 was replaced by the following modified Step 3:

Step 3′
Modify matrix $\mathbf{H}_k$ to $\hat{\mathbf{H}}_k = \mathbf{H}_k + 0.1\mathbf{I}_n$

[4] A Toeplitz matrix is a matrix whose entries along each diagonal are constant [6] [**?**]

The quasi-Newton algorithm used was Algorithm 7.3 with the modifications described in the solution of Example 9.2.

A lowpass FIR filter that would satisfy the required specifications can be obtained by assuming a complex-valued idealized frequency response of the form

$$H_d(\omega) = \begin{cases} e^{-j23\omega} & \text{for } \omega \in [0,\ \omega_p] \\ 0 & \text{for } \omega \in [\omega_a,\ \omega_s/2] \end{cases}$$
$$= \begin{cases} e^{-j23\omega} & \text{for } \omega \in [0,\ 0.45\pi] \\ 0 & \text{for } \omega \in [0.55\pi,\ \pi] \end{cases}$$

(see Sec. B.9.2). The integrations in Eqs. (9.489.48), (9.49a9.49a), and (9.49c9.49c) can be carried out by using one of several available numerical methods for integration. A fairly simple and economical approach, which works well in optimization, is as follows: Given a continuous function $f(\omega)$ of ω, an approximate value of its integral over the interval $[a,\ b]$ can be obtained as

$$\int_a^b f(\omega)d\omega \approx \delta \sum_{i=1}^{K} f(\omega_i)$$

where $\delta = (b-a)/K$ and $\omega_1 = a+\delta/2$, $\omega_2 = a+3\delta/2$, …, $\omega_K = a+(2K-1)\delta/2$. That is, we divide interval $[a,\ b]$ into K subintervals, add the values of the function at the midpoints of the K subintervals, and then multiply the sum obtained by δ.

The objective function in Eq. (9.489.48) was expressed as

$$f(\mathbf{h}) = \left[\int_0^{0.45\pi} e_2^p(\omega)d\omega\right]^{1/2p} + \left[\int_{0.55\pi}^{\pi} e_2^p(\omega)d\omega\right]^{1/2p}$$

and each integral was evaluated using the above approach with $K = 500$. The integrals in Eqs. (9.49a9.49a) and (9.49c9.49c) were evaluated in the same way.

The initial $\mathbf{h}$ was obtained by applying L_2 optimization to Eq. (9.50) Eqs. (9.50a)–(9.50c). All trials converged to the same near minimax design, and the sequential approach turned out to be more efficient than the direct approach. The Newton and quasi-Newton algorithms required 21.1 and 40.7 s of CPU time, respectively, on a PC with a Pentium 4, 3.2 GHz CPU. The amplitude response, passband error, and group delay characteristic of the filter obtained are plotted in Fig. 9.99.9a, b, and c, respectively. We note that an equiripple amplitude response was achieved in both the passband and stopband. The passband group delay varies between 22.9 and 23.1 but it is not equiripple. This is because the minimax optimization was carried out for the *complex-valued* frequency response $H_d(\omega)$, not the phase-response alone (see Eq. (9.459.45)). ■

Example 9.5 Using the above direct and sequential approaches first with a Newton and then with a quasi-Newton algorithm, design a bandpass FIR filter of order $N = 160$ that would have approximately constant passband group delay of 65 s. Assume idealized passband and stopband gains of 1 and 0, respectively; normal-

ized sampling frequency= 2π; passband edges $\omega_{p1} = 0.4\pi$ and $\omega_{p2} = 0.6\pi$; stopband edges $\omega_{a1} = 0.375\pi$ and $\omega_{a2} = 0.625\pi$; $W(\omega) = 1$ in the passband and $W(\omega) = 50$ in the stopbands, and $W(\omega) = 0$ elsewhere.

Solution The required design was carried out using the direct approach with $p = 128$ and the sequential approach with $p = 2, 4, 8, \ldots, 128$ by minimizing the objective function in Eq. (9.489.48) with the Newton and quasi-Newton algorithms described in Example 9.4.

A bandpass FIR filter that would satisfy the required specifications can be obtained by assuming a complex-valued idealized frequency response of the form

$$H_d(\omega) = \begin{cases} e^{-j65\omega} & \text{for } \omega \in [\omega_{p1}, \omega_{p2}] \\ 0 & \text{for } \omega \in [0, \omega_{a1}] \bigcup [\omega_{a2}, \omega_s/2] \end{cases}$$
$$= \begin{cases} e^{-j65\omega} & \text{for } \omega \in [0.4\pi, 0.6\pi] \\ 0 & \text{for } \omega \in [0, 0.375\pi] \bigcup [0.625\pi, \pi] \end{cases}$$

(see Sec. B.9.2). The objective function in Eq. (9.489.48) was expressed as

$$f(\mathbf{h}) = \left[\int_0^{0.375\pi} 50e_2^p(\omega)d\omega\right]^{1/2p} + \left[\int_{0.4\pi}^{0.6\pi} e_2^p(\omega)d\omega\right]^{1/2p}$$
$$+ \left[\int_{0.625\pi}^{\pi} 50e_2^p(\omega)d\omega\right]^{1/2p}$$

and the integrals at the right-hand side were evaluated using the numerical method in the solution of Example 9.4 with $K = 382, 236, 382$ respectively.
The integrals in Eqs. (9.49a9.49a) and (9.49c9.49c) were similarly evaluated in order to obtain the gradient and Hessian of the problem.

As in Example 9.4, the sequential approach was more efficient. The Newton and quasi-Newton algorithms required 173.5 and 201.8 s, respectively, on a Pentium 4 PC.

The amplitude response, passband error, and group delay characteristic are plotted in Fig. 9.109.10a, b, and c, respectively. We note that an equiripple amplitude response has been achieved in both the passband and stopband. ■

We conclude this chapter with some remarks on the numerical results of Examples 9.2, 9.4 and 9.5. Quasi-Newton algorithms, in particular algorithms using an inexact line-search along with the BFGS updating formula (e.g., Algorithm 7.3), are known to be very robust and efficient relative to other gradient-based algorithms [7][**?**]–[8][**?**]. However, the basic Newton algorithm used for these problems, namely, Algorithm 9.1, turned out to be more efficient than the quasi-Newton algorithm. This is largely due to certain unique features of the problems considered, which favor the basic Newton algorithm. The problem in Example 9.2 is a simple problem with only three independent variables and an well defined gradient and Hessian that can be easily computed through closed-form formulas. Furthermore, the inversion of the Hessian is almost a trivial task. The problems in Examples 9.4 and 9.5 are significantly more complex than that in Example 9.2; however, their

gradients and Hessians are fairly easy to compute accurately and efficiently through closed-form formulas as in Example 9.2. In addition, these problems are convex with unique global minimums that are easy to locate. On the other hand, a large number of variables in the problem tends to be an impediment in quasi-Newton algorithms because, as was shown in Chap. 7, these algorithms would, in theory, require n iterations in an n-variable problem to compute the inverse-Hessian in a well defined convex quadratic problem (see proof of Theorem 7.3), more in noncovex nonquadratic problems. However, in multimodal[5] highly nonlinear problems with a moderate number of independent variables, quasi-Newton algorithms are usually the most efficient.

9.5 Source Localization

Source localization refers to a class of problems where the distances between a radiating source located at $\mathbf{x}$ and several sensors at known locations $\{\mathbf{s}_i\}$, namely,

$$d_i = \|\mathbf{x} - \mathbf{s}_i\| \qquad \text{for } i = 1, 2, \ldots, m \tag{9.52}$$

are available and the problem is to determine $\mathbf{x}$ based on measurements $\{d_i,\ i = 1, 2, \ldots, m\}$. In the literature, these d_i's are called range measurements. A variant of the above problem is one where an additional sensor, $\mathbf{s}_0$, is placed at the origin, namely $\mathbf{s}_0 = \mathbf{0}$, and the so-called range-difference measurements

$$\hat{d}_i = \|\mathbf{x} - \mathbf{s}_i\| - \|\mathbf{x}\| \qquad \text{for } i = 1, 2, \ldots, m \tag{9.53}$$

are available. The problem is to determine $\mathbf{x}$ based on measurements $\{\hat{d}_i,\ i = 1, 2, \ldots, m\}$.

Source localization problems occur in wireless communications, navigation, geophysics, and many other areas [9]–[12]. In this section, we formulate them as unconstrained optimization problems and apply several algorithms learned from previous chapters to solve them.

9.5.1 Source localization based on range measurements

For the sake of simplicity we consider source localizations problems on a plane, namely $\mathbf{s}_i \in R^2$ and $\mathbf{x} \in R^2$. Realistically, measurement noise always exist, hence the range measurements are not precisely d_i as given in Eq. (9.52), but are contaminated by noise and given by

$$r_i = \|\mathbf{x} - \mathbf{s}_i\| + n_i \qquad \text{for } i = 1, 2, \ldots, m \tag{9.54}$$

[5] Problems with multiple minima.

where n_i denotes the unknown noise for the ith measurement. The source localization problem is to estimate the location vector $\mathbf{x}$ given the locations of the m sensors $\{\mathbf{s}_i,\ i = 1, 2, \ldots, m\}$ and range measurements $\{r_i,\ i = 1, 2, \ldots, m\}$. A natural way to address the problem is to formulate the localization problem as the minimization problem

$$\underset{\mathbf{x}}{\text{minimize}} \quad f(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{m}(\|\mathbf{x} - \mathbf{s}_i\| - r_i)^2 \tag{9.55}$$

For the problem at hand, it is reasonable to assume that $\mathbf{x} \neq \mathbf{s}_i$ for $1 \leq i \leq m$ because the radiating source is typically away from the sensors. In this case the objective function in Eq. (9.55) is differentiable, and its gradient and Hessian are given by

$$\mathbf{g}(\mathbf{x}) = \sum_{i=1}^{m}\left(1 - \frac{r_i}{\|\mathbf{x} - \mathbf{s}_i\|}\right)(\mathbf{x} - \mathbf{s}_i) \tag{9.56}$$

and

$$\mathbf{H}(\mathbf{x}) = \sum_{i=1}^{m}\frac{r_i}{\|\mathbf{x} - \mathbf{s}_i\|^3}(\mathbf{x} - \mathbf{s}_i)(\mathbf{x} - \mathbf{s}_i)^T + \tau\mathbf{I} \tag{9.57a}$$

respectively, where $\mathbf{I}$ denotes the 2×2 identity matrix and

$$\tau = m - \sum_{i=1}^{m}\frac{r_i}{\|\mathbf{x} - \mathbf{s}_i\|} \tag{9.57b}$$

From Eq. (9.57), it is observed that $f(\mathbf{x})$ becomes nonconvex when τ is negative with a large magnitude. As an example, we consider a system [12] with five sensors located at

$$\mathbf{s}_1 = \begin{bmatrix}6\\4\end{bmatrix},\ \mathbf{s}_2 = \begin{bmatrix}0\\-10\end{bmatrix},\ \mathbf{s}_3 = \begin{bmatrix}5\\-3\end{bmatrix},\ \mathbf{s}_4 = \begin{bmatrix}1\\-4\end{bmatrix},\ \mathbf{s}_5 = \begin{bmatrix}3\\-3\end{bmatrix} \tag{9.58}$$

The source is located at $\mathbf{x}_s = [-2\ \ 3]^T$, and the exact and noisy range measurements are given by

$$\{d_i\} = \{8.0622,\ 13.1529,\ 9.2195,\ 7.6157,\ 7.8102\} \tag{9.59a}$$

and

$$\{r_i\} = \{8.0051,\ 13.0112,\ 9.1138,\ 7.7924,\ 8.0210\} \tag{9.59b}$$

respectively. A contour plot of $f(\mathbf{x})$ over the region

$$\mathcal{R} = \{x : -7 \leq x_1 \leq 15,\ -15 \leq x_2 \leq 7\}$$

is depicted in Fig. 9.11. It is observed from the figure that there are two minimizers, three maximizers, and four saddle points over the region.

The global and local minimizers were found to be (see Sec. 9.5.2)

$$\mathbf{x}^* = \begin{bmatrix} -1.990678 \\ 3.047388 \end{bmatrix} \quad \text{and} \quad \mathbf{x}_* = \begin{bmatrix} 11.115212 \\ -2.678506 \end{bmatrix} \tag{9.60a}$$

respectively, at which the objective function assume the values

$$f(x^*) = 0.1048 \quad \text{and} \quad f(\mathbf{x}_*) = 15.0083 \tag{9.60b}$$

As expected, the global minimizer of $f(\mathbf{x})$ is near the exact location of the source $\mathbf{x}_s = [-2 \;\; 3]^T$, but is unlikely at $\mathbf{x}_s$ exactly because $f(\mathbf{x})$ is defined using the noisy measurements r_i's rather than d_k's, see Eq. (9.55).

Since the problem only involves two variables and the Hessian is available in closed form, the Newton algorithm is well suited for solving the problem. To apply the Newton algorithm, the Hessian $\mathbf{H}(\mathbf{x})$ in Eq. (9.57) needs to be modified so as to ensure its positive definiteness. To this end, we write the Hessian in Eq. (9.57a) as

$$\mathbf{H}(\mathbf{x}) = \mathbf{H}_1(\mathbf{x}) + \tau \mathbf{I} \tag{9.61a}$$

where

$$\mathbf{H}(\mathbf{x}) = \sum_{i=1}^{m} \frac{r_i}{\|\mathbf{x} - \mathbf{s}_i\|^3} (\mathbf{x} - \mathbf{s}_i)(\mathbf{x} - \mathbf{s}_i)^T \tag{9.61b}$$

Let an eigen-decomposition of $\mathbf{H}_1(\mathbf{x})$ be given by

$$\mathbf{H}_1(\mathbf{x}) = \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^T$$

where $\mathbf{U}$ is orthogonal and $\boldsymbol{\Lambda} = \text{diag}\,\{\lambda_1, \; \lambda_2\}$. Therefor, the Hessian $\mathbf{H}(\mathbf{x})$ can be expressed as

$$\mathbf{H}(\mathbf{x}) = \mathbf{U} \hat{\boldsymbol{\Lambda}} \mathbf{U}^T$$

where $\hat{\boldsymbol{\Lambda}} = \text{diag}\,\{\lambda_1 + \tau, \; \lambda_2 + \tau\}$. To ensure a descent Newton step, the Hessian is modified to

$$\tilde{\mathbf{H}}(\mathbf{x}) = \mathbf{U} \tilde{\boldsymbol{\Lambda}} \mathbf{U}^T \tag{9.62a}$$

where

$$\tilde{\boldsymbol{\Lambda}} = \text{diag}\,\{\max(\lambda_1 + \tau, \; 0.05) \;\; \max(\lambda_2 + \tau, \; 0.05)\} \tag{9.62b}$$

so that the matrix $\tilde{\mathbf{H}}(\mathbf{x})$ in Eq. (9.62a) is guaranteed positive definite. The search direction in the kth iteration of the modified Newton algorithm is now given by

$$\mathbf{d}_k = -\mathbf{U} \tilde{\boldsymbol{\Lambda}}^{-1} \mathbf{U}^T g(\mathbf{x}_k)$$

where $g(\mathbf{x})$ is given by Eq. (9.56).

9.5.2 An illustrative example

Consider the five-sensor system described in Sec. 9.5.1, where the sensors' locations $\{\mathbf{s}_i,\ i = 1, 2, \ldots, 5\}$ and noisy range measurements $\{r_i,\ i = 1, 2, \ldots, 5\}$ are given by Eqs. (9.58) and (9.59b), respectively. The Newton algorithm with modified Hessian described in Sec. 9.5.1 and the quasi-Newton algorithm with the BFGS updating formula (see Sec. 7.6) which is well known for its good performance were applied to the source localization problem at hand.

The performance of these algorithms was evaluated by applying them with each of a total of 98×98 initial points that are uniformly placed over the region $R = \{\mathbf{x} : -8 \le x_1 \le 8,\ -8 \le x_2 \le 8\}$. From Fig. 9.11, it is observed that the region includes the global minimizer of the objective function $f(\mathbf{x})$ in Eq. (9.55) as well as several maximizers and saddle points.

Fig. 9.12 shows a profile of the values of the minimized objective function obtained using the modified Newton algorithm versus the initial points $\mathbf{x}_0$ from region $\mathcal{R}$. Among the 9604 initial points, there are 7476 $\mathbf{x}_0$'s for which the modified Newton algorithm converges to the global minimizer $\mathbf{x}^*$ given in Eq. (9.60a). For each of the remaining 2128 initial points, the algorithm converges to the local minimizer $\mathbf{x}_*$. The profile of the values of the minimized objective function obtained by the BFGS algorithm versus the initial points $\mathbf{x}_0$ in $\mathcal{R}$ is shown in Fig. 9.13. There are 7453 initial point in region $\mathcal{R}$ where the BFGS algorithm converges to the global minimizer $\mathbf{x}^*$ while at each of the remaining 2151 initial points the algorithm converges to $\mathbf{x}_*$.

If we normalize the average CPU time required by the modified Newton algorithm per run to unity, the average CPU time required by the BFGS algorithm per run was found to be 1.2545. Based on these, we conclude that both algorithms were able to converge to the global minimizer $\mathbf{x}^*$ from initial points over a sizable vicinity of $\mathbf{x}^*$ with comparable computational complexity.

9.5.3 Source localization based on range-difference measurements

Another type of source localization problem that has attracted considerable attention [9][11][12] is that of localizing a radiating source using range-difference measurements. In practice, range-difference measurements may be obtained from the time differences of arrival measured by an array of passive sensors.

Mathematically the problem assumes that there exists an additional sensor, $\mathbf{s}_0$, at the origin, namely $\mathbf{s}_0 = \mathbf{0}$. The range-difference d_i is defined as the difference between the distance from sensor $\mathbf{s}_i$ to source $\mathbf{x}$ and the distance from sensor $\mathbf{s}_0$ to source $\mathbf{x}$, namely

$$d_i = \|\mathbf{x} - \mathbf{s}_i\| - \|\mathbf{x}\| \qquad \text{for } i = 1, 2, \ldots, m \tag{9.63}$$

The source localization problem is to estimate the components of $\mathbf{x}$ given the locations of the $m + 1$ sensors $\{\mathbf{s}_i,\ i = 0, 1, \ldots, m\}$ and noise-contaminated range-

difference measurements $\{r_i,\ i = 1, 2, \ldots, m\}$ where

$$r_i = d_i + \varepsilon_i \qquad \text{for } i = 1, 2, \ldots, m \tag{9.64}$$

and ε_i is the noise for the ith measurement. Obviously, the localization problem can be formulated as the nonlinear least-squares problem

$$\text{minimize} \quad f(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{m}(\|\mathbf{x} - \mathbf{s}_i\| - \|\mathbf{x}\| - r_i)^2 \tag{9.65}$$

Assume that $\mathbf{x} \neq \mathbf{s}_i$ for $i = 0, 1, \ldots, m$, thus $f(\mathbf{x})$ is differentiable. Its gradient and Hessian are computed as

$$\mathbf{g}(\mathbf{x}) = \sum_{i=1}^{m} c_i(\mathbf{q}_i - \tilde{\mathbf{x}}) \tag{9.66}$$

and

$$\mathbf{H}(\mathbf{x}) = \sum_{i=1}^{m}[(\mathbf{q}_i - \tilde{\mathbf{x}})(\mathbf{q}_i - \tilde{\mathbf{x}})^T + c_i(\mathbf{Q}_{1i} - \mathbf{Q}_2)] \tag{9.67}$$

respectively, where

$$\begin{aligned}
\mathbf{c}_i &= \|\mathbf{x} - \mathbf{s}_i\| - \|\mathbf{x}\| - r_i \\
\mathbf{q}_i &= \frac{\mathbf{x} - \mathbf{s}_i}{\|\mathbf{x} - \mathbf{s}_i\|} \\
\tilde{\mathbf{x}} &= \frac{\mathbf{x}}{\|\mathbf{x}\|} \\
\mathbf{Q}_{1i} &= \frac{1}{\|\mathbf{x} - \mathbf{s}_i\|}(\mathbf{I} - \mathbf{q}_i\mathbf{q}_i^T) \\
\mathbf{Q}_2 &= \frac{1}{\|\mathbf{x}\|}(\mathbf{I} - \tilde{\mathbf{x}}\tilde{\mathbf{x}}^T)
\end{aligned}$$

Like the modified Newton algorithm based on range measurements, the Hessian $\mathbf{H}(\mathbf{x})$ in Eq. (9.67) may or may not be positive definite. Let the eigen-decomposition of $\mathbf{H}(\mathbf{x})$ be given by

$$\mathbf{H}(\mathbf{x}) = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T \tag{9.68}$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix with the eigenvalue values λ_1 and λ_2 of $\mathbf{H}(\mathbf{x})$ on its diagonal. The Hessian is modified to

$$\tilde{\mathbf{H}}(\mathbf{x}) = \mathbf{U}\tilde{\boldsymbol{\Lambda}}\mathbf{U}^T \tag{9.69}$$

where

$$\tilde{\boldsymbol{\Lambda}} = \text{diag}\,\{\max(\lambda_1,\ 0.05),\ \max(\lambda_2,\ 0.05)\}$$

so that $\tilde{\mathbf{H}}(\mathbf{x})$ is guaranteed to be positive definite. Since $\mathbf{H}(\mathbf{x})$ has a very small size, computing $\tilde{\mathbf{H}}(\mathbf{x})$ in Eq. (9.69) is rather economical, hence the modified Newton algorithm using $\tilde{\mathbf{H}}(\mathbf{x})$ in Eq. (9.69) is a good choice for solving the problem in

Eq. (9.65). This is especially the case when the number of sensors, m, is small. If m is large, the size of Hessian $\mathbf{H}(\mathbf{x})$ remains to be 2 by 2. However, as can be seen from Eq. (9.67), the evaluation of $\mathbf{H}(\mathbf{x})$ becomes expensive for a large m. In this case, a quasi-Newton algorithm becomes more attractive as it only requires the gradient information which, as can be observed from Eq. (9.66), can be computed efficiently.

9.5.4 An illustrative example

Consider a six-sensor system [12] where sensor $\mathbf{s}_0$ was placed at the origin and the other five sensors were located at

$$\mathbf{s}_1 = \begin{bmatrix} -5 \\ -13 \end{bmatrix},\ \mathbf{s}_2 = \begin{bmatrix} -12 \\ 1 \end{bmatrix},\ \mathbf{s}_3 = \begin{bmatrix} -1 \\ -5 \end{bmatrix},\ \mathbf{s}_4 = \begin{bmatrix} -9 \\ -12 \end{bmatrix},\ \mathbf{s}_5 = \begin{bmatrix} -3 \\ -12 \end{bmatrix}$$

The radiating source was located at $\mathbf{x}_s = [-5 \ \ 11]^T$. The problem is to estimate $\mathbf{x}_s$ based on noisy range-difference measurements which are defined by Eqs. (9.63) and (9.64):

$$\{r_i\} = \{11.8829,\ 0.1803,\ 4.6399,\ 11.2402,\ 10.8183\}$$

by solving the unconstrained problem in Eq. (9.65).

The modified Newton and BFGS algorithms were applied to the source localization problem with a total of 98×98 initial points that are uniformly placed over the region $\mathcal{R} = \{x : -13 \le x_1 \le 3,\ 0 \le x_2 \le 16\}$. The modified Newton algorithm converged from every initial point to the minimizer

$$\mathbf{x}^* = \begin{bmatrix} -4.988478 \\ 10.637469 \end{bmatrix}$$

which is very close to the true location of the radiating source, $\mathbf{x}_s$. The BFGS algorithm converged to the same solution $\mathbf{x}^*$ from 9592 initial points while it failed to work with the other 12 initial points, most of which were in the area $\{\mathbf{x} : 0.43 \le x_1 \le 0.53,\ 1.7 \le x_2 \le 2.3\}$. If we normalize the average CPU time required by the modified Newton algorithm per run to unity, the average CPU time required by the BFGS algorithm was found to be 0.8640. We see that both algorithms were able to converge to the global solution $\mathbf{x}^*$ from a sizable vicinity of $\mathbf{x}^*$ with comparable computational complexity.

References

1. M. Werman and D. Weinshall, "Similarity and affine invariant distances between 2D point sets," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, pp. 810–814, August 1995.
2. J. J. Craig, *Introduction to Robotics*, 2nd ed., Addison-Wesley, 1989.

3. A. Antoniou, *Digital Signal Processing: Signals, Systems, and Filters*, McGraw-Hill, New York, 2005.
4. T. W. Parks and J. H. McClellan, "Chebyshev approximation for nonrecursive digital filters with linear phase," *IEEE Trans. Circuit Theory*, vol. 19, pp. 189-194, 1972.
5. T. W. Parks and C. S. Burrus, *Digital Filter Design*, Wiley, New York, 1987.
6. G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, 1989.
7. R. Fletcher, *Practical Methods of Optimization*, vol. 1, Wiley, New York, 1980.
8. R. Fletcher, *Practical Methods of Optimization*, 2nd ed., Wiley, New York, 1987.
9. J. O. Smith and J. S. Abel, "Closed-form least-squares source location estimation from range-difference measurements," *IEEE Trans. Signal Processing*, vol. 12, pp. 1661–1669, Dec. 1987.
10. K. W. Cheung, W. K. Ma, and H. C. So, "Accurate approximation algorithm for TOA-based maximum likelihood mobile location using semidefinite programming," *Proc. ICASSP*, vol. 2, pp. 145–148, 2004.
11. P. Stoica and J. Li, "Source localization from range-difference measurements," *IEEE Signal Processing Magazine*, vol. 23, pp. 63–65, 69, Nov. 2006.
12. A. Beck, P. Stoica, and J. Li, "Exact and approximate solutions of source localization problems," *IEEE Trans. Signal Processing*, vol. 56, no. 5, pp. 1770–1777, May 2008.

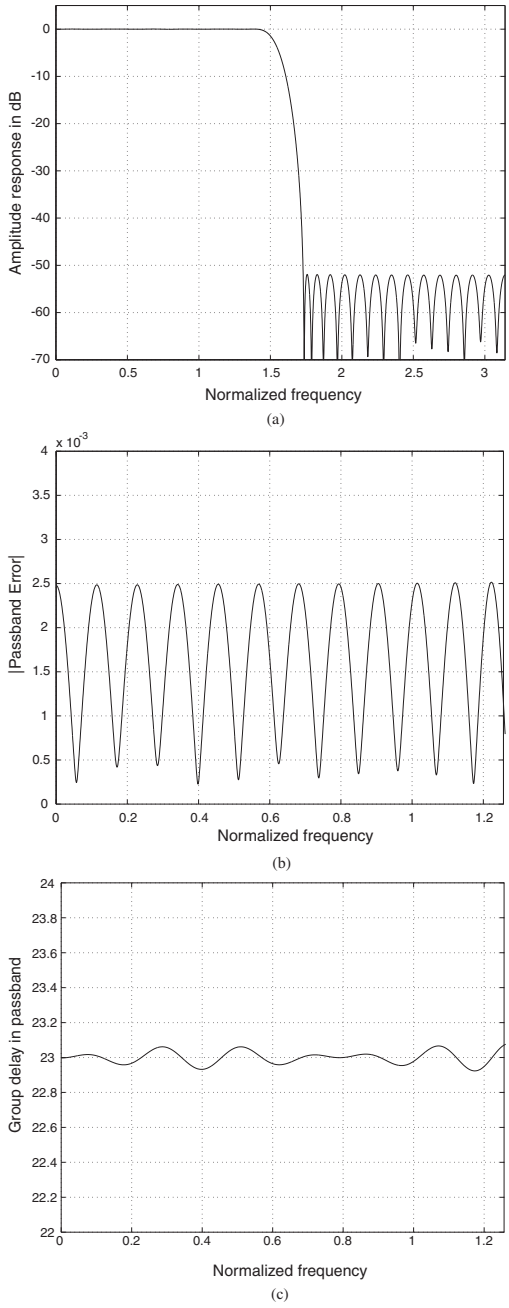

Fig. 9.9: Minimax design of a lowpass filter with low passband group delay for Example 9.4: (a) Frequency response, (b) magnitude of the passband error, and (c) passband group delay.

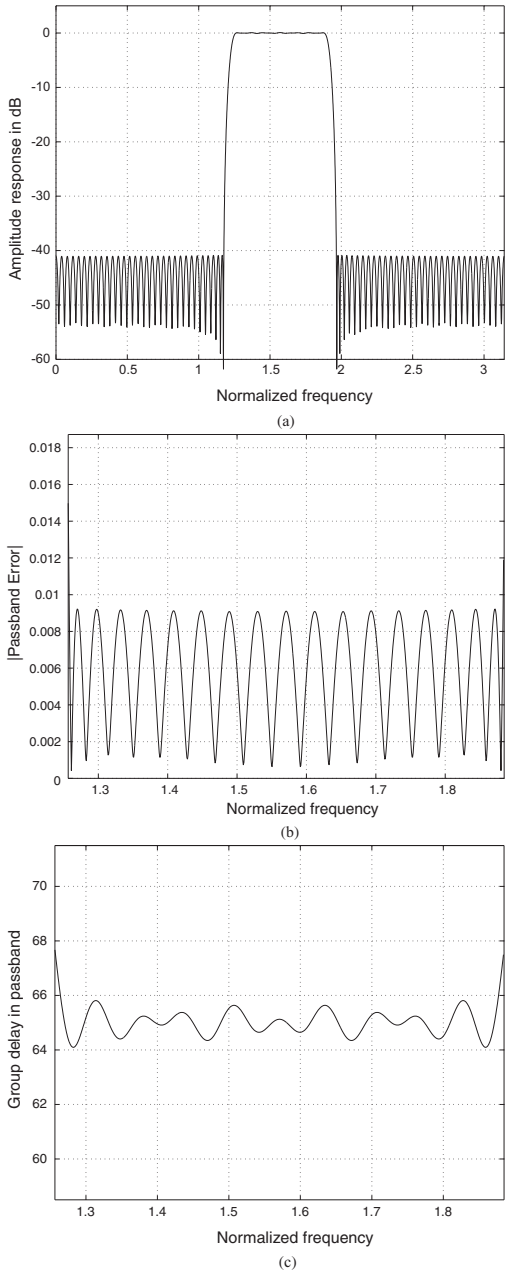

Fig. 9.10: Minimax design of a bandpass filter with low passband group delay for Example 9.5: (a) Frequency response, (b) magnitude of passband error, (c) passband group delay.

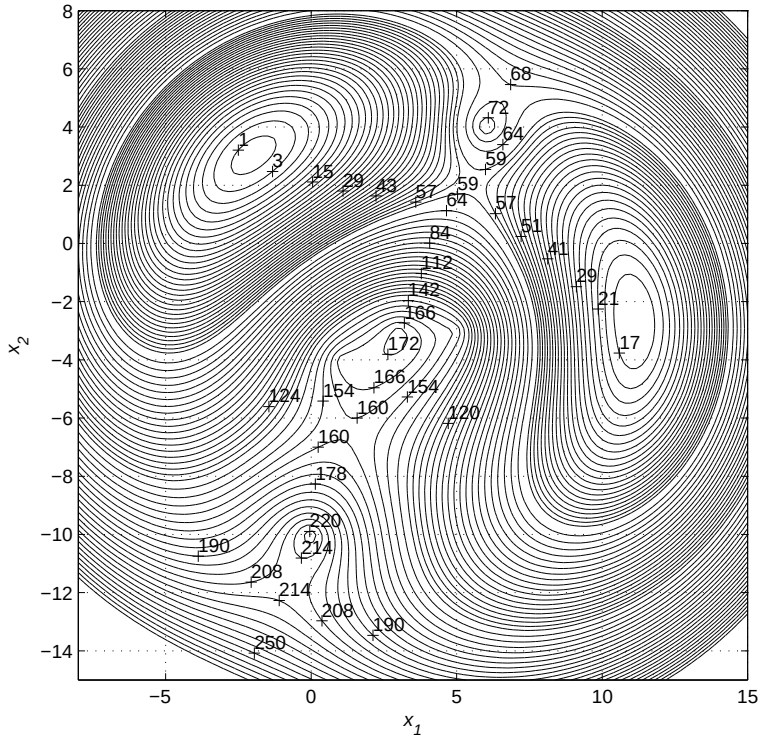

Fig. 9.11: Contours of $f(\mathbf{x})$ over $\mathcal{R}$.

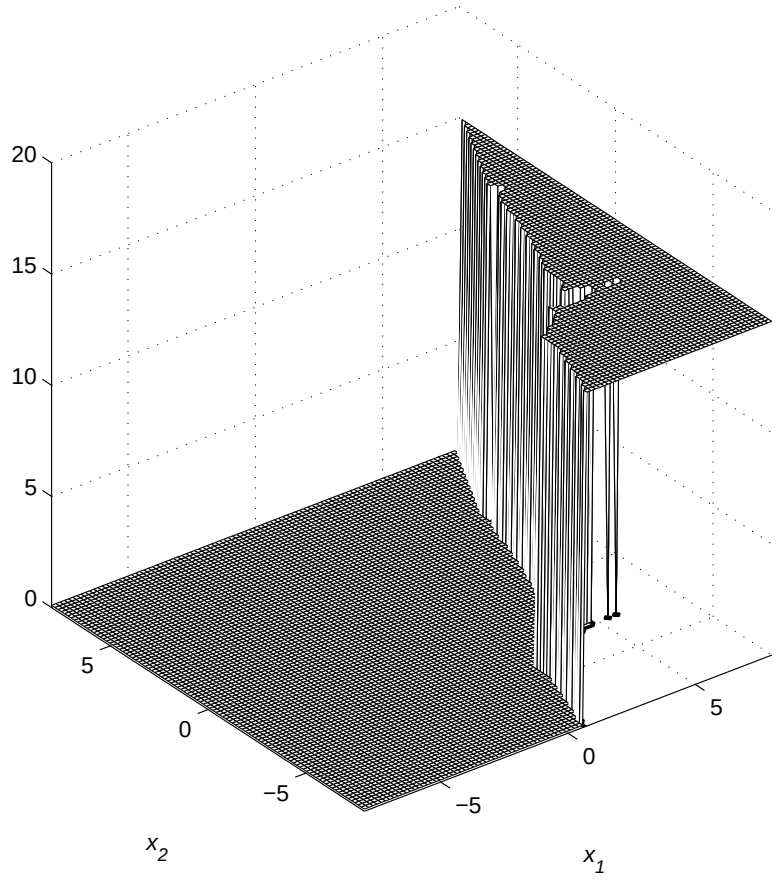

Fig. 9.12: Values of the minimized objective function obtained using the modified Newton algorithm versus the initial points $\mathbf{x}_0$ over the region $\mathcal{R}$.

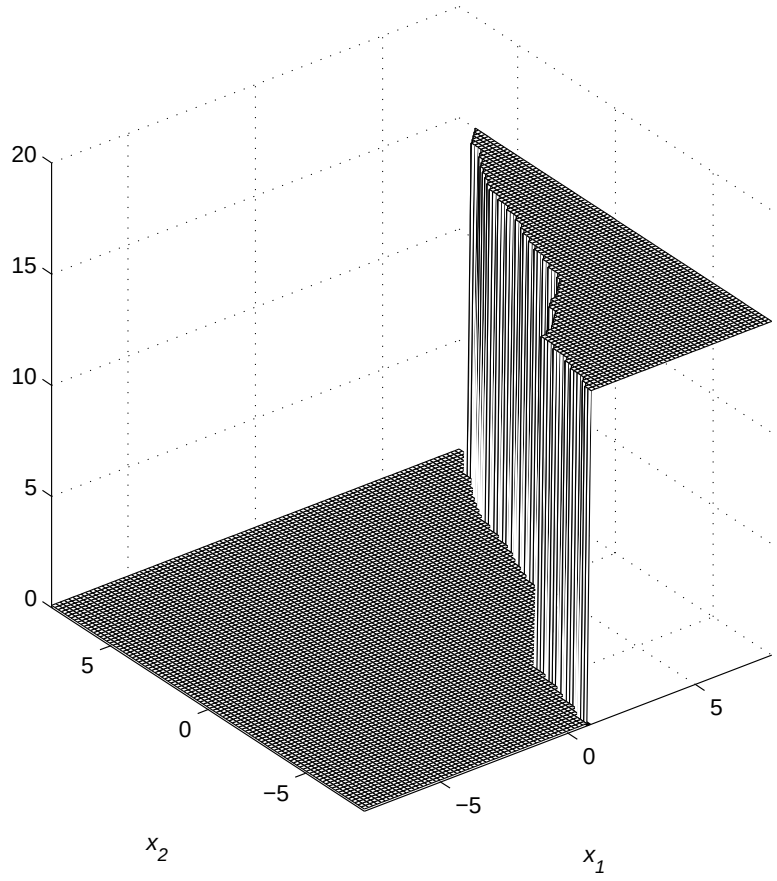

Fig. 9.13: Values of the minimized objective function obtained using the BFGS algorithm versus the initial points $\mathbf{x}_0$ over the region $\mathcal{R}$.

References

Problems

9.1 (*a*) Verify Eqs. (9.6a9.6a)–(9.6d9.6d)).

(*b*) Show that matrix $\mathbf{H}$ in Eq. (9.6b9.6b) is positive definite.

(*c*) Show that the inverse matrix $\mathbf{H}^{-1}$ in Eq. (9.79.7) can be evaluated as

$$\mathbf{H}^{-1} = \begin{bmatrix} \gamma_4 \mathbf{I}_2 & -\frac{\gamma_4}{n}\sum_{i=1}^{n}\mathbf{R}_i^T \\ -\frac{\gamma_4}{n}\sum_{i=1}^{n}\mathbf{R}_i & \frac{1}{n}\left(1+\frac{\gamma_3\gamma_4}{n}\right)\mathbf{I}_2 \end{bmatrix}$$

where

$$\gamma_1 = \sum_{i=1}^{n} p_{i1},\ \gamma_2 = \sum_{i=1}^{n} p_{i2},\ \gamma_3 = \gamma_1^2 + \gamma_2^2,\ \gamma_4 = \left(\|\mathbf{P}\|_F^2 - \frac{\gamma_3}{n}\right)^{-1}$$

9.2 The dissimilarity measure $e(\tilde{\mathcal{P}}, \mathcal{Q})$ defined in Eq. (9.59.5) is not symmetric, i.e., in general $e(\tilde{\mathcal{P}}, \mathcal{Q}) \neq e(\mathcal{Q}, \tilde{\mathcal{P}})$, which is obviously undesirable.

(*a*) Obtain a dissimilarity measure for two point patterns that is symmetric.

(*b*) Solve the minimization problem associated with the new dissimilarity measure.

9.3 (*a*) Verify Eqs. (9.9a9.9a)–(9.9c9.9c).

(*b*) Prove that the objective function given in Eq. (9.89.8) is globally convex. Hint: Show that for any $\mathbf{y} \in R^4$, $\mathbf{y}^T \nabla^2 e_{2p}(\mathbf{x})\mathbf{y} \geq 0$.

9.4 Derive formulas for the evaluation of $\nabla^2 f_k(\mathbf{x})$ for $k = 1,\ 2$, and 3 for the set of functions $f_k(\mathbf{x})$ given by Eq. (9.16) Eqs. (9.16a)–(9.16d).

9.5 Show that for a nontrivial weighting function $W(\omega) \geq 0$, the matrix $\mathbf{Q}$ given by Eq. (9.309.30) is positive definite.

9.6 Derive the expressions of $\mathbf{Q}_l$ and $\mathbf{b}_l$ given in Eqs. (9.41), (9.42), and (9.43) Eqs. (9.41a)–(9.43c)

9.7 Write a MATLAB program to implement the unconstrained optimization algorithm for the weighted least-squares design of linear-phase lowpass FIR digital filters studied in Sec. 9.4.1.2.

9.8 Develop an unconstrained optimization algorithm for the weighted least-squares design of linear-phase highpass digital filters.

9.9 Prove that the objective function given in Eq. (9.459.45) is globally convex. Hint: Show that for any $\mathbf{y} \in R^{N+1}$, $\mathbf{y}^T \nabla^2 f(\mathbf{h})\mathbf{y} \geq 0$.

9.10 Develop a method based on unconstrained optimization for the design of FIR filters with low passband group delay allowing coefficients with complex values.

9.11 Consider the double inverted pendulum control system described in Example 1.2, where $\alpha = 16$, $\beta = 8$, $T_0 = 0.8$, $\Delta t = 0.02$, and $K = 40$. The initial state is set to $\mathbf{x}(0) = [\pi/6\ 1\ \pi/6\ 1]^T$ and the constraints on the magnitude of control actions are $|u(i)| \leq m$ for $i = 0, 1, \ldots, K-1$ with $m = 112$.

(*a*) Use the singular-value decomposition technique (see Sec. A.9, especially Eqs. (A.43**??**) and (A.44**??**)) to eliminate the equality constraint $\mathbf{a}(\mathbf{u}) = \mathbf{0}$ in Eq. (1.9b**??**).

(*b*) Convert the constrained problem obtained from part (*a*) to an unconstrained problem of the augmented objective function

$$F_\tau(\mathbf{u}) = \mathbf{u}^T\mathbf{u} - \tau \sum_{i=0}^{K-1} \ln[m - u(i)] - \tau \sum_{i=0}^{K-1} \ln[m + u(i)]$$

where the barrier parameter τ is fixed to a positive value in each round of minimization, which is then reduced to a smaller value at a fixed rate in the next round of minimization.

Note that in each round of minimization, a line search step should be carefully executed where the step-size α is limited to a *finite* interval $[0, \bar{\alpha}]$ that is determined by the constraints $|u(i)| \leq m$ for $0 \leq i \leq K-1$.

9.12 (*a*) Verify the gradient given by Eq. (9.56).

(*b*) Verify the Hessian given by Eq. (9.57).

(*c*) Verify the gradient given by Eq. (9.66).

(*d*) Verify the Hessian given by Eq. (9.67).