

第一部分 方法论与通用方法

第 1 章 引言

第 2 章 算法设计方法

第 3 章 信号的数学建模

第 4 章 噪声的数学建模

第 5 章 信号模型选择

第 6 章 噪声模型选择

第 7 章 性能评估、测试与文档

第 8 章 使用大定理的最佳方法

第 1 章 引 言

1.1 动机和目标

在过去的 40 多年中，数字信号处理(Digital Signal Processing, DSP)的概念、方法、技术及在民用产品和军事系统中的应用得到了爆炸式的实际增长。一些致力于数字信号处理的基础刊物，如创刊于 1974 年的《IEEE 声学、语音及信号处理学报》(*IEEE Transaction on ASSP*)最初是双月刊，每期约 100 页。而在今天，专注于信号处理的《IEEE 信号处理学报》(*IEEE Transaction on Signal Processing*)则是月刊，每期约 500 页，在论文的数量上呈现了 10 倍的增长，这还没有考虑一些扩展出来的刊物，例如《IEEE 声学、语音及语言处理学报》(*IEEE Transaction on Audio, Speech, and Language Processing*)、《IEEE 图像处理学报》(*IEEE Transaction on Image Processing*)等。那些必须选择并实现一种方法的算法设计者现在不得不面对着许多可用算法。这些算法令人眼花缭乱，甚至从公开文献中找出一种期望的算法也是一项巨大的工作。因此，对于算法设计者来说，拥有一个他尝试过并信赖的算法库显得尤为重要。这些方法可能不能完全解决他所面临的问题，但至少为算法的开发提供了一个良好的开端。

除了积累一套可信的算法之外，关键的问题是理解这些算法是如何工作的，以及何时可能失效。DSP 算法以及更特殊的统计信号处理算法从本质上来讲都是高度数学化和统计化的，并不能轻易地产生应用的秘诀。但当设计者实现这些算法并考察其性能时，他们的直觉与未来算法选择的成功息息相关，这种直觉只能从实践中获得。幸运的是，我们无须在硬件上实算法才能评估它们的性能，软件实现很容易，并且允许进行性能的无损评估。MATLAB 是一种流行的多功能软件，它是实现算法和评估性能的工具，它的应用允许我们“运行”提出的算法，提供一种软件上的“首次切割”实现。事实上，MATLAB 的实现也常常带来 DSP 上或专用数字硬件上的实现。基于以上理由，我们将在本书中强调 MATLAB 的应用。

本书的内容是关于统计信号处理的算法。另一方面，那些数学形式完全已知且没有受到过多噪声影响的信号的处理，已经有很多标准的、十分可靠的技术存在。例如，典型的技术包括滤波器设计和傅里叶变换的计算(即快速傅里叶变换, FFT)。很多优秀的教材描述了这些算法以及它们的实现[Ingles and Proakis 2007, Lyons 2009]。而我们的目的是介绍可用来从随机数据中分析和提取信息的那些算法。例如，给定一些指标，如信号的频谱本质上是低频的信号，那么在做进一步处理之前，应当先通过低通数字滤波器进行滤波，这样的指标要求自然就需要根据一个预先设定的截止频率来进行数字滤波器设计。而对于一个中心频率未知的带通信号的滤波，其指标就会有些差别。第一种情况下，其指标是完备的；而在第二种情况下，需要确定如何调整好滤波器的中心频率让信号通过，从而尽可能地消除噪声。前者称为确定性信号的处理；而后者要求中心频率的估计，最好是在线估计，以便在信号中心频率变化时我们的算法仍能提供合适的带通滤波器。当信号的特性存在不确定性时，只有统计学的方法才能奏效。

随机数据分析的算法是针对一些特定问题的。也就是说，每一个实际的信号处理问题都

是独一无二的，需要特定的方法，尽管它们通常可以和许多其他问题联系起来。因为新的电子系统和产品的开发是无止境的，不可能使用一些现成的算法。令人欣慰的是，还是存在一组“核心”算法，它们位于大多数实际的信号处理系统的中心，这就是本书要介绍并用 MATLAB 实现的算法，下面我们对这些核心算法做一般性的讨论。

1.2 核心算法

信号处理问题要求信号的检测和信号参数的估计，这就存在一些统计的且被广泛接受的方法。例如，用于检测的匹配滤波器，用于参数估计的最大似然估计器及其实现、最小二乘估计等，我们重点关注的正是这些广泛接受的方法。在实际工作中，信号处理的算法设计者至少有一个好的起点来开始实际的设计，这是令人鼓舞的。许多核心的方法(除了一些高级的但还没有得到实践检验的方法)在 *Fundamentals of Statistical Signal Processing* [Kay 1993, Kay 1998] 的前两卷和 *Modern Spectral Estimation: Theory and Application* [Kay 1988] 中进行了详细的介绍，后一本书是关于谱分析的，对于随机信号的建模是十分重要的，该书还提供了有关随机信号的由计算机产生的许多有用算法。鼓励读者参考这些书，以便充分理解这些方法的理论基础。本书内容包括：

1. 介绍实际应用的重要算法；
2. 介绍这些算法得到成功应用的一些假设；
3. 介绍它们的性能和实际中的一些限制。

本书将完成上述目标而不参考上面提及的书籍。

1.3 容易的、难的和不可能的问题

由于我们的目标是介绍在实际中广泛使用的统计信号处理算法，或许会问，这些算法获得了这么高的荣耀值得我们去介绍吗？这有两方面的原因。第一是它们在“工作”，第二是它们可以方便地用数字软件/硬件来实现。一个算法在“工作”，它必须满足系统的性能指标。例如，可能是估计一个参数所要求的性能指标，估计器的性能应该是相对误差不大于 2.5%。因此，一个算法是否在“工作”并不取决于算法本身，如果提出的性能指标不合理，那么所提出的算法或者任何一种方法都不能“工作”。因此，评估满足性能要求的可行性则是重要的。对于后一个例子，常用的可行参数估计精度是克拉美-罗下限 (Cramer-Rao Lower Bound, CRLB) (参见 8.2.1 节)，它提供了无偏估计器(即估计器的均值等于真值)方差的下限。如果性能指标在理论上不能满足，那么做进一步的设计是毫无意义的。如果可能，或许我们需要更精确的传感器，或者更多的数据。给定一个信号模型和噪声模型(我们将在第 3~6 章做进一步的讨论)，信号处理有能力提供获取所有有用信息的可实现算法，我们希望这些信息足以产生期望的性能。然而，信号处理也不能做不可能的事情，尽管我们是多么想要这么去做。例如，假定希望估计一个恒定的离散时间信号 $s[n]$ 的值 A ，该值也称为 DC(直流)电平信号(假定是连续时间信号经 A/D 转换器采样而来)。这个信号为

$$s[n] = A \quad n = 0, 1, \dots, N-1$$

信号被淹没在功率为 σ^2 的零均值高斯白噪声 (White Gaussian Noise, WGN) 中(参见 4.3 节)。

那么, 观测数据为 $x[n] = A + w[n]$, $n = 0, 1, \dots, N-1$ 。我们希望根据这些数据尽可能精确地确定 A 的值。众所周知, 这样做的最佳方法是采用由样本均值给出的估计器

$$\hat{A}_N = \frac{1}{N} \sum_{n=0}^{N-1} x[n]$$

(字母上的“帽子”总是表示估计)。假定 $A = 10$, 我们的指标要求是: 对于 $N = 20$ 、 $\sigma^2 = 1$, 按 2.5% 的最大相对误差, 估计值应当落在 $[A - 0.25, A + 0.25]$ 内, 也就是 $[9.75, 10.25]$ 。计算机模拟结果如图 1.1 所示, 图中给出的是估计量 \hat{A}_N 与数据记录长度 N 之间的关系图^①。由于对应的是 $N = 20$ 时的指标, 因此结果看起来是满足的, 也就是当 $N = 20$ 时, 估计值落在虚线 9.75 和 10.25 之间。

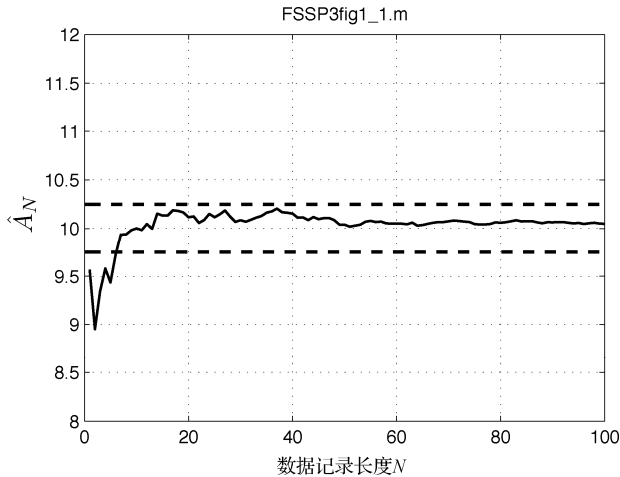


图 1.1 DC 电平 A 的估计与数据记录长度 N 的关系

然而, 如果我们重复进行实验, 这样好的结果可能只是偶然出现, 对于不同的 WGN 样本 $w[n]$, 可能得到不同的结果。对于 5 个不同的 WGN 现实(每个现实是不同的, 它们的长度都是 $N = 100$), 在图 1.2 中给出了 5 个不同的实验结果。

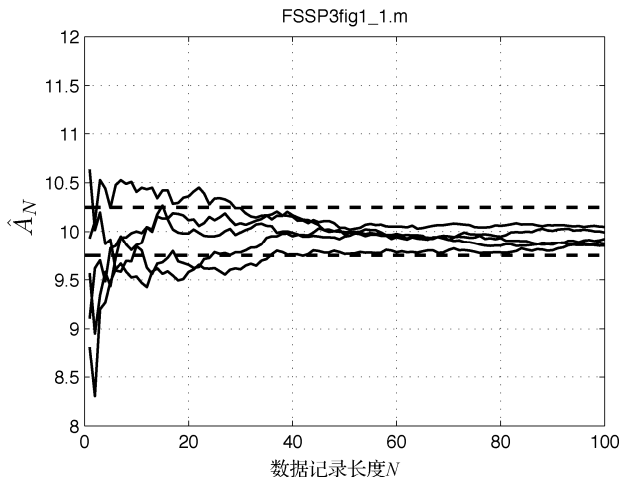


图 1.2 DC 电平估计器的 5 个不同现实

^① 产生本图的 MATLAB 程序名列在图的顶部, 这一程序可以从作者那里获得。

现在看来在 $N = 20$ 个数据点时性能指标并不满足，似乎只有当 $N > 40$ 个点时才满足。事实上，如果要求 $N = 20$ 时估计值有 95.5% 的值落在虚线内，那么可以证明估计量的方差 $\text{var}(\hat{A}_N)$ 必须满足

$$2\sqrt{\text{var}(\hat{A}_N)} \leq 0.25$$

或者等价于

$$\text{var}(\hat{A}_N) \leq \frac{1}{64}$$

但是克拉美-罗下限却说明所有无偏估计必须满足

$$\text{var}(\hat{A}_N) \geq \frac{\sigma^2}{N} \quad (1.1)$$

在 $\sigma^2 = 1$ 、 $N = 20$ 时得到的下限是 $1/20$ ，所以，这时的性能指标是无法满足的。后面将会看到，估计量 \hat{A}_N 称为样本均值估计，它确实会达到下限值，所以它的方差由式 (1.1) 给出。因此，要满足性能指标必须要求

$$\frac{\sigma^2}{N} = \frac{1}{64}$$

也就是说，当 $\sigma^2 = 1$ 时， N 最小为 64。顺便提醒一下，读者只要留意一下图 1.1 和图 1.2 就可知道其含义，由于结果依赖于产生的特定的噪声序列，实验必须重复多次，没有什么结论只从少数几个现实得出。实际上，图 1.2 中应当增加更多的现实。

练习 1.1 性能分析时多个现实的必要性

- 运行 MATLAB 程序 FSSP3exer1_1.m，取 100 个噪声现实， $N = 64$ 。大约有 95 个估计值落在指定的间隔 $[9.75, 10.25]$ 之内吗？
- 如果精度的性能指标增加到 1% 的最大相对误差，要求将误差控制在 1% 内，方差 $\text{var}(\hat{A}_N)$ 应该是多少？数据记录长度 N 应该等于多少才能满足达到要求的方差？最后，修改 FSSP3exer1_1.m 的程序代码来模拟这个更大的 N 的性能，有多少估计值满足性能指标？

信号处理在许多领域都有应用。例如，当信号淹没在噪声中时，如何精确地确定正弦信号的频率？数学上最佳的方法就是使用周期图，这是一种谱估计器，取周期图最大值的位置作为频率的估计（见算法 9.3），这在实际工作中很有效并被广泛采用。此外，即使最佳算法所假定的条件有些不符合，性能也不会显著下降，这就是“稳健算法” (robust algorithm) 的例子。在实践中，稳健性是一个算法非常重要的特性。现实的数据极少完全符合理论上的假设，这些假设通常都是为了简化最佳算法的数学推导而给出的。我们可以把单一正弦信号频率的估计器的设计称为“容易问题”，因为它的解就是周期图，并且性能好、实现方便（通过傅里叶变换）。当然，在应用中，我们假定了单一的正弦信号加上高斯白噪声，这是不可忽视的假设。如果出现其他的正弦信号或干扰信号，或者噪声是高度相关的，那么这一方法就没有所说的那样好。特别是，如果有另一个正弦信号的频率靠近感兴趣的正弦信号的频率，那么频率估计器就可能有严重的偏差，从而产生了一个不精确的估计。图 1.3 中给出了这样一个例子。在

图 1.3(a) 中, 显示一个幅度为 1、频率为 0.2 的正弦信号的周期图, 从图中可以看出周期图的最大值出现在正确的频率位置。而图 1.3(b) 给出了两个正弦信号的和, 期望信号的频率为 0.2, 干扰信号的频率为 0.22, 幅度都为 1。可以看出峰值现在偏离期望的频率 0.2。为了考虑这种可能性, 我们必须改变想法, 要认识到潜在的有一个或者多个干扰正弦信号。增加的复杂性就导致了“难的问题” [Kay 1988], 可能没有一种好的解决方法, 尤其是当干扰正弦信号的频率未知时。读者会毫不怀疑地认为, 对于一个算法, 当设计的假设得到满足时, 这个算法就会工作得很好(有一些可以说性能是最佳的)。因此, 问题的关键就是要验证这些假设在实际中成立。为了对令人失望的结果有所准备, 我们不仅需要评估算法的性能, 还要评估算法的局限性。

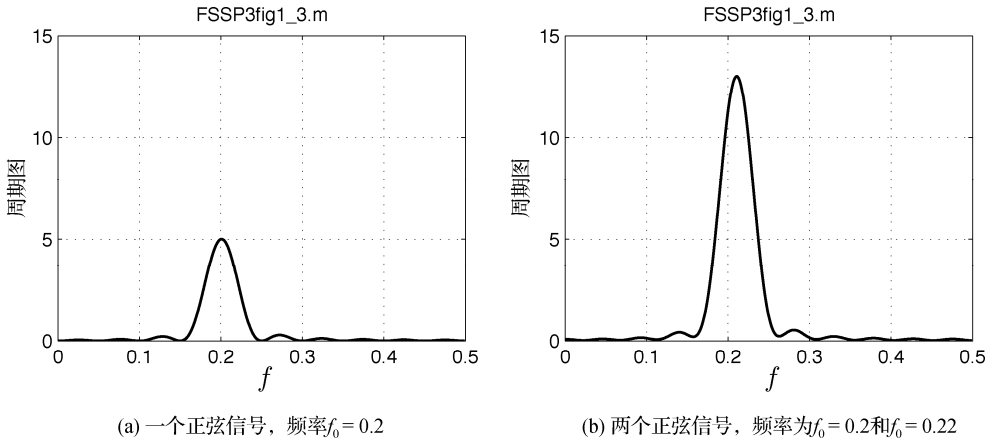


图 1.3 一个正弦信号和一个正弦信号加一个干扰信号的周期图

往往会有这样的情况, 一个看似非常困难的或者“难的问题”, 如果处理得当, 能将问题转化为一个容易解决的问题。众所周知, 线性信号模型(见 3.5 节和 3.6.4 节)能够得出最优性能和易于实现的算法。在现实世界中, 并非所有信号都适合使用线性模型。然而, 一旦知道了线性模型所期望的性质, 我们会理所当然地将非线性模型变换成线性模型。仍以正弦信号为例, 它可以用离散时间信号的形式来描述,

$$s[n] = A \cos(2\pi f_0 n + \phi) \quad n = 0, 1, \dots, N-1 \tag{1.2}$$

其中 A 是未知幅度, 并且 $A > 0$; f_0 是未知的频率, 并且 $0 < f_0 < 1/2$; ϕ 是未知的相位, 并且 $-\pi \leq \phi < \pi$, 我们想要估计未知的幅度和未知的相位。按照现在的情况, 这个信号与相位之间是非线性关系(因为 $A \cos(2\pi f_0 n + \phi_1 + \phi_2) \neq A \cos(2\pi f_0 n + \phi_1) + A \cos(2\pi f_0 n + \phi_2)$), 这将使估计算法的建立变得复杂。为了使问题更易于处理, 利用三角恒等式 $\cos(C+D) = \cos(C)\cos(D) - \sin(C)\sin(D)$, 可得

$$s[n] = A \cos(\phi) \cos(2\pi f_0 n) - A \sin(\phi) \sin(2\pi f_0 n)$$

然后令 $\alpha_1 = A \cos(\phi)$, $\alpha_2 = -A \sin(\phi)$ (刚好是极坐标到直角坐标的变换), 可得

$$s[n] = \alpha_1 \cos(2\pi f_0 n) + \alpha_2 \sin(2\pi f_0 n) \tag{1.3}$$

现在信号与未知的变换后的参数 α_1 、 α_2 是线性的, 这样就把原来难的问题转化为一个相对容易的问题, 并且它的解也是众所周知的。可以证明, 如果观测到的为正弦信号加噪声, 那么

基于观测数据集 $\{x[0], x[1], \dots, x[N-1]\}$ 的幅度和相位的好的估计器为 (见 3.5.4 节和算法 9.2)

$$\hat{A} = \frac{2}{N} \left| \sum_{n=0}^{N-1} x[n] \exp(-j2\pi f_0 n) \right|$$

$$\hat{\phi} = \arctan \left(\frac{-\sum_{n=0}^{N-1} x[n] \sin 2\pi f_0 n}{\sum_{n=0}^{N-1} x[n] \cos 2\pi f_0 n} \right) \quad (1.4)$$

但如果这样做, 我们必须熟悉容易问题 (线性信号模型), 接着选择一个合适的变换。在实际中, 关于那些早已经知道其解的容易问题的知识是不可缺少的, 本书包含了许多这种众所周知的解。

在实际中, 大多数信号处理问题都是难的问题 (如果不难, 肯定有人早就解决了这些问题!)。但是知道这些容易问题的解, 以及建立了熟悉这些问题所带来的直觉, 就可以引导我们求出更难问题的解。例如在式 (1.3) 中, 可能不知道频率 f_0 , 如何才能估计频率、幅度和相位呢? 可以证明, 只要 f_0 不在 0 或者 1/2 附近, 时间截断的正弦信号 [即 (1.2) 式] 的离散时间傅里叶变换为

$$|S(f)| = \left| \sum_{n=0}^{N-1} s[n] \exp(-j2\pi f n) \right| \quad (1.5)$$

$$\approx \frac{NA}{2} \left| \frac{\sin[N\pi(f - f_0)]}{N \sin[\pi(f - f_0)]} \right| \quad (1.6)$$

接下来的练习中要求你证明这一关系 (解包含在附录 1A 中)。

练习 1.2 关于截断的正弦信号的离散时间傅里叶变换的推导

证明式 (1.6) 给出的表达式。为此, 首先将式 (1.2) 中实的正弦信号分成两个复共轭分量 $\exp(j2\pi f_0 n)$ 和 $\exp(-j2\pi f_0 n)$, 并代入式 (1.5) 中, 然后利用复几何级数的结果

$$\left| \sum_{n=0}^{N-1} z^n \right| = \left| \frac{1 - z^N}{1 - z} \right| = \left| \frac{z^{-N/2} - z^{N/2}}{z^{-1/2} - z^{1/2}} \right|$$

其中第一个等式对所有复 z 都是成立的, 第二个等式只对 $|z|=1$ 成立, 其中 $|\cdot|$ 表示复数的模。最后, 放弃负频率部分, 如果我们只计算 $f > 0$ (f_0 不在 0 或 1/2 时) 的傅里叶变换, 则这是可以忽略的。

$|S(f)|$ 的图像如图 1.4 所示, 其中 $A=1$, $N=20$, $f_0=0.2$ 。这表明可以用峰值的位置来估计频率。实际上这也形成了前面提到的图 1.3(a) 显示的周期图估计器的基础, 尽管在我们的讨论中并没有包含噪声的影响, 但即使有噪声, 周期图估计器仍然是性能优良的 (见算法 9.3)。按照直观的理解, 对于大数据记录 (即 $N \rightarrow \infty$), 这一结果是显然的, 因为傅里叶变换变成了冲激函数, 在任何噪声背景下, 冲激都是非常突出的。但也不要过分强调直觉, 防止在设计算法时走入死胡同, 除了数学上的证明之外, 这也解释为什么好的算法工作得好。如何才能获得这样一种直觉呢? 答案是: 实践、实践、再实践!

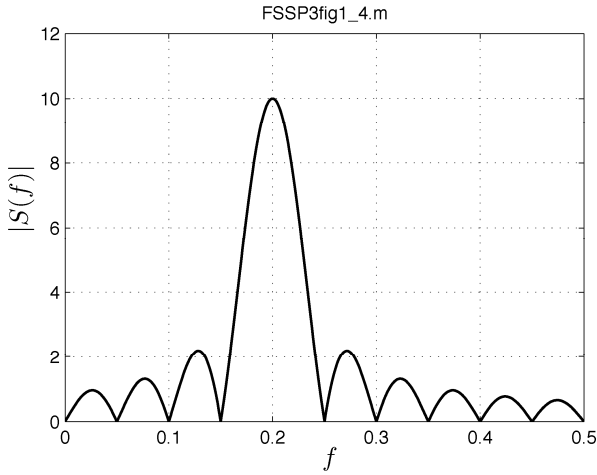


图 1.4 时间截断正弦信号的离散时间傅里叶变换的幅度

1.4 增加成功的概率——提升直觉

为了建立我们的直觉，必须边学边做。幸运的是，现在有很多可为我们提供实践的软件包和硬件开发装置。因为我们的目标是用软件设计算法，所以我们的讨论将限定软件的实现上。我们发现 MathWorks 软件包的 MATLAB 特别适合在实践中遇到的信号处理问题，因此，本书将使用 MATLAB 作为建立和测试统计信号过程算法的基本工具，甚至书中的一些图形也是用 MATLAB 产生的(所使用的程序都列在图形的顶部，也可以向作者申请获得这些程序)，同时也使用 MATLAB 来求解练习中的问题。练习的解答、实现算法的程序以及一些有用的函数都由随书的光盘提供^①。

分析算法的 MATLAB 实现是理解算法工作原理的基本方式。此外，利用 MATLAB 产生的控制数据集来测试算法是验证算法效率的基本的方法，在验证该算法有效性时也是十分必要的。在观察算法的 MATLAB 程序的输出并与预期的输出相比较时，你会得到一种强烈的直觉意识。最后，也可以利用相同的程序代码、采用计算机仿真的方法来评估算法的理论性能，尽管这样做通常都不是实时处理的。对于所有这些情况，在发展个人直觉能力方面，实践是基本的。在完成分布在书中的练习的过程中，我们将会拥有足够多的机会去实践。

1.5 应用领域

本书将要描述的统计信号处理算法广泛应用于许多领域。例如：

1. 通信——数字信息的传输与接收[Proakis and Salehi 2007]
2. 雷达和声呐——目标检测、定位、跟踪[Richards 2005, Burdick 1984]
3. 生物医学——心律失常检测、脑机接口[Sörnmo and Laguna 2005, Sanei and Chambers 2007]

^① 中文翻译版不再配有光盘，相关内容请登录华信教育资源网(www.hxedu.com.cn)注册下载。

4. 图像处理——医疗诊断成像、数据压缩[Gonzalez and Woods 2008]
5. 语音处理——语音识别与综合[Rabiner and Schafer 2007]
6. 无线电导航——全球定位系统[Pany 2010]

以上仅是众多应用领域的一小部分，而且应用领域每日都在增加。初看可能会感到奇怪，这些不同的领域采用几乎相同的统计信号处理算法。这并非总是如此，这种影响力主要是由于信号处理的实现都是使用数字计算机，无论信号是电(如雷达)、声(如声呐和语音)还是光(如成像)，最终都归结为输入并存储在数字计算机中的一组数。从物理信号(如语音)变换到一组数是由传感器(如话筒)接一个 A/D 转换器来完成的，经 A/D 转换器产生的字节存储在数字计算机中并做进一步处理。从信号处理的角度来看，它们唯一的区别是信号的特征。信号可能是低通的，信号的大部分能量集中在低频段，如语音信号；信号也可能是带通的，信号的大部分能量集中在一个给定的频带内，如雷达信号。信号从本质上可以是一维的，如声信号；也可以是二维的，如图像信号。A/D 转换器的采样率就需要调整以适应所考虑的信号，导致所处理的数据量不同。然而，算法在处理这些不同的信号时，即使不相同，也是有很大的相似之处。匹配滤波器应用于声呐时，信号的频谱在 kHz 频率范围内。匹配滤波器也可以应用于雷达，但雷达信号的频谱更高，通常在 GHz 范围内。FFT 既可以用于一维信号(如语音)，二维 FFT 也可用于图像分析。因此，通常我们都是根据信号的数学描述来设计算法，而不管信号的来源。当然这样也许会忽略现实世界中的一些先验知识，如实际信号必须服从的一些约束，所以后续的任何算法如果采用了这些约束，之后都有改善性能的潜力。例如，对于从一个移动目标反射回来的信号，雷达设计就可以根据目标的最大速度来约束它的最大多普勒频率。

1.6 注意事项

1.6.1 信号类型

在描述算法时，我们假定物理信号是实的低通信号，并以合适的速率(至少是信号最高频率的两倍，即奈奎斯特速率)进行采样。而对于一些处理带通信号的应用领域，如雷达、声呐，通常采用复杂解调、紧跟一个同相和正交信号的采样，这导致了稍微有些复杂的复信号表示，复信号的扩展将在第 12 章中描述。此外，由于信号经过采样并存储在数字计算机中，所以信号本质上是离散时间的，我们总是假定所接收的数据形式为 $x[n]$ ，这是一个实数 n 序列，如果数据记录由 N 个连续时间样本组成，通常就使用索引集 $n = 0, 1, \dots, N-1$ 。注意，我们已经把样本看做时间的函数，然而索引 n 也可能表示空间的样本，即在某个给定的时间由放置在一条线上的等间隔的 N 个传感器得到的。

1.6.2 本书的特点和符号表示

教材中使用的 MATLAB 版本是 7.8(R2009A)，运行 MATLAB 程序并不要求使用工具箱。在附录 B 简要介绍了 MATLAB，附录 C 描述了随书光盘的所有 MATLAB 程序。此外，光盘上的 `readme.txt` 介绍了这些程序的内容。`run_simulation.m` 这种字体表示 MATLAB 程序的名称和代码。

书中给出了一些练习，这些练习为读者提供了简单的理论分析和 MATLAB 算法实现的一些实践。理论分析的答案包含在对应章节的附录中。MATLAB 练习只给出了粗略的答案，完整的解答在随书的光盘上能够找到。注意，答案是用 MATLAB R2009A 版本给出的，所以，新的 MATLAB 版本可能会产生略有不同的结果。鼓励读者去尝试这些练习，以便更好地理解本书的内容。

在每章的最后都有一节“小结”，其中给出一些重要的结论，许多结论都成为了“经验法则”，因此应努力记住。对于某些应用，它们是探索算法效率的基础，或者是算法不适合当前应用而被拒绝的依据。

所有常见符号的数学表示总结在附录 A 中，区别一个连续时间和离散时间信号或序列的表示方法是 $x(t)$ (连续) 和 $x[n]$ (离散)。然而，离散时间数据的图 (例如 $x[n]$) 可能是连续时间形式，点之间用直线连接更方便查看，图 1.1 中就是这样的一个例子。所有的向量和矩阵都用黑体表示，所有的向量都是列向量。当一个随机变量需要与它的值进行区分时，我们就使用大写字母表示随机变量，如 X ，而用小写字母表示它的值，如 x 。附录 A 还给出了其他符号的含义。另外，读者经常会被警告出现潜在的“陷阱”。常见的误解所导致的设计错误将被记录和描述。下面的陷阱或警示标志应特别注意！



1.7 小结

下面列出的内容是本书反复出现的主要内容，在考察算法及性能时，需要注意：

- 在算法设计中，评估算法要求的可行性是第一步。通常情况下，克拉美-罗下限用于估计，奈曼-皮尔逊 (Neyman-Pearson) 似然比检验的检测概率用于检测。对于分类问题，采用最大后验概率 (MAP) 分类器。对于所有这些情况，概率密度函数必须是已知的。
- 如果达不到性能指标的要求，需要对目标进行再评估/或要求更精确的数据。
- 信号处理不能做不可能的事情——信号处理的成功必须要有合理的信噪比作为保障。
- 通过计算机模拟确定算法的性能时，实验要重复进行多次，比如 1000 次或更多。确保你的性能度量本质上是统计的，如估计量的方差、检测器的检测概率、分类器的错误概率等。增加实验次数，直到这些指标的评估产生一致的结果。
- 确信算法在不同工作条件下进行过测试，评估算法的稳健性，也称为灵敏度。
- 通过分析实际的数据来验证算法的假设在实际中是成立的。
- 试着将问题转化为一个更简单的问题，例如，转化为线性信号模型。
- 首先用 MATLAB 产生的数据来测试算法，如果算法有效，结果应该与理论预测一致，在 MATLAB 使用的条件下得到的性能应该是现场数据的上限。
- 在提出一个算法之前，通过公开文献搜索其他领域和用过的一些方法中类似的信号处理问题。

参考文献

- Gonzales, R.C., R.E.Woods, *Digital Image Processing*, Prentice Hall, Upper Saddle River, NJ, 2008.
- Ingle, V.K., J.G. Proakis, *Digital Signal Processing Using MATLAB*, 2nd ed., Cengage Learning, Stamford, CT, 2007.
- Kay, S., *Modern Spectral Estimation: Theory and Application*, Prentice-Hall, Englewood Cliffs, NJ, 1988.
- Kay, S., *Fundamentals of Statistical Signal Processing: Estimation Theory, Vol. I*, Prentice-Hall, Englewood Cliffs, NJ, 1993.
- Kay, S., *Fundamentals of Statistical Signal Processing: Detection Theory, Vol. II*, Prentice-Hall, Englewood Cliffs, NJ, 1998.
- Lyons, R.G., *Understanding Digital Signal Processing*, Prentice Hall, Upper Saddle River, NJ, 2009.
- Pany, T., *Navigation Signal Processing for GNSS Software Receivers*, Artech House, Norwood, MA, 2010.
- Proakis, J., M. Salehi, *Digital Communications*, 5th ed., McGraw-Hill, NY, 2007.
- Rabiner, L.R., R.W. Schafer, *Introduction to Digital Speech Processing*, Now, Boston, 2007.
- Richards, M.A., *Fundamentals of Radar Signal Processing*, McGraw-Hill, NY, 2005.
- Sanei, S., Chambers, J.A., *EEG Signal Processing*, Wiley-Interscience, NY, 2007.
- Sörnmo, L., P. Laguna, *Bioelectrical Signal Processing in Cardiac and Neurological Applications*, Elsevier Academic Press, NY, 2005.

附录 1A 练习解答

为了得到下面描述的结果, 在程序的开始用 $\text{rand}('state', 0)$ 和 $\text{randn}('state', 0)$ 初始化随机数产生器。这些命令分别是针对均匀随机数产生器和高斯随机数产生器。

- 1.1 对于 a 部分, 开始运行代码与 $\text{randn}('state', 0)$ 程序, 使随机数产生器初始化。应该观察到 92 个估计满足性能指标, 即位于区间[9.75, 10.25]上。对于 b 部分, 要求 $2\sqrt{\text{var}(\hat{A}_N)} \leq 0.1$, 根据 CRLB, $\text{var}(\hat{A}_N) = \sigma^2 / N = 1/400$, 因此, 要求 $N = 400$ 。修改程序并运行它, 产生 95 个满足性能指标的估计, 即位于区间[9.9, 10.1]上。

1.2

$$\begin{aligned}
 S(f) &= \sum_{n=0}^{N-1} s[n] \exp(-j2\pi fn) \\
 &= \sum_{n=0}^{N-1} A \cos(2\pi f_0 n + \phi) \exp(-j2\pi fn) \\
 &= \sum_{n=0}^{N-1} \left[\frac{A}{2} \exp(j2\pi f_0 n + \phi) + \frac{A}{2} \exp(-j2\pi f_0 n - \phi) \right] \exp(-j2\pi fn) \\
 &= \frac{A}{2} \exp(j\phi) \sum_{n=0}^{N-1} \exp[-j2\pi(f - f_0)n] \tag{1A.1}
 \end{aligned}$$

$$+ \frac{A}{2} \exp(-j\phi) \sum_{n=0}^{N-1} \exp[-j2\pi(f + f_0)n] \tag{1A.2}$$

令 $z = \exp[-j2\pi(f - f_0)]$, 注意到 $|z| = 1$ 。去掉第二个和, 可得

$$S(f) = \frac{A}{2} \exp(j\phi) \sum_{n=0}^{N-1} z^n = \frac{A}{2} \exp(j\phi) \frac{1 - z^N}{1 - z}$$

取它的幅度，得

$$\begin{aligned} |S(f)| &= \frac{A}{2} \left| \frac{1-z^N}{1-z} \right| \\ &= \frac{A}{2} \left| \frac{z^{N/2}(z^{-N/2}-z^{N/2})}{z^{1/2}(z^{-1/2}-z^{1/2})} \right| \\ &= \frac{A}{2} \left| \frac{z^{-N/2}-z^{N/2}}{z^{-1/2}-z^{1/2}} \right| \end{aligned}$$

接下来令 $\alpha = -2\pi(f - f_0)$ ，我们有 $z = \exp(j\alpha)$ 和

$$\begin{aligned} |S(f)| &= \frac{A}{2} \left| \frac{\exp[-j\alpha(N/2)] - \exp[j\alpha(N/2)]}{\exp[-j\alpha(1/2)] - \exp[j\alpha(1/2)]} \right| \\ &= \frac{A}{2} \left| \frac{\sin(N\alpha/2)}{\sin(\alpha/2)} \right| \end{aligned}$$

由于 $\alpha/2 = -\pi(f - f_0)$ ，最终可得

$$|S(f)| = \frac{A}{2} \left| \frac{\sin[N\pi(f - f_0)]}{\sin[\pi(f - f_0)]} \right| = \frac{NA}{2} \left| \frac{\sin[N\pi(f - f_0)]}{N \sin[\pi(f - f_0)]} \right|$$

注意，这是近似表达式，因为我们忽略了求和中的第二项，即式(1A.2)，由于这一项是正频率项(我们得到的那一项)的镜像，且中心频率位于 $f = f_0$ ，只要对正频部分没有干扰，它的贡献是很小的，只要 f_0 不在 0 或 1/2 附近，频谱的幅度就是这样一种情况。图 1A.1 画出了式(1A.1)的幅度(正频率分量)和式(1A.2)的幅度(负频率分量)，整个频率范围为 $-0.5 \leq f \leq 0.5$ ，信号参数是 $N = 20$ ， $A = 1$ ， $\phi = 0$ ， $f_0 = 0.2$ ，可以看出，在这种情况下，正频和负频部分的干扰很小。

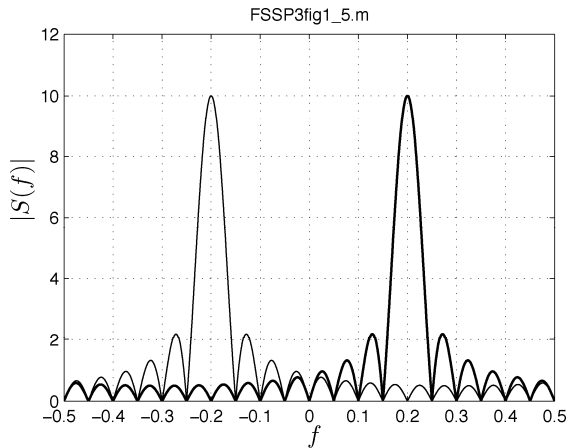


图 1A.1 频率分量的离散时间傅里叶变换的幅度，其中 $f_0 = 0.2$ ， $N = 20$ 。式(1A.1)的正频部分用粗线表示，式(1A.2)的负频部分用细线表示，两部分的显著交叉发生在 $0 < f_0 < 1/N = 0.05$ 或 $0.45 = 1/2 - 1/N < f_0 < 1/2$