

第1章 自适应滤波导论

1.1 引言

在这一节中，将对本书讨论的信号处理系统进行简要介绍。

在过去30年里，信号处理领域作出了巨大贡献。数字电路设计技术的进步，是促使人们对数字信号处理领域越来越感兴趣的关键技术原因。由于数字信号处理系统具有成本低、可靠性好、精度高、体积小及灵活性强等特点，导致数字信号处理系统的吸引力也越来越强。

数字信号处理系统的一个典型例子是滤波器。滤波是一种信号处理操作，其目的是通过对某个信号的处理，以便利用其中所包含的信息。换句话说，滤波器是一个器件，它将输入信号映射为另一个输出信号，从而便于提取出输入信号中所包含的期望信息。数字滤波器是处理以数字形式表示的离散时间信号的滤波器。对于时不变滤波器而言，其内部参数和结构都是固定的，而如果滤波器是线性的，则其输出信号是输入信号的线性函数。一旦预先给定了滤波器的技术规范和要求，则时不变线性滤波器的设计过程包括三个基本步骤，即利用有理传输函数对规范做近似，选取适合于算法的合理结构，以及选取算法的实现形式等。

如果固定的设计规范是未知的，或者采用时不变滤波器不能满足设计规范的要求，就需要采用自适应滤波器。严格地讲，自适应滤波器是一种非线性滤波器(因为其特征取决于输入信号)，因此不满足齐次性和叠加性条件。然而，如果在某个给定时刻使滤波器参数固定，则其输出信号是输入信号的线性函数，从这个意义上讲，除第11章所讨论的自适应滤波器以外，本书所讨论的大多数自适应滤波器都是线性的。

为了满足性能要求，自适应滤波器的参数需要不断变化，因此它是一种时变滤波器。从这个意义上讲，可以将自适应滤波器解释为一种以在线方式完成近似步骤的滤波器。在通常情况下，定义性能准则时往往需要参考信号，该参考信号通常隐含在固定滤波器设计的近似步骤中。这种说法会让人觉得，在设计固定(非自适应)滤波器过程中，需要知道输入和参考信号的完整特征才能设计出满足一定性能的最合适的滤波器。然而，在实际中很少遇到这种情形，因为实际环境的定义并非很明确。构成环境的信号有输入信号和参考信号，如果这两种信号都不太明确，采用的工程方法是首先对信号进行建模，然后再设计滤波器。但是这种方法复杂度很高，而且很难用在线方式实现。为了解决这个问题，通常采用自适应滤波器，该滤波器只利用环境中的可用信息，通过一个相当简单的算法对其参数进行在线更新。换句话说，自适应滤波器完成了一种数据驱动近似步骤。

本书的主题是自适应滤波，主要讨论对于一个参数(或者系数)能自适应调整的滤波器，如何根据预先确定的性能准则选择其结构和算法的问题。滤波器系数的更新是利用在某个给定时刻的可用信息来完成的。

随着数字甚大规模集成(VLSI, Very Large Scale Integration)技术的发展，使得自适应信号处理技术在许多应用领域中取得了广泛应用。因此，本书只考虑了自适应滤波器的离散时间实现方式。显然，我们假设从现实中获取的连续时间信号都可以通过合适的采样，即以大于其

最高频率两倍的速率进行采样,从而可以利用离散时间信号来对其进行表示。一般而言,当通过对连续时间信号采样来产生离散时间信号时,都假设满足奈奎斯特定理或者采样定理^[1-9]。

1.2 自适应信号处理

正如前面所讨论的,在设计系数固定的数字滤波器时要求很明确的设计规范。然而,在某些情况下设计规范是未知或者是时变的。其解决办法是采用具有自适应系数的数字滤波器,即自适应滤波器^[10-17]。

由于没有具体的设计规范和求,因此决定滤波器系数更新的自适应算法需要一些其他的信息,这些信息通常是以信号的形式给出的。该信号通常被称为期望信号或者参考信号,其选择取决于具体应用,并且往往具有一定的技巧性。

由于自适应滤波器是一种非线性系统,因此其特性分析比固定滤波器更加复杂。另一方面,由于自适应滤波器是一种“自我设计”的滤波器,因此从设计人员的观点来看,它比固定滤波器的设计更加简单。

在图 1.1 中,给出了自适应滤波的一般结构,其中 k 为迭代次数, $x(k)$ 表示输入信号, $y(k)$ 表示输出信号, $d(k)$ 表示期望信号。误差信号 $e(k)$ 可以根据 $d(k) - y(k)$ 计算得出。为了确定滤波器系数的适当更新方式,通常利用误差信号构造自适应算法所需的性能(或者目标)函数。目标函数的最小化意味着自适应滤波器的输出信号与期望信号在某种意义上实现了最佳匹配。

正如图 1.1 所示,一个自适应系统的完整规范由以下三个要素组成:

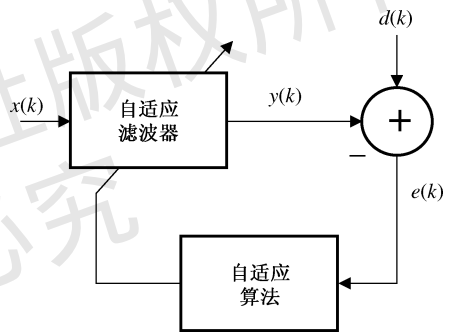


图 1.1 自适应滤波器的一般结构

1. **应用:** 应用类型是根据具体环境选取的输入信号和输出期望信号来确定的。在过去 30 年里,自适应技术取得成功应用的不同领域越来越多。比如回波消除、色散信道的均衡、系统辨识、信号增强、自适应波束形成、噪声消除,以及控制等许多领域^[14~20]。尽管本书主要目的不是对自适应滤波器的各种不同应用进行研究,然而也会根据需要对某些应用进行详细讨论。
2. **自适应滤波器结构:** 自适应滤波器可以采用许多不同结构来实现。结构的选取不仅会影响到处理的计算复杂度(即每次迭代的运算操作次数),还会对达到期望性能标准所需的迭代次数产生影响。根据冲激响应的不同形式,自适应数字滤波器的实现结构主要有两种,即有限冲激响应(FIR, Finite-duration Impulse Response)滤波器和无限冲激响应(IIR, Infinite-duration Impulse Response)滤波器。FIR 滤波器通常利用非递归结构实现,而 IIR 滤波器则利用递归结构实现。
 - **自适应 FIR 滤波器实现:** 应用最广泛的自适应 FIR 滤波器结构是横向滤波器,也被称为抽头延迟线,它利用无反馈的标准直接形式实现全零点传输函数。对于这种结构,输出信号 $y(k)$ 是滤波器系数的线性组合,它产生具有唯一最优解的二次均方误差函数($MSE = E[|e(k)|^2]$)。为了使计算复杂度、收敛速度和有限字长特性等性能

指标优于横向滤波器结构,也会采用其他自适应 FIR 结构,本书后面将会对其进行介绍。

- **自适应 IIR 滤波器实现:** 自适应 IIR 滤波器采用得最多的结构是标准直接形式结构^[5],因为它的实现和分析都很简单。然而,采用递归自适应滤波器会存在一些由其结构所决定的内在问题,比如极点的稳定性问题、收敛速度慢的问题等。为了克服直接形式结构带来的这些局限性问题,人们提出了不同的实现方式。在这些实现中,本书考虑了级联结构、格型结构和并行结构等,因为它们具有一些特殊的性质,这将在第 10 章专门进行讨论。

3. **算法:** 算法是为了使某个预先确定的准则达到最小化,而使滤波器系数自适应调整所采用的方法。算法是通过定义搜索方法(或者最小化算法)、目标函数和误差信号的特性来确定的。算法的选择会对整个自适应过程的一些关键特性,如次优解的存在性、有偏最优解和计算复杂度等产生直接影响。

1.3 自适应算法简介

自适应滤波器的基本目标,是通过某种方式对参数 $\theta(k)$ 进行调整的,使滤波器输出尽可能使包含参考信号的某个特定目标函数达到最小化。通常而言,自适应滤波器的目标函数 F 是其输入信号、参考信号和输出信号的一个函数,即 $F = F[x(k), d(k), y(k)]$ 。定义的目标函数必须满足如下两个特性:

- **非负性:** 即对于任意 $y(k)$ 、 $x(k)$ 和 $d(k)$, 都有 $F[x(k), d(k), y(k)] \geq 0$ 。
- **最优性:** 即 $F[x(k), d(k), d(k)] = 0$ 。

可以理解为在自适应滤波过程中,自适应算法试图使函数 F 达到最小化,从而 $y(k)$ 与 $d(k)$ 近似相等,使得 $\theta(k)$ 收敛到 θ_0 , 其中 θ_0 为导致目标函数最小化的最优系数构成的集合。

目标函数的另一种解释方式,是将其视为某个一般误差信号 $e(k)$ 的直接函数,而该误差信号又是信号 $x(k)$ 、 $y(k)$ 和 $d(k)$ 的某个函数,即 $F = F[e(k)] = F[e(x(k), y(k), d(k))]$ 。在这种观点下,我们认为一个自适应算法需要确定三个基本要素:即最小化算法、目标函数形式和误差信号。下面对其分别进行讨论。

1. **确定函数 F 的最小化算法:** 这个要素属于最优化理论研究的主题^[21, 22],而且它会从本质上影响自适应过程的收敛速度和计算复杂度。

在实际中,任何具有高阶参数模型的连续函数都可以在某个给定点 $\theta(k)$, 用截尾 Taylor 级数表示如下:

$$F[\theta(k) + \Delta\theta(k)] \approx F[\theta(k)] + \mathbf{g}_\theta^T \{F[\theta(k)]\} \Delta\theta(k) + \frac{1}{2} \Delta\theta^T(k) \mathbf{H}_\theta \{F[\theta(k)]\} \Delta\theta(k) \quad (1.1)$$

其中, $\mathbf{H}_\theta \{F[\theta(k)]\}$ 是目标函数的黑塞矩阵 (Hessian matrix), $\mathbf{g}_\theta \{F[\theta(k)]\}$ 为梯度向量,关于黑塞矩阵和梯度向量的详细内容在本书后面会逐步涉及。自适应滤波的目的,是通过参数集的迭代使得目标函数达到最小化,即

$$\theta(k+1) = \theta(k) + \Delta\theta(k) \quad (1.2)$$

其中,步长或者修正项 $\Delta\boldsymbol{\theta}(k)$ 旨在使目标函数 $F[\boldsymbol{\theta}(k)]$ 的二次近似最小化。牛顿方法要求 $F[\boldsymbol{\theta}(k)]$ 在任意点都存在一阶和二阶导数,并且其函数值也存在,因为在计算式(1.1)时需要这些信息。如果 $\mathbf{H}_\theta(\boldsymbol{\theta}(k))$ 是正定矩阵,则二次近似具有唯一的、明确的最小值点。其求解方法是,在 $k+1$ 时刻使二次函数相对于参数修正项的梯度值为零,从而得到

$$\mathbf{g}_\theta\{F[\boldsymbol{\theta}(k)]\} = -\mathbf{H}_\theta\{F[\boldsymbol{\theta}(k)]\}\Delta\boldsymbol{\theta}(k) \quad (1.3)$$

在自适应信号处理领域中,最常用的最优化方法有:

- 牛顿方法:该方法寻找目标函数二阶近似的最小值,其参数向量的迭代更新方程为

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) - \mu\mathbf{H}_\theta^{-1}\{F[e(k)]\}\mathbf{g}_\theta\{F[e(k)]\} \quad (1.4)$$

其中, μ 是控制算法步长的因子,它决定了参数向量变化的快慢。值得注意的是,修正项 $\Delta\boldsymbol{\theta}(k)$ 的方向是根据式(1.3)来选择的。 $F[e(k)]$ 的二阶导数矩阵,即 $\mathbf{H}_\theta\{F[e(k)]\}$ 是目标函数的黑塞矩阵,而且 $\mathbf{g}_\theta\{F[e(k)]\}$ 是目标函数相对于自适应滤波器系数的梯度。这里需要注意的是,误差 $e(k)$ 取决于参数向量 $\boldsymbol{\theta}(k)$ 。如果目标函数 $F[e(k)]$ 本身是二次函数,则在式(1.1)的模型中不需要近似,并且当 $\mu=1$ 时只需一步即可达到其全局最小值。对于非二次函数, μ 的取值应该减小。

- 准牛顿方法:这一类算法是上述方法的简化形式,因为它通过递推计算来估计黑塞矩阵的逆矩阵,使目标函数达到最小化,即

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) - \mu\mathbf{S}(k)\mathbf{g}_\theta\{F[e(k)]\} \quad (1.5)$$

其中, $\mathbf{S}(k)$ 是 $\mathbf{H}_\theta^{-1}\{F[e(k)]\}$ 的估计值,使得

$$\lim_{k \rightarrow \infty} \mathbf{S}(k) = \mathbf{H}_\theta^{-1}\{F[e(k)]\}$$

通常采用矩阵求逆引理(参见参考文献[23],或者后面的某些章节内容)来计算黑塞矩阵的逆矩阵。另外,还通常采用估计值来代替梯度向量,使得计算更加有效。

- 最陡下降方法:这一类方法沿着与目标函数梯度向量相反的方向,搜索目标函数的最小值点。因此,其更新方程为

$$\boldsymbol{\theta}(k+1) = \boldsymbol{\theta}(k) - \mu\mathbf{g}_\theta\{F[e(k)]\} \quad (1.6)$$

在本书以及许多公开文献中,最陡下降方法也经常被称为梯度方法。

通常而言,梯度方法更容易实现,但是另一方面,采用牛顿方法达到最小值点的邻域所需的迭代次数更少。在许多情况下,为了在梯度方法的计算效率和牛顿方法的快速收敛之间进行折中,可以采用准牛顿方法。然而,由于在估计黑塞矩阵的逆矩阵时采用了递推形式,因此准牛顿算法容易带来不稳定问题。在参考文献[21, 22]中,对广泛采用的最小化算法进行了详细讨论。

应该指出的是,对于任意一种最小化方法,收敛因子 μ 控制着整个自适应过程的稳定性、收敛速度,以及残留误差的某些特征。通常而言,合理选择该参数需要充分了解某个感兴趣的具体自适应问题的相关信息,因此,不存在一个普遍适用的选取方法。在实际中,计算机仿真起着重要作用,并且是解决这个问题的最常用方法。

2. 确定目标函数 $F[e(k)]$: 有许多方法可以定义出满足前面描述的最优性和非负性的目标函数。其定义会对梯度向量和黑塞矩阵的计算复杂度产生影响。如果利用算法的计算复杂度作为定义准则,则可以列出在推导自适应算法过程中,最广泛采用的一些目标函数形式为:

- 均方误差 (MSE): $F[e(k)] = E[|e(k)|^2]$
- 最小二乘 (LS): $F[e(k)] = \frac{1}{k+1} \sum_{i=0}^k |e(k-i)|^2$
- 加权最小二乘 (WLS): $F[e(k)] = \sum_{i=0}^k \lambda^i |e(k-i)|^2$, 其中 λ 是小于 1 的常数。
- 瞬时平方值 (ISV): $F[e(k)] = |e(k)|^2$

从严格意义上讲, 均方误差 (MSE) 只是理论值, 因为它要求对信号进行无数次测量。实际上, 这个理想的目标函数可以用上面给出的另外三个目标函数来近似。LS、WLS 和 ISV 函数的实现复杂度及收敛特性都不相同, 一般而言, ISV 更加容易实现, 但它存在噪声收敛特性, 因为它是一种非常简化的目标函数。LS 便于在平稳信号环境下采用, 而 WLS 则更加适用于慢变化信号环境下的应用。

3. 确定误差信号 $e(k)$: 误差信号的选取对于确定算法是很关键的, 因为它会影响到整个算法的多个特征, 包括计算复杂度、收敛速度、鲁棒性, 而且更重要的是对于 IIR 自适应滤波情形, 它会对是否出现有偏解和多个解产生影响。

上面介绍的最小化算法、目标函数和误差信号, 为我们解释、分析和研究自适应算法给出了一种简单的结构化方法。事实上, 几乎所有已知的自适应算法都能够用这种方法进行理解, 或者对这种结构化方法稍加变化。在本书的剩余章节中, 我们利用这种框架给出自适应算法的基本原理。可以发现, 最小化算法和目标函数会影响到自适应过程的收敛速度, 选择误差信号是确定一个自适应算法的重要步骤, 因为它会对整个收敛过程的多个方面带来直接影响。

1.4 应用

在本节中, 将讨论可能用到的一些输入信号和期望信号, 以及这些信号是如何与实际应用相联系的。自适应滤波的一些经典应用包括系统辨识、信道均衡、信号增强和预测等。

在系统辨识应用中, 期望信号是未知系统受某个宽带信号激励时产生的输出, 在大多数情况下, 输入是白噪声信号。该宽带信号同时也用于图 1.2 所示的自适应滤波器的输入。当输出 MSE 达到最小时, 自适应滤波器就代表了未知系统的模型。

在信道均衡应用中, 将受信道和环境噪声失真影响的原始发射信号作为自适应滤波器的输入信号, 而期望信号是原始信号的时延形式, 如图 1.3 所示。通常情况下, 输入信号的时延形式在接收端是可以得到的, 其形式是采用标准的训练信号。当 MSE 达到最小时, 表明自适应滤波器代表了信道的逆模型 (均衡器)。

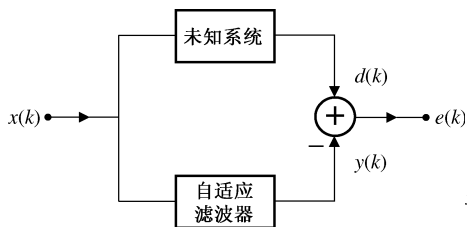


图 1.2 系统辨识

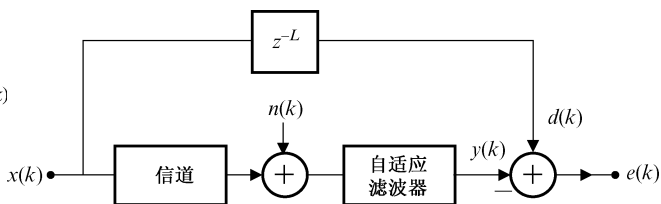


图 1.3 信道均衡

对于信号增强的情况, 信号 $x(k)$ 受到噪声 $n_1(k)$ 的污染, 而与噪声相关的信号 $n_2(k)$ 是可

以得到的(可测量的)。如果将 $n_2(k)$ 作为自适应滤波器的输入, 而受到噪声污染的信号作为期望信号, 则当滤波器收敛以后, 其输出误差就是信号的增强形式。图 1.4 给出了一种信号增强的典型结构。

最后, 对于预测情形, 期望信号是自适应滤波器输入信号的前向(有时可能是后向)形式, 如图 1.5 所示。当滤波器收敛以后, 则自适应滤波器就代表了输入信号的模型, 可以用来作为输入信号的预测模型。

在后续章节中, 将对这里给出的应用进一步详细讨论。

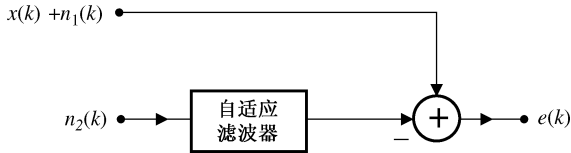


图 1.4 信号增强 [$n_1(k)$ 和 $n_2(k)$ 是彼此相关的噪声信号]

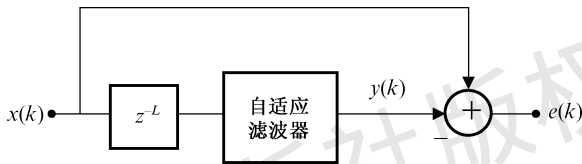


图 1.5 信号预测

例 1.1 在本章结束前, 我们给出一个简单例子, 说明自适应滤波器如何用于解决图 1.1 所示的一般性问题。这里选择图 1.4 所示的信号增强应用。

在这个例子中, 参考(或者期望)信号由受到有色噪声污染的离散时间三角波信号组成, 如图 1.6 所示。自适应滤波器的输入信号是与污染三角波信号的噪声信号相关的白噪声, 如图 1.7 所示。

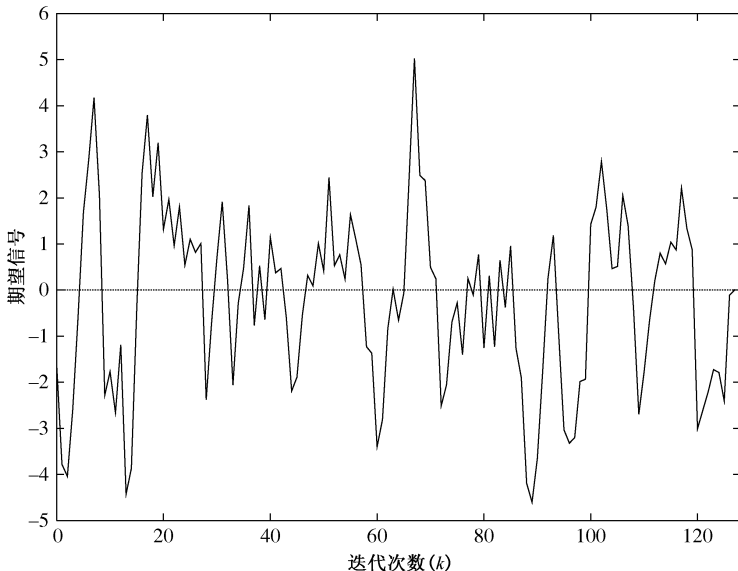


图 1.6 期望信号

为了使输出误差的平方值尽可能小,需要对自适应滤波器的系数进行调整。从图 1.8 中可以发现,当迭代次数增加时,误差信号与该图中的离散时间三角波信号(虚线)相似。

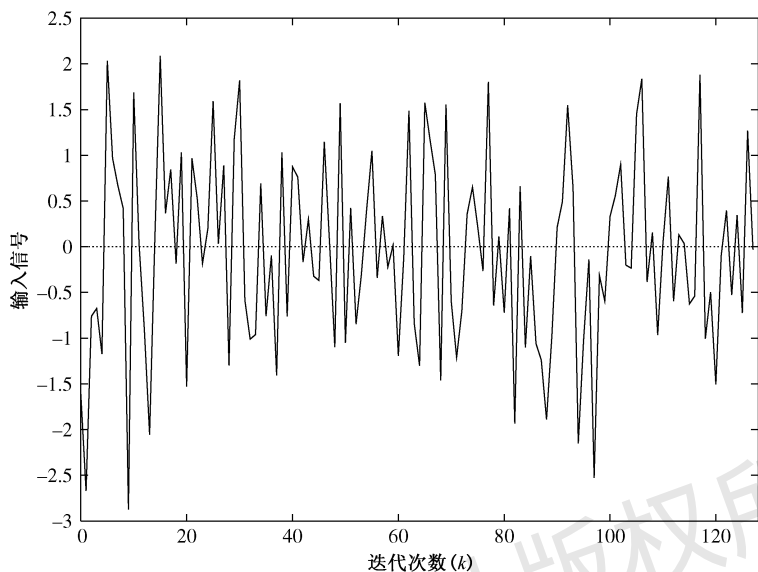


图 1.7 输入信号

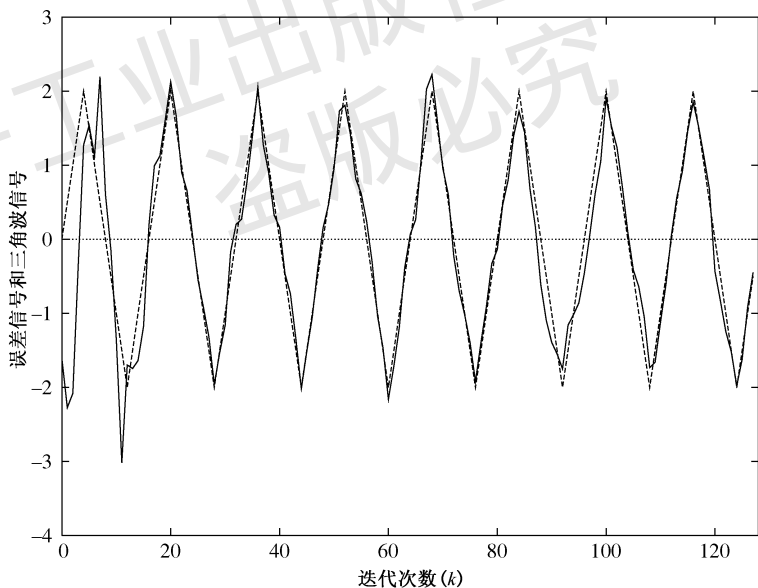


图 1.8 误差信号(实线)和三角波形信号(虚线)

参考文献

1. P.S.R. Diniz, E.A.B. da Silva, S.L. Netto, *Digital Signal Processing: System Analysis and Design*, 2nd edn. (Cambridge University Press, Cambridge, 2010)
2. A. Papoulis, *Signal Analysis* (McGraw Hill, New York, 1977)
3. A.V. Oppenheim, A.S. Willsky, S.H. Nawab, *Signals and Systems*, 2nd edn. (Prentice Hall, Englewood Cliffs, 1997)

4. A.V. Oppenheim, R.W. Schaffer, *Discrete-Time Signal Processing* (Prentice Hall, Englewood Cliffs, 1989)
5. A. Antoniou, *Digital Signal Processing: Signals, Systems, and Filters* (McGraw Hill, New York, 2005)
6. L.B. Jackson, *Digital Filters and Signal Processing*, 3rd edn. (Kluwer Academic, Norwell, 1996)
7. R.A. Roberts, C.T. Mullis, *Digital Signal Processing* (Addison-Wesley, Reading, 1987)
8. J.G. Proakis, D.G. Manolakis, *Digital Signal Processing*, 4th edn. (Prentice Hall, Englewood Cliffs, 2007)
9. T. Bose, *Digital Signal and Image Processing* (Wiley, New York, 2004)
10. M.L. Honig, D.G. Messerschmitt, *Adaptive Filters: Structures, Algorithms, and Applications* (Kluwer Academic, Boston, 1984)
11. S.T. Alexander, *Adaptive Signal Processing* (Springer, New York, 1986)
12. M. Bellanger, *Adaptive Digital Filters*, 2nd edn. (Marcel Dekker, Inc., New York, 2001)
13. P. Strobach, *Linear Prediction Theory* (Springer, New York, 1990)
14. B. Widrow, S.D. Stearns, *Adaptive Signal Processing* (Prentice Hall, Englewood Cliffs, 1985)
15. J.R. Treichler, C.R. Johnson Jr., M.G. Larimore, *Theory and Design of Adaptive Filters* (Wiley, New York, 1987)
16. B. Farhang-Boroujeny, *Adaptive Filters: Theory and Applications* (Wiley, New York, 1998)
17. S. Haykin, *Adaptive Filter Theory*, 4th edn. (Prentice Hall, Englewood Cliffs, 2002)
18. A.H. Sayed, *Fundamentals of Adaptive Filtering* (Wiley, Hoboken, 2003)
19. L.R. Rabiner, R.W. Schaffer, *Digital Processing of Speech Signals* (Prentice Hall, Englewood Cliffs, 1978)
20. D.H. Johnson, D.E. Dudgeon, *Array Signal Processing* (Prentice Hall, Englewood Cliffs, 1993)
21. D.G. Luenberger, *Introduction to Linear and Nonlinear Programming*, 2nd edn. (Addison Wesley, Reading, 1984)
22. A. Antoniou, W.-S. Lu, *Practical Optimization: Algorithms and Engineering Applications* (Springer, New York, 2007)
23. T. Kailath, *Linear Systems* (Prentice Hall, Englewood Cliffs, 1980)