

第 5 章

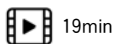


傅里叶变换和小波变换

在信号处理领域,傅里叶变换(Fourier Transform, FT)和小波变换(Wavelet Transform, WT)是非常重要的信号分析算法,本章将浅显易懂地介绍两种算法的理论,并用典型案例来理解它们的特点与应用。



5.1 傅里叶变换



傅里叶变换是一种线性的积分变换,它的目的是可将时域上的信号转变为频域上的信号,类似于将直角坐标系变换成极坐标系。随着域的不同,对同一个事物的了解角度也就随之改变,因此在时域中某些不好处理的问题,在频域就可以较为简单地解决。

5.1.1 傅里叶级数的频谱

1807 年在法国科学学会上法国学者 Fourier 发表了一篇论文,论文里描述运用正弦曲线来描述温度分布,论文里有个在当时具有争议性的决断:任何连续周期信号都可以由一组适当的正弦曲线组合而成。方波可以分解为无限多个正弦信号的组合,如图 5-1 所示。

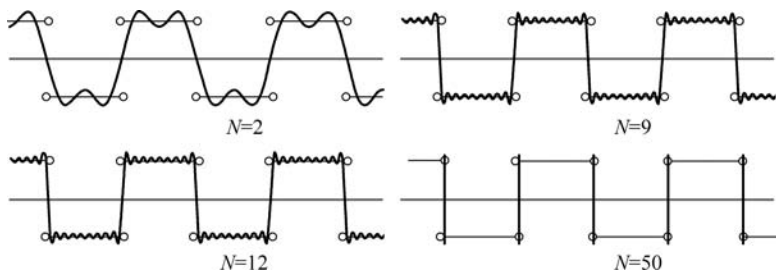


图 5-1 多个正弦波组合成方波

从图 5-1 可知,随着正弦波数量的逐渐增长,它们最终会叠加成一个标准的方波。虽然组成方波的这些信号都是正弦信号,但是这些正弦信号之间还需要满足一定的条件。考虑组成方波的正弦信号,方波可用公式(5-1)表示,其中 n 为奇数:

$$y = \frac{4}{\pi} \left(\sin(\omega t) + \frac{1}{3} (3\omega t) + \cdots + \frac{1}{n} (n\omega t) + \cdots \right) \quad (5-1)$$

这里, ω 称为基波频率, 而 3ω 、 5ω 、 $n\omega$ 等均为 ω 的整数倍。这些大于 0 基波频率, 且是基波频率整数倍的各次分量称为谐波。对于方波, 基波的各偶数次谐波的幅值为 0。这些谐波成分组成了方波。

这里, 引入“频域”的概念, 我们将方波及分解的正弦波用三维立体图展示, 如图 5-2 所示。图 5-2 中, 最前面黑色的线就是所有正弦波叠加而成的总和, 也就是越来越接近方波的那个图形, 而后面依不同颜色排列而成的正弦波就是组合为方波的各个分量。这些正弦波按照频率从低到高及从前向后排列开来, 而每个波的振幅满足公式(5-1)。每两个正弦波之间都还有一条直线, 那并不是分割线, 而是振幅为 0 的偶数谐波。

如果把第 1 个频率最低的频率分量看作“1”, 就有了构建频域的最基本单元。时域的基本单元就是 1s, 如果我们将一个角频率为 ω 的正弦波 $\sin(\omega t)$ 看作基础, 则频域的基本单元就是 ω 。时域的方波信号就被投影到了频域。因为前面的方波信号的横轴为时间轴, 而在频域, 横轴为频率。这样, 一组随时时间变化的时域正弦信号被表示为了频域的一组离散点。频域每个离散点的横坐标代表一个谐波频率, 而其纵坐标则代表该频率的谐波所对应的振动幅度。频域的图像称为频谱, 方波的频谱图如图 5-3 所示。

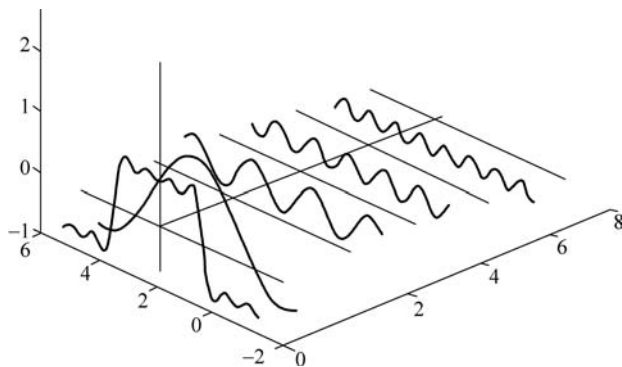


图 5-2 方波三维分解图

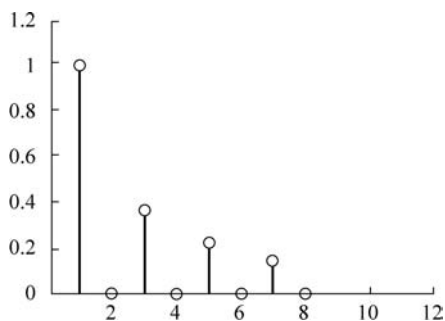


图 5-3 方波频谱

为了形象地理解, 将时域图和频域图用三维图表示, 如图 5-4 所示。从垂直频率轴且与时间相反的方向去看, 就可以得到频谱图了。

5.1.2 傅里叶级数的相位谱

通过时域到频域的变换, 我们得到了一个从侧面看的频谱, 但是这个频谱并没有包含时域中全部的信息。因为频谱只代表每个对应的正弦波的振幅是多少, 而没有提到相位。在基础正弦波 $A\sin(\omega t + \theta)$ 中, 振幅、频率、相位缺一不可, 不同相位决定了波的位置, 所以对于频域分析, 仅仅有频谱是不够的, 还需要一个相位谱。回忆一下相位的概念, 例如正弦波的相位, 如图 5-5 所示。

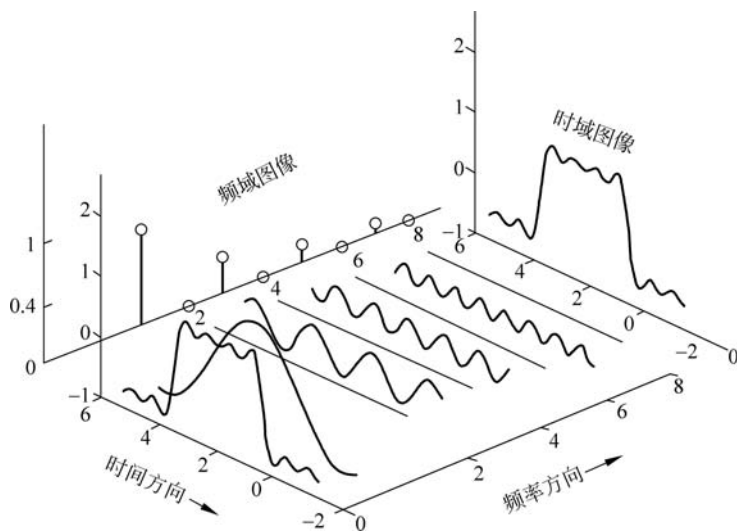


图 5-4 方波的时域和频域三维立体图

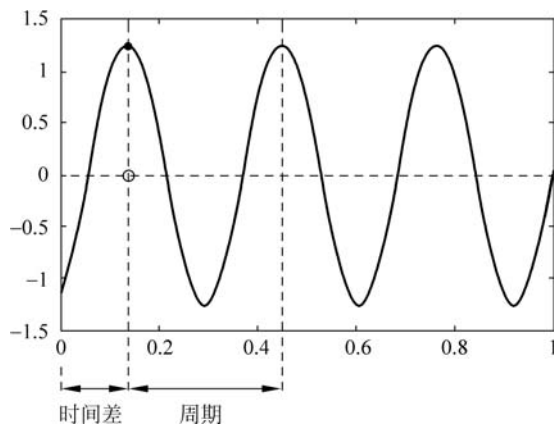


图 5-5 分解正弦波的相位

在图 5-5 中用“实心点”来标记正弦波位置,实心点是距离频率轴最近的波峰,将“实心点”投影到下平面,投影点用“空心点”来表示。当然,这些点只标注了波峰距离频率轴的距离,并不是相位。将投影得到的时间差除以所在频率的周期,就得到了相位。我们将方波分解的所有正弦波都求相位,就可以得到如图 5-6 所示的三维相位谱了。

注意,相位谱中的相位除了 0,就是 π 。因为 $\sin(t+\pi)=-\sin(t)$,所以实际上相位为 π 的波只是上下翻转了而已。另外由于 $\sin(t+\pi)=\sin(t)$,所以相位差是周期的 π 、 3π 、 5π 、 7π 都是相同的相位。人为将相位谱的值域定义为 $(-\pi, \pi]$,所以图中的相位差均为 π 。

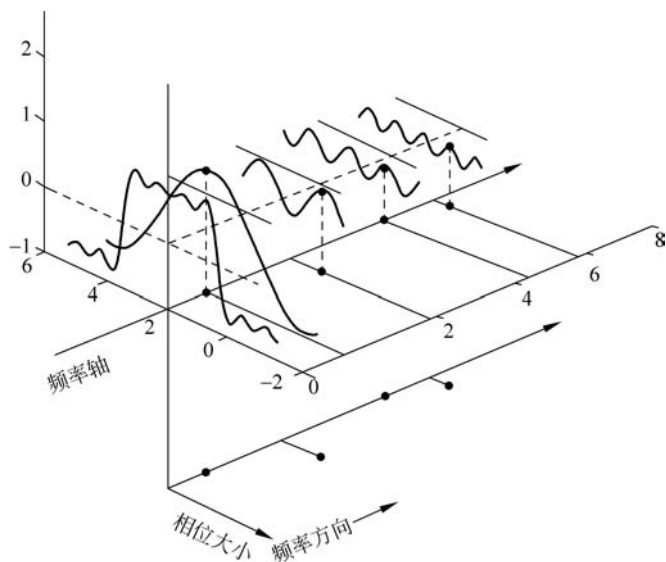


图 5-6 三维相位谱

5.1.3 傅里叶变换表示形式

傅里叶变换是把问题变得更简单,而不是变得更复杂。傅里叶变换选择了正弦波,而没有选择方波或其他波形,这是因为正弦波有个其他任何波形(恒定的直流波形除外)所不具备的特点:正弦波输入至任何线性系统,出来的还是正弦波,改变的仅仅是幅值和相位,当正弦波输入线性系统时,不会产生新的频率成分;当正弦波输入非线性系统时就会产生新的频率成分,这种新的频率成分即为谐波。

在学习自控原理时指导线性系统具备一个特点,就是多个正弦波叠加后输入一个系统,输出是所有正弦波独立输入时对应输出的叠加。也就是说,我们只要研究正弦波的输入和输出关系,就可以知道该系统对任意输入信号的响应。由此可见傅里叶的伟大之处不在于如何进行傅里叶变换,而是在于给出了“任何连续周期信号可以由一组适当的正弦曲线组合而成”这一伟大的论断。

在不同的研究领域,傅里叶变换具有多种不同的变体形式,如连续傅里叶变换和离散傅里叶变换。可以分为 4 个类别:

1. 连续时间、连续频率——傅里叶变换(FT)

$$X(j\Omega) = \int_{-\infty}^{\infty} x(t) e^{-j\Omega t} dt \quad (5-2)$$

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(j\Omega) e^{j\Omega t} d\Omega \quad (5-3)$$

2. 连续时间、离散频率——傅里叶级数(FS)

$$X(jk\Omega_0) = \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} x(t) e^{-jk\Omega_0 t} dt \quad (5-4)$$

$$x(t) = \sum_{k=-\infty}^{\infty} X(jk\Omega_0) e^{jk\Omega_0 t} \quad (5-5)$$

3. 时间离散、连续频率——序列傅里叶变换(DTFT)

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x(n) e^{-j\omega n} \quad (5-6)$$

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega}) e^{j\omega n} d\omega \quad (5-7)$$

4. 离散时间、离散频率——离散傅里叶变换(DFT)

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}nk} \quad (5-8)$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j\frac{2\pi}{N}nk} \quad (5-9)$$

注意: $x(n)$ 有个 $1/N$ 。

离散傅里叶变换(DFT)在频域和时域都是离散的,即计算机可以处理的,因此 DFT 是可以实际进行编程并实现的。在实际应用中通常采用快速傅里叶变换(Fast Fourier Transform,FFT)计算 DFT。由于数字信号处理的其他运算都可以由 DFT 实现,因此 FFT 算法是数字信号处理的重要基石。

5.1.4 MATLAB 的傅里叶变换函数

在 MATLAB 中有 fft、fft2 傅里叶函数进行一维和二维快速傅里叶变换,这里介绍一维的快速傅里叶函数。

1. 一维快速正傅里叶变换函数 fft

格式如下:

```
X = fft(x, N)
```

功能: 采用 FFT 算法计算序列向量 x 的 N 点 DFT 变换,当 N 缺省时,fft 函数自动按 x 的长度计算 DFT。当 N 为 2 整数次幂时,fft 按基 2 算法计算。

2. 一维快速逆傅里叶变换函数 ifft

格式如下:

```
x = ifft(X, N)
```

功能：采用 FFT 算法计算序列向量 X 的 N 点 IDFT 变换。

3. fft 函数使用说明

1) 采样频率 F_s 的确定

实际工程采用 A/D 转换对连续信号采样,要确定一个参数采样频率 F_s ,采样要满足采样定理要求,即 $F_s \geq 2f_c$ 的采样时,这样信息不能丢失(f_c 为原始连续信号的最大截止频率),采样周期为 $T_s = 1/F_s$ 。

2) 序列个数 L 的确定

确定了采样频率 F_s ,那么信号的时域应该怎么确定呢?换句话说,将采样好的信号给计算机处理时,应该给计算机多少个。这里就出现了第 2 个参数,即序列个数 L ,可由采样周期 T_s 得出时域信号的横坐标时间的变化范围,也就是 $t = (0:L-1)T_s$ 要用 DTF 可以表示这个信号时, t 的范围应该是信号的一个周期范围或整数倍周期范围,否则则会发生频谱泄露,频谱泄露就是出现了本来没有的频率分量。 L 是用 $LT_s = mT$ (T 为信号的最小周期, m 为正整数)来确定,通常 $m=1$ 。只要保证 $LT_s \geq T$ 就可以了,当然如果可以取整数倍就最好了。有时, L 都是取和 F_s 一样的值(频率分辨率为 1Hz)。

3) N 的确定

第 3 个参数 $N \geq L$,否则就会发生时域混叠, N 通常为 $2m$,有时取 $L=N$,当发生频谱泄露比较严重时,我们就可以考虑将 L 增大,这样就可以减小频谱泄露的影响。

综上,以上就是 MATLAB 相关的所有参数:采样频率 F_s ,原序列长度 L ,N-DFT 变换的 N ,通常取值为 $F_s=1024$, $N=1024$, $L=1024$,注意需满足上述条件。对于 MATLAB 的 fft 函数,其用法主要是 $Y=\text{fft}(x,N)$, x 为原信号序列, Y 为 DFT 变换后的,但是会发现其变换后幅度变了而且变了很大,其频域幅值增大了 N 倍。

5.1.5 傅里叶变换的信号降噪应用

工程中的信号一般会受到噪声的干扰,从而导致在提取信号特征信息时造成误差,所以在处理信号之前应先进行降噪。一个含噪声的一维信号模型可表示为

$$S(k) = f(k) + e(k) \quad (5-10)$$

其中, $S(k)$ 为含噪信号, $f(k)$ 是有用信号, $e(k)$ 是噪声信号。这里假定噪声是高斯白噪声,频谱一般分布为整个频域。

在实际工程中 $f(k)$ 通常为低频信号或者频谱范围分布有限的信号,因此,通过离散傅里叶变换得到信号的频谱以后,可以把有用的频谱之外的噪声频谱去除,然后经过离散傅里叶逆变换就可以得到降噪的信号了。

【例 5-1】 利用傅里叶变换对含有噪声的 $f=50\text{Hz}$ 正弦信号进行降噪处理。

代码如下:

```

% chapter5/test1.m
t = 0:0.001:1
r = sin(2 * pi * 50 * t)           % 频率为 50Hz 正弦信号
x = r + 0.3 * rand(size(t))       % 产生含有噪声的信号
subplot(3,1,1)
plot(t,r)                         % 显示原信号
title('原始信号')
xlabel('时间/s')
ylabel('幅值')
subplot(3,1,2)
plot(t,x)                         % 显示含噪声信号
title('含噪声信号')
xlabel('时间/s')
ylabel('幅值')
y = fft(x,1024)                  % 傅里叶变换
f = 1000 * (0:512)/1024
subplot(3,1,3)
plot(f,abs(y(1:513)))            % 频谱图
title('频谱图')
xlabel('频率/Hz')
ylabel('幅值')

```

提示：rand 函数产生由在(0,1)均匀分布的随机数组成的数组。 $f_s = 1000\text{Hz}$, Nyquist 频率为 $f_s/2 = 500\text{Hz}$ 。整个频谱图是以 Nyquist 频率为对称轴的, FFT 对信号进行谱分析, 只需考察 0~Nyquist 频率范围内的幅频特性。

运行的结果如图 5-7 所示。

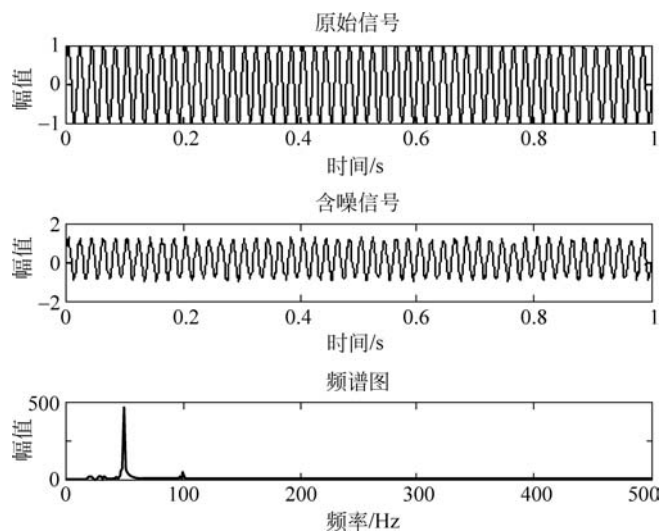


图 5-7 原信号、含噪声信号及频谱

【例 5-2】 设计一个 100Hz 的低通模拟滤波器。

(1) 构造一个 20Hz 正弦信号和一个 200Hz 余弦信号的混合信号。

(2) 构造低通滤波器对混合信号进行滤波,并用 FFT 算法处理得到滤波后信号的频谱图。

代码如下:

```
% chapter5/test2.m
t = 0:0.001:0.5
s1 = sin(40 * pi * t)
s2 = cos(400 * pi * t)
x = s1 + s2           % 生成混合信号
figure(1)             % 分别显示正弦信号、余弦信号、混合信号
subplot(4,1,1)
plot(t,s1)
title('信号 s1')
xlabel('时间/s')
ylabel('幅值')
subplot(4,1,2)
plot(t,s2)
title('信号 s2')
xlabel('时间/s')
ylabel('幅值')
subplot(4,1,3)
plot(t,x)
title('混合信号 x')
xlabel('时间/s')
ylabel('幅值')
y = fft(x,1024)
f = 1000 * (0:512)/1024
subplot(4,1,4)
plot(f,abs(y(1:513)))
title('频谱图')
xlabel('频率/Hz')
ylabel('幅值')
```

程序运行的效果如图 5-8 所示,从频谱图中可以看出两个信号的频率值。

构造巴特沃斯低通滤波器滤除 200Hz 信号,代码如下:

```
fs = 1000;           % 采样频率
[b a] = butter(8,0.1,'low') % 巴特沃斯低通滤波
u = filter(b,a,x)
figure(2)
```

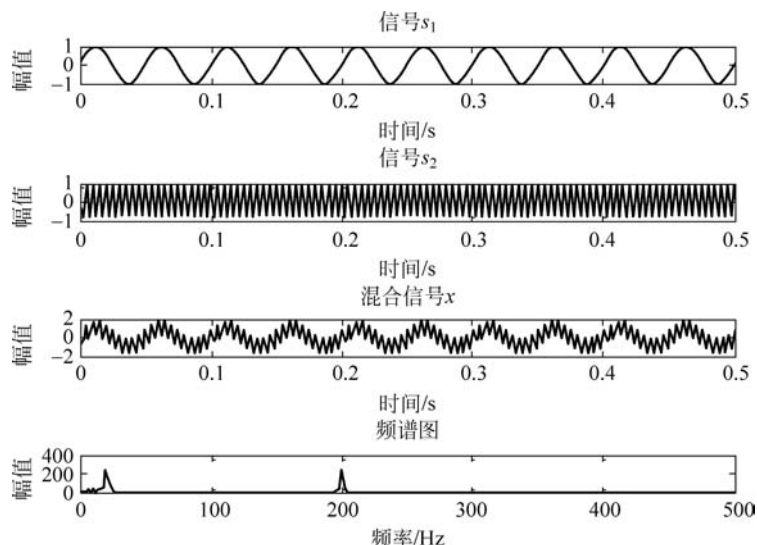



图 5-8 原信号、含噪声信号及频谱

```
subplot(2,1,1)
plot(t,u)           % 滤波后的信号
title('滤波后的信号 x')
xlabel('时间/s')
ylabel('幅值')
y = fft(u,1024)
f = 1000 * (0:512)/1024
subplot(2,1,2)
plot(f,y(1:513))    % 频谱图
title('频谱图')
xlabel('频率/Hz')
ylabel('幅值')
```

提示：butter 函数是求 Butterworth 数字滤波器的系数，在求出系数后对信号进行滤波时用 filter 函数。

$[b,a]=\text{butter}(n,W_n)$ ，根据阶数 n 和截止频率 W_n 计算 Butterworth 滤波器分子和分母系数 (b 为分子系数的向量形式， a 为分母系数的向量形式)， W_n 的确定跟所采样频率 F_s 有关。对于原始信号 x ，设截止频率 $f=50\text{Hz}$ ，因为 $F_s=1000\text{Hz}$ 、 $W_n=f/F_a$ 、 F_a 是分析频率、 $F_a=F_s/2$ ，所以 $W_n=0.1$ 。

程序运行的效果如图 5-9 所示，从波形图和频谱图中可以看到 20Hz 信号被完好地保留了。

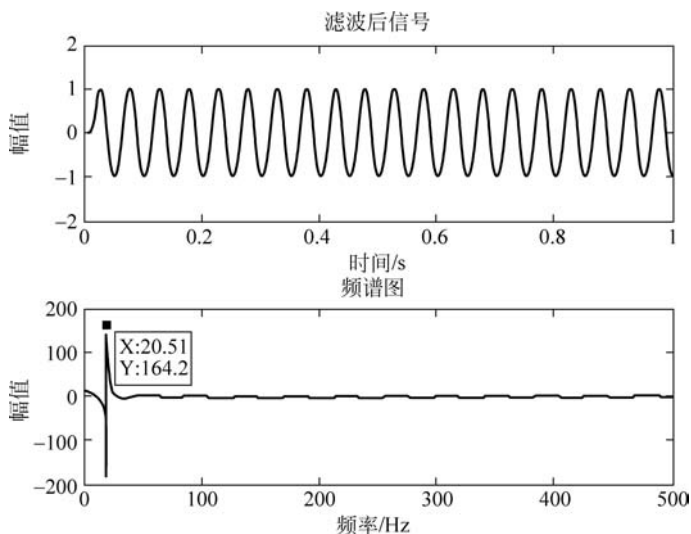


图 5-9 滤波后信号及频谱

5.2 小波变换

传统的信号理论是建立在傅里叶变换的基础上的,而傅里叶变换作为一种全局性的变化,其有一定的局限性,如不具备局部化分析能力、不能分析非平稳信号等。对于这样的非平稳信号,只知道包含哪些频率成分是不够的,还要知道各个成分出现的时间。知道信号频率随时间变化的情况,各个时刻的瞬时频率及其幅值——时频分析。

对于非平稳信号分析一个简单可行的方法就是加窗。把整个时域过程分解成无数个等长的小过程,每个小过程近近平稳,再进行傅里叶变换,这样就知道在哪个时间点上出现了什么频率了。这种方法就是短时傅里叶变换(Short-time Fourier Transform, STFT),但是如果窗太窄,窗内的信号太短,则会导致频率分析不够精准,频率分辨率差;如果窗太宽,时域上又不够精细,则会导致时间分辨率低。

1980年数学家 Y. Meyer 偶然构造出一个真正的小波基,并与 S. Mallat 合作建立了构造小波基的统一方法多尺度分析之后,小波分析才开始蓬勃发展起来,其中比利时女数学家 I. Daubechies 撰写的《小波十讲》对小波的普及起到了重要的推动作用。小波变换是一种新的变换分析方法,它继承和发展了短时傅里叶变换局部化的思想,同时又克服了窗口大小不随频率变化等缺点,能够提供一个随频率改变的时间与频率的窗口,是进行信号时频分析和处理的理想工具。小波变换的出发点和 STFT 还是不同的。STFT 是给信号加窗,分段做 FFT,而小波直接将傅里叶变换的无限长的三角函数换成了有限长的会衰减的小波函数。这样不仅能够获取频率,还可以定位时间。

14min 数 $\psi(t) \in L^2(R)$ 需满足:

$$\int_{-\infty}^{+\infty} \psi(t) dt = 0 \quad (5-11)$$

其中, $\psi(t)$ 为基小波函数, 它可通过尺度的伸缩和时间轴上的平移构成一簇函数系:

$$\psi_{a,b}(t) = |a|^{-1/2} \psi\left(\frac{t-b}{a}\right), \quad a, b \in \mathbf{R}, a \neq 0 \quad (5-12)$$

式中, $\psi_{a,b}(t)$ 为子小波; a 为尺度因子, 反映小波的周期长度; b 为平移因子, 反映时间上的平移。

常用的小波函数有 Haar、Daubechies(dbN)、Mexican Hat(mexh)、Morlet 和 Meyer 共 5 种, 下面分别介绍它们。

1. Haar 小波函数

Haar 小波函数是小波分析中最早用到的一个具有紧支撑的正交小波函数, 也是最简单的一个小波函数, 它是支撑域在 $t \in [0, 1]$ 的单个矩形波。Haar 小波函数的定义如下:

$$\psi(t) = \begin{cases} 1, & 0 \leq t < \frac{1}{2} \\ -1, & \frac{1}{2} \leq t \leq 1 \\ 0, & \text{其他} \end{cases} \quad (5-13)$$

Haar 小波函数在时域上是不连续的, 所以作为基本小波函数性能不是特别好, 但它具有计算简单的优点。Haar 小波函数的时域波形和频域波形如图 5-10 所示。

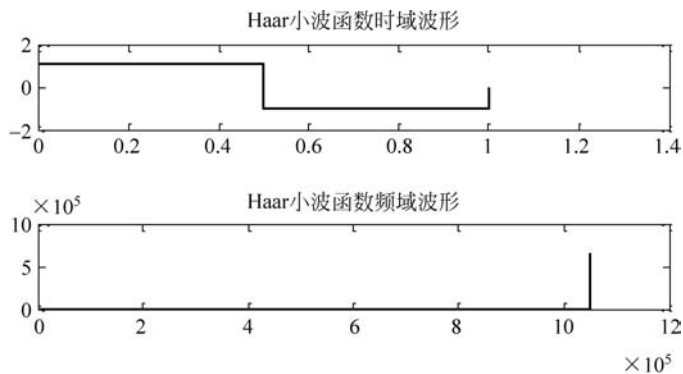


图 5-10 Haar 小波函数的时域波形和频域波形

2. Daubechies(dbN)小波函数

Daubechies 小波函数简写为 dbN, N 是小波的阶数。Daubechies 小波函数常用来分解和重构信号,作为滤波器使用。

小波函数 $\psi(t)$ 和尺度函数 $\phi(t)$ 中的支撑区为 $2N-1$, $\psi(t)$ 的消失矩为 N 。除 $N=1$ (Harr 小波函数)外,dbN 不具有对称性(非线性相位)。dbN 没有明确的表达式。db6 的时域波形和频域波形如图 5-11 所示。

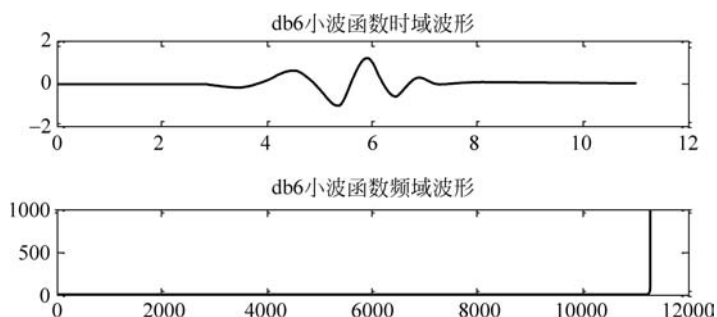


图 5-11 db6 小波函数的时域和频域波形

3. Mexihat 小波函数

Mexihat 小波函数为 Gauss 函数的二阶导数:

$$\psi(t) = (1 - t^2) e^{-\frac{t^2}{2}} \quad (5-14)$$

因为它的形状像墨西哥帽的截面,所以也称为墨西哥帽函数。Mexihat 小波函数的特点是在时间域与频率域都有很好的局部化,不存在尺度函数,所以 Mexihat 小波函数不具有正交性。Mexihat 小波函数的时域波形和频域波形如图 5-12 所示。

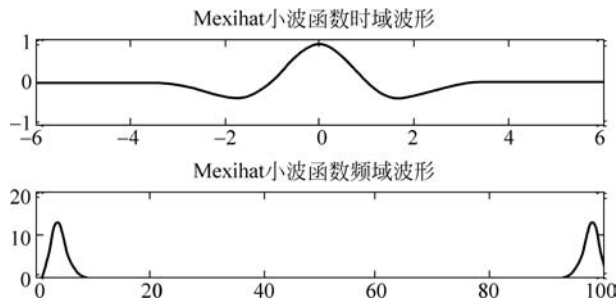


图 5-12 Mexihat 小波函数的时域和频域波形

4. Morlet 小波函数

Morlet 函数如下:

$$\psi(t) = e^{-t^2/2} e^{i\omega_0 t} \quad (5-15)$$

Morlet 小波函数没有尺度函数,而且是非正交分解。Morlet 小波函数的时域波形和频

域波形如图 5-13 所示。

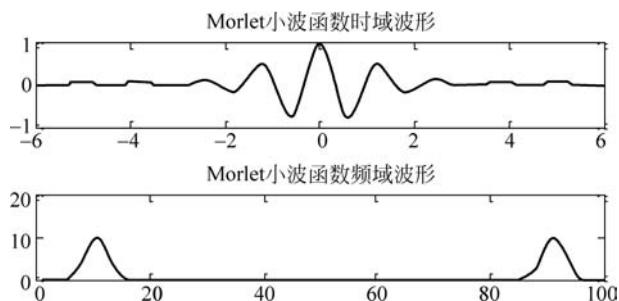


图 5-13 Morlet 小波的时域和频域波形

5. Meyer 小波函数

Meyer 小波函数与尺度函数都是在频域中进行定义的。小波函数定义如下：

$$\phi(\omega) = (2\pi)^{-\frac{1}{2}} e^{i\frac{\omega}{2}} \begin{cases} \sin\left(\frac{\pi}{2}v\left(\frac{3}{2\pi}|\omega| - 1\right)\right), & \frac{2\pi}{3} \leq |\omega| \leq \frac{4\pi}{3} \\ \cos\left(\frac{\pi}{2}v\left(\frac{3}{4\pi}|\omega| - 1\right)\right), & \frac{4\pi}{3} \leq |\omega| \leq \frac{8\pi}{3} \\ 0, & |\omega| \notin \left[\frac{2\pi}{3}, \frac{8\pi}{3}\right] \end{cases} \quad (5-16)$$

Meyer 小波函数的时域波形和频域波形如图 5-14 所示。

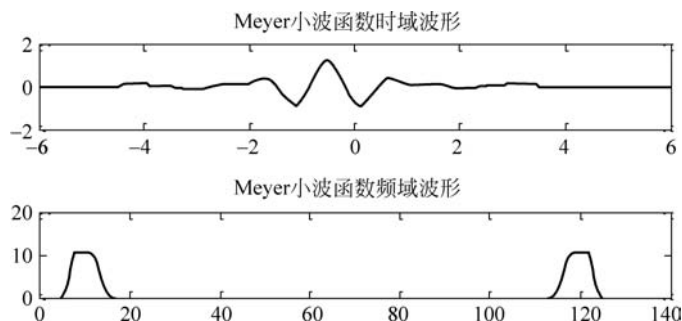


图 5-14 Meyer 小波函数的时域波形和频域波形

选择合适的小波函数是进行小波变换的前提。在实际应用研究中,应针对具体情况选择所需的小波函数;同一信号或时间序列,若选择不同的小波函数,则所得的结果往往会有所差异,有时甚至差异很大。目前,主要通过对比不同小波变换处理信号时所得的结果与理论结果的误差来判定基小波函数的好坏,并由此选定该类研究所需的小波函数。

5.2.2 小波变换理论

若 $\psi_{a,b}(t)$ 是由式(5-12)给出的子小波,对于给定的能量有限信号 $f(t) \in L^2(\mathbb{R})$,其连

续小波变换(Continue Wavelet Transform, CWT)为

$$W_f(a, b) = |a|^{-1/2} \int_R f(t) \bar{\psi}\left(\frac{t-b}{a}\right) dt \quad (5-17)$$

式中, $W_f(a, b)$ 为小波变换系数; $f(t)$ 为一个信号或平方可积函数; a 为伸缩尺度; b 平移参数; $\bar{\psi}\left(\frac{x-b}{a}\right)$ 为 $\psi\left(\frac{x-b}{a}\right)$ 的复共轭函数。

CWT 的变换过程可分成如下 5 个步骤。

第 1 步: 把小波 $\psi(t)$ 和原始信号 $f(t)$ 的开始部分进行比较。

第 2 步: 计算系数 C 。该系数表示该部分信号与小波的近似程度。系数 C 的值越高表示信号与小波越相似, 因此系数 C 可以反映这种波形的相关程度。

第 3 步: 把小波向右移, 距离为 k , 得到的小波函数为 $\psi(t-k)$, 然后重复步骤 1 和步骤 2。

再把小波向右移, 得到小波 $\psi(t-2k)$, 重复步骤 1 和步骤 2。按上述步骤一直进行下去, 直到信号 $f(t)$ 结束。

第 4 步: 扩展小波 $\psi(t)$, 例如开展一倍, 得到的小波函数为 $\psi\left(\frac{t}{2}\right)$ 。

第 5 步: 重复步骤 1~4。

CWT 的整个变换过程如图 5-15 所示。

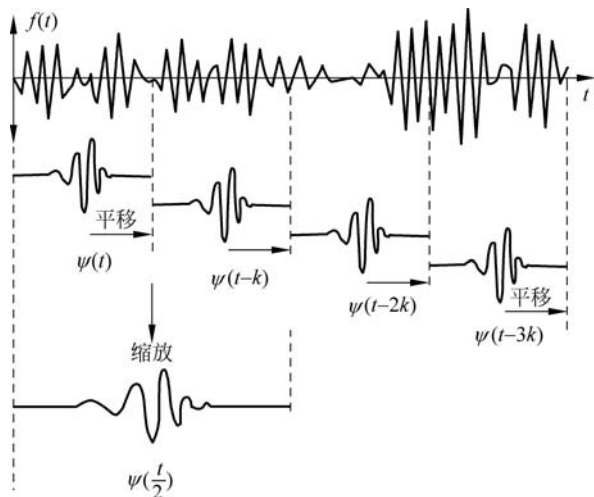


图 5-15 连续小波变换的过程

小波的伸缩尺度 a 与信号频率 ω 之间的关系可以这样来理解。伸缩尺度 a 小, 表示小波比较窄, 度量的是信号细节, 表示频率 ω 比较高; 相反, 伸缩尺度 a 大, 表示小波比较宽, 度量的是信号的粗糙程度, 表示频率 ω 比较低。

从工程中实际得到的信号数据是离散的, 设函数 $f(k\Delta t)$ ($k=1, 2, \dots, N$; Δt 为取样间隔), 则式(5-18)的离散小波变换形式为

$$W_f(a, b) = |a|^{-1/2} \Delta t \sum_{k=1}^N f(k \Delta t) \bar{\psi}\left(\frac{k \Delta t - b}{a}\right) \quad (5-18)$$

由式(5-17)或式(5-18)可知小波变换的基本原理,即通过增加或减小伸缩尺度 a 来得到信号的低频或高频信息,然后分析信号的概貌或细节,实现对信号不同时间尺度和空间局部特征的分析。



5.2.3 小波分解与重构



22min

在计算连续小波变换时,是用离散的数据进行计算的,只是所用的缩放因子和平移参数比较小而已。连续小波变换的计算量是巨大的。为了解决计算量的问题,伸缩尺度和平移参数都选择 2^j ($j > 0$ 的整数)的倍数。使用这样的伸缩尺度和平移参数的小波变换叫作双尺度小波变换,它是离散小波变换(Discrete Wavelet Transform, DWT)的一种形式,离散小波变换通常指的就是双尺度小波变换。

执行离散小波变换的有效方法是使用滤波器。该方法是 Mallat 在 1988 年开发的,叫作 Mallat 算法,这种方法实际上是一种信号的分解方法,在数字信号处理中称为双通道子带编码。用滤波器执行离散小波变换的概念如图 5-16 所示。图中 S 表示原始的输入信号,通过两个互补的滤波器产生 A 和 D 两个信号, A 表示信号的近似值(Approximations), D 表示信号的细节值(Detail)。在许多应用中,信号的低频部分是最重要的,而高频部分起一个“添加剂”的作用。

小波变换中,近似值 A 是由大的伸缩尺度产生的系数,表示信号的低频分量。细节值 D 是由小的伸缩尺度产生的系数,表示信号的高频分量。

在使用滤波器对真实的数字信号进行变换时,得到的数据将是原始数据的两倍。例如,如果原始信号的数据样本为 1000 个,通过滤波之后每个通道的数据均为 1000 个,总共为 2000 个。于是,根据尼奎斯特(Nyquist)采样定理就提出了降采样(Downsampling)的方法,即在每个通道中每两个样本数据取一个,得到的离散小波变换的系数(Coefficient)分别用 cD 和 cA 表示,如图 5-17 所示。

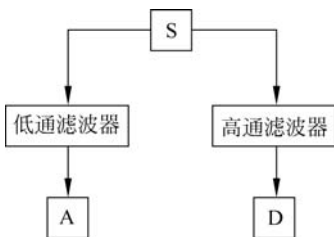


图 5-16 双通道滤波过程

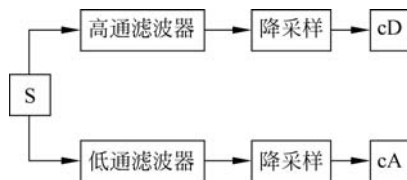


图 5-17 降采样过程

离散小波变换可以被表示成由低通滤波器和高通滤波器组成的一棵树。原始信号通过这样的一对滤波器进行的分解叫作一级分解。信号的分解过程可以迭代,也就是说可进行多级分解。如果对信号的高频分量不再分解,而对低频分量连续进行分解,就可得到许多分

分辨率较低的低频分量,形成如图 5-18 所示的一棵比较大的树。这种树叫作小波分解树 (Wavelet Decomposition Tree)。分解级数的多少取决于要被分析的数据和用户的需要。

小波重构(Wavelet Reconstruction)是把分解的系数还原成原始信号的过程,数学上叫作逆离散小波变换(Inverse Discrete Wavelet Transform, IDWT)。在使用滤波器做小波变换时包含滤波和降采样两个过程,在小波重构时要包含升采样(Upsampling)和滤波过程。小波重构的方法如图 5-19 所示。

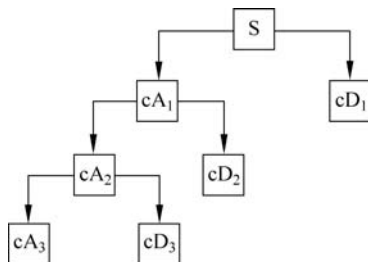


图 5-18 小波分解树

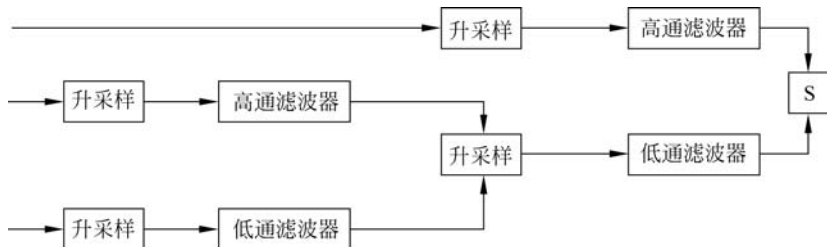


图 5-19 小波重构

在重构的过程中滤波器的选择也是一个重要的研究问题,这是关系到能否重构出满意的原始信号的问题。在信号的分解期间,降采样会引进畸变,这种畸变叫作混叠(Aliasing)。这就需要在分解和重构阶段精心选择关系紧密但不一定一致的滤波器才有可能取消这种混叠。

如果原始信号为 256 点长的超声波信号,采样频率为 25MHz。信号是利用 2.25MHz 的变频器来产生的,因此最主要的频谱分量在 2.25MHz。最后 128 个采样点对应着 [6.25, 12.5] 频率范围的信号。这 128 个数据点的数值均很小,几乎不包含任何信息,因此扔掉这些点不会造成任何重要信息的损失。前面 64 个数据点对应着 [3.12, 6.25] MHz 频率范围的信号,同样幅度很小,没有携带有用的信息。很小的毛刺对应着信号中的高频噪声分量。再前面的 32 点对应着 [1.5, 3.1] MHz 频率范围的信号。信号最主要的能量集中在这 32 个数据点,这与所期望的是一致的。再前面的 16 点对应着 [0.75, 1.5] MHz 频率范围的信号,这层中的峰值可能表示信号的低频包络。再往前的采样点看起来也没有包含什么很重要的信息。于是,只需要保留第 3 层和第 4 层的系数。这也就是说,我们可以仅用 $16 + 32 = 48$ 个数据点来表示这个 256 点长的信号,这大大降低了信号的数据量。

5.2.4 MATLAB 的小波变换函数

1. 连续小波变换函数 cwt

格式如下:

```
COEFS = cwt(x, SCALES, 'wname', 'plot')
```

功能：实现一维连续小波变换,采用 'wname' 小波,在正、实尺度 SCALES 下计算向量一维小波系数,加以图形显示。

2. 离散小波变换函数 dwt

格式如下:

```
[cA, cD] = dwt(x, 'wname')
```

功能：使用小波 'wname' 对信号 x 进行单层分解,求得的近似系数存放在数组 cA 中,细节系数存放在数组 cD 中,ss=idwt(cA,cD,'wname') 可信号重构。

3. 小波分解函数 wavedec

格式如下:

```
[C, L] = wavedec(X, N, 'wname')
```

功能：利用小波 'wname' 对信号 X 进行多层分解;返回的近似系数和细节系数都存放在 C 中,即 C=[cA,cD],L 用于存放近似和各阶细节系数对应的长度。

4. 提取第 N 层近似系数函数 appcoef

格式如下:

```
aN = appcoef(C, L, 'wname', N)
```

功能：利用小波 'wname' 从分解系数 [C,L] 中提取第 N 层近似系数。

5. 提取第 N 层细节系数函数 detcoef

格式如下:

```
dN = detcoef(C, L, N)
```

功能：从分解系数 [C,L] 中提取第 N 层细节系数。

6. 第 N 单支重构函数 wrcoef

格式如下:

```
x = wrcoef('type', C, L, 'wname', N)
```

功能：用小波分解结构 $[C,L]$ 进行第 N 层单支重构,'type'='a'用于重构低频,'type'='d'用于重构高频。

7. 一维信号的小波消噪函数 wden

格式如下：

```
[XD,CXD,LXD] = wden(X,TPTR,SORH,SCAL,N,'wname')
```

功能：函数 wden 用于一维信号的自动消噪。X 为原始信号, $[C,L]$ 为信号的小波分解,N 为小波分解的层数。

TPTR='rigrsure',自适应阈值选择使用 Stein 的无偏风险估计原理。

TPTR='heursure',使用启发式阈值选择。

TPTR='sqtwolog',阈值等于 $\sqrt{2 * \log(\text{length}(X))}$ 。

TPTR='minimaxi',用极大极小原理选择阈值。

SORH 是软阈值或硬阈值的选择(分别对应's'和'h')。

SCAL 指所使用的阈值是否需要重新调整,包含以下 3 种：

* SCAL='one'不调整；

* SCAL='sln'根据第一层的系数进行噪声层的估计,以此来调整阈值。

* SCAL='mln'根据不同的噪声估计来调整阈值。

XD 为消噪后的信号。

$[CXD,LXD]$ 为消噪后信号的小波分解结构。

返回对信号 X 经过 N 层分解后的小波系数进行阈值处理后的消噪信号 XD 和信号 XD 的小波分解结构 $[CXD,LXD]$ 。

5.2.5 小波变换在信号处理中的应用

在信号处理中,小波变换的一个思想是在时间和频率两个方面提供有效的局部化,另一个中心思想是多分辨率,即信号的分解是按照不同分辨率的细节一层一层进行的。也就是小波变换的两种作用——时频分析和多分辨率分析。本节通过两个案例来体会小波变换的作用。

【例 5-3】 利用小波变换对信号进行降噪处理。

源信号和加噪信号,代码如下：

```
% chapter5/test3.m
t = 1:1000;
f = 2 * sin(0.03 * t)
```



```
load noissin % 加载噪声信号
n = noissin
n1 = 0.5 * randn(size(n))
e = f + n1
figure(1)
subplot(3,1,1)
plot(f)
title('源信号')
xlabel('时间/s')
ylabel('幅值')

subplot(3,1,2)
plot(n1)
title('噪声信号')
xlabel('时间/s')
ylabel('幅值')

subplot(3,1,3)
plot(e)
title('加噪后信号')
xlabel('时间/s')
ylabel('幅值')
```

提示：noissin 是 MATLAB 软件中自带噪声的正弦信号。

运行结果如图 5-20 所示。

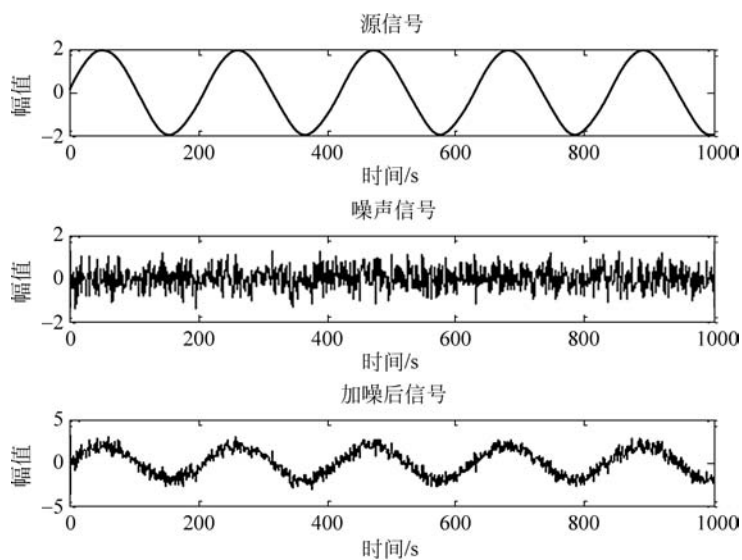


图 5-20 源信号、噪声和含噪声信号

不同阈值小波去噪,代码如下:

```
s1 = wden(e, 'minimaxi', 's', 'one', 5, 'db10') % 用极大极小原理选择阈值
figure(1)
subplot(2,1,1)
plot(s1)
title('极大极小阈值小波降噪后的波形')
xlabel('时间/s')
ylabel('幅值')

s2 = wden(e, 'heursure', 's', 'one', 5, 'sym8') % 使用启发式阈值选择
subplot(2,1,2)
plot(s2)
title('启发式阈值小波降噪后的波形')
xlabel('时间/s')
ylabel('幅值')
```

程序运行结果如图 5-21 所示,从图中可以看出两种阈值选择的小波变换都能很好地将噪声去除,从而恢复原信号。

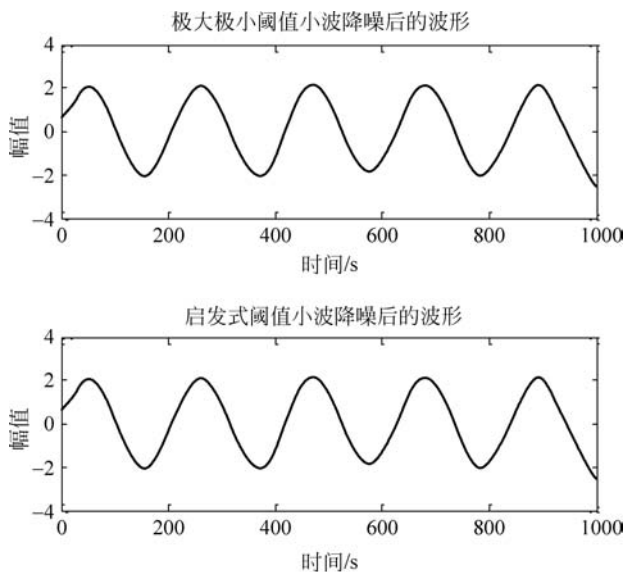


图 5-21 两种小波降噪后的波形

小波分解的近似系数和细节系数,代码如下:

```
figure(3)
[C,L] = wavedec(e,5,'db10') % 小波 5 层分解
```

```

for i = 1:5
a = wrcoef('a',C,L,'db10',6-i) % 小波重构低频
subplot(5,1,i)
plot(a)
ylabel(['a',num2str(6-i)])
end
xlabel('时间/s')
figure(4)
for i = 1:5
d = wrcoef('d',C,L,'db10',6-i) % 小波重构高频
subplot(5,1,i)
plot(d)
ylabel(['d',num2str(6-i)])
end
xlabel('时间/s')

```

运行结果如图 5-22 和图 5-23 所示。

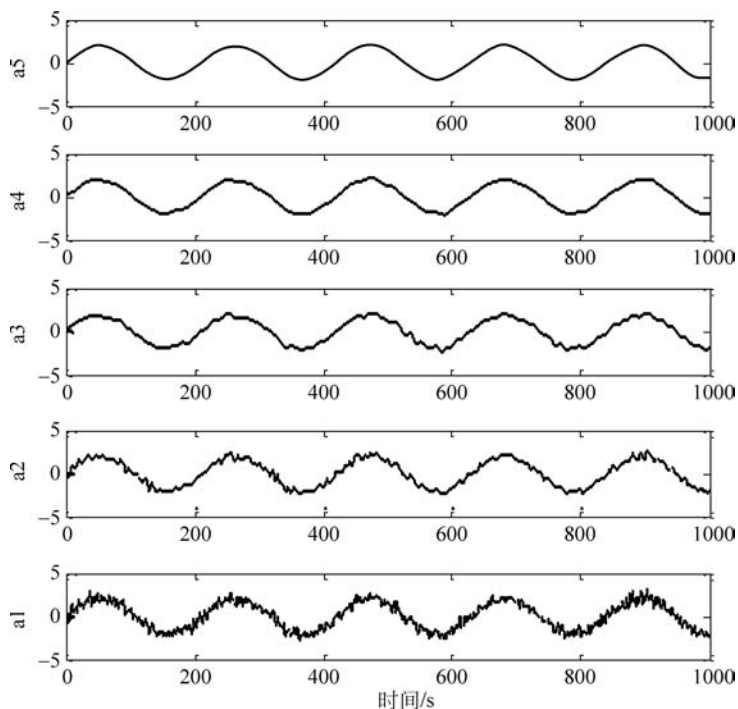


图 5-22 各层的近似系数波形

【例 5-4】 利用小波分解系数检测正弦信号的突变时间起点。
突变信号及其频谱,代码如下:

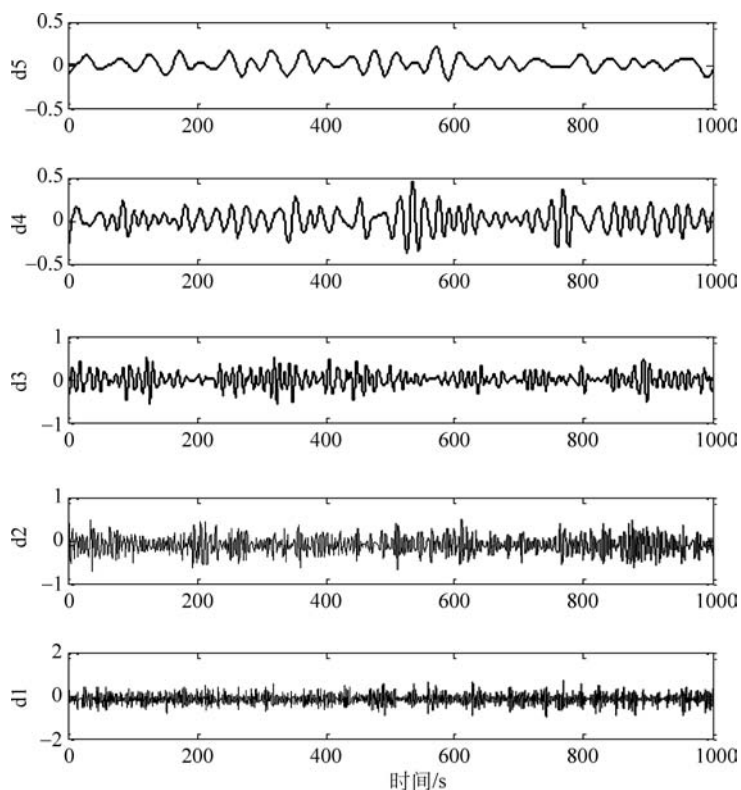


图 5-23 各层的细节系数波形

```
% chapter5/test4.m
t = 0:0.001:1
s1 = sin(40 * pi * t)
s2 = sin(400 * pi * t)
s3 = sin(40 * pi * t)
s = [s1, s2, s3]      % 产生突变信号
figure(1)
subplot(2,1,1)
plot(s)
y = fft(s,1024)      % 求频谱
f = 1000 * (0:512)/1024
subplot(2,1,2)
plot(f,abs(y(1:513)))
```

运行结果如图 5-24 所示。从频谱中可以显示出非突变频谱,但突变部分的频谱不明显,同时也无法分析出突变起止时间点,由此可见,对于非平稳信号,傅里叶变换无法进行分析,这时我们可借助小波变换来分析。

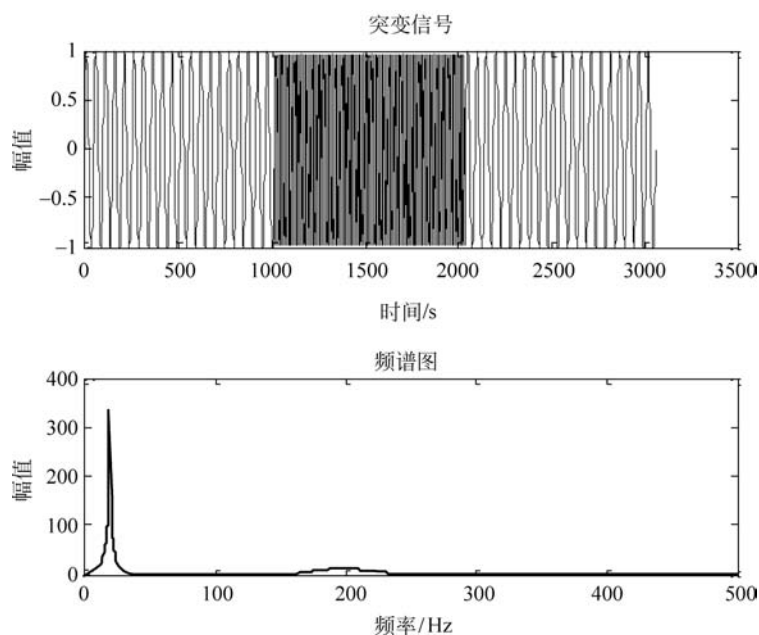


图 5-24 突变信号及其频谱

小波分解的细节系数,代码如下:

```
[c,l] = wavedec(s,7,'sym4')    % 小波分解
figure(2)
for i = 1:7                    % 用 sym4 小波分解 7 层的细节系数
    decmp = wrcoef('d',c,l,'sym4',8-i)
    subplot(7,1,i)
    plot(decmp)
    ylabel(['d',num2str(8-i)])
end
```

提示: symN 小波由 dbN 小波改进而来,在连续性、支集长度、滤波器长度等方面与 dbN 小波一致,但 symN 小波具有更好的对称性,即一定程度上能够减少对信号进行分析和重构时的相位失真。

程序运行结果如图 5-25 所示,从细节系数 d1 和 d2 的波形可以看出信号突变的起止点。

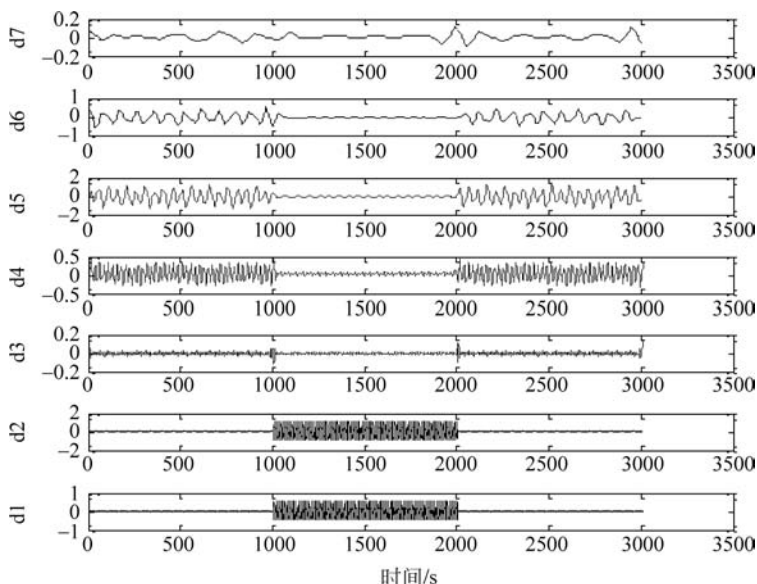


图 5-25 细节系数波形

5.3 小波包变换

小波分析是一种窗口面积固定但其形状可改变的分析方法,即时间和频率窗都可改变的时频局部化分析方法,由于它在分解的过程中只对低频信号再分解,而对高频部分(信号的细节部分)不再继续分解,使它的频率分辨率随频率升高而降低。使小波变换能够很好地表征一大类以低频信息为主要成分的信号,但它不能很好地分解和表示包含大量细节信息(细小边缘或纹理)的信号,如非平稳机械振动信号、遥感图像、地震信号和生物医学信号等。

小波包变换正是对小波变换的进一步优化,在小波变换的基础上,在对每一级信号进行分解时,除了对低频子带进行进一步分解,也对高频子带进行进一步精细分解,所以对包中含大量中、高频信息的信号能够进行更好的时频局部化分析。

5.3.1 小波包变换理论

小波包分解和重构算法的数学表达式如下:

小波包分解是由 $\{d_l^{j,n}\}$ 得到

$$\begin{cases} d_k^{j+1,2n} = \sum_l h_{2l-k} d_l^{j,n} \\ d_k^{j+1,2n+1} = \sum_l g_{2l-k} d_l^{j,n} \end{cases} \quad (5-19)$$

小波包分解是由 $\{d_k^{j+1,2n}\}$ 和 $\{d_k^{j+1,2n+1}\}$ 得到 $\{d_l^{j,n}\}$:



$$d_l^{j,n} = \sum_k h_{l-2k} d_k^{j+1,2n} + \sum_k g_{l-2k} d_k^{j+1,2n+1} \quad (5-20)$$

其中, h, g 为滤波器系数; d 为小波包分解系数; i, t, k 为分解层数; j, n 为小波包节点号。以一个信号的 3 层小波包分解为例, 如图 5-26 所示。

图 5-26 中, 信号 S 分解为 $S_{1,0}$ 和 $S_{1,1}$, $S_{1,0}$ 进一步分解为 $S_{2,0}$ 和 $S_{2,1}$, $S_{1,1}$ 进一步分解为 $S_{2,2}$ 和 $S_{2,3}$, 以此类推, 逐级分解。信号 S 经过 N 层小波包分解之后可以表示:

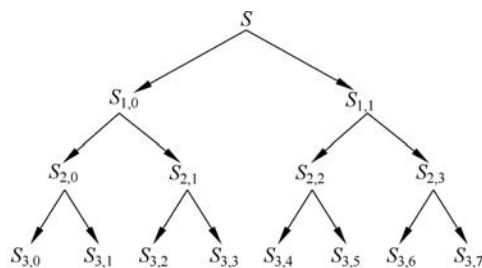


图 5-26 信号的 3 层小波包分解

$$S = S_{N,0} + S_{N,1} + \cdots + S_{N,2^{N-2}} + S_{N,2^{N-1}} \quad (5-21)$$

对比式(5-21)的小波分解可以看出, 小波包分解将每一层所有的分析频带均进行了低频和高频分解, 并依次传至下一层。每一层上的信号分析频带没有交叠现象, 也没有遗漏任何信号。式中 $S_{N,0}, S_{N,1}, \dots, S_{N,2^{N-1}}$ 表示 S 经过 N 层小波包分解后的子频带, 每个子频带信号都有自己的信号特征。对不同特性的信号, 小波包可以构造出相应的“最佳”滤波器组。利用小波包对信号进行重构的优点就在于可根据需要选择全部或部分频段内的信息, 而把其余频段的干扰及噪声信息清零, 同时对信号进行重构, 利于信息数据的挖掘处理。

小波包降噪具体步骤主要包含以下三步。

第 1 步: 首先选择合适的小波基和分解层数, 对信号进行小波包分解。

第 2 步: 其次确定阈值和阈值函数, 对分解得到的小波包系数进行量化处理。

第 3 步: 小波包重构处理后的小波包系数, 获得降噪后的信号。

5.3.2 小波包变换的 MATLAB 函数

1. wpdec 函数

格式如下:

```
[T,D] = wpdec(x,N,'wname',E,P)
```

功能: 实现对信号 x 的一维小波包分解, 返回结构 $[T,D]$ (T 为树结构, D 为数据结构)。

其中 $wname$ 为小波基函数, E 为熵类型, P 是一个可选的参数, 其选择根据 E 的值来决定:

$E = 'shannon'$ 或 $'log\ energy'$, 则 P 不用。

$E = 'threshold'$ 或 $'sure'$, 则 P 是阈值, 并且必须为正数。

$E = 'norm'$, 则 P 是指数, 且有 $1 \leq P < 2$ 。

$E = 'user'$, 则 P 是一个包含 $*.m$ 的文件名字符串, $*.m$ 文件是在一个输入变量 x 下用户自己的熵函数。

2. wpcoef 函数

格式如下：

```
X = wpcoef(T,N)
```

功能：提取一维小波包变换的系数,其返回与小波包树 T 结构 N 层对应的系数。

【例 5-5】 计算小波包分解的相应系数。

代码如下：

```
% chapter5/test5.m
f0 = 100; % 信号频率
fs = 500; % 采样率
Ts = 1/fs;
n = 1:1:100;
N = length(n); % 得到序列的长度
y2 = sin(2 * pi * f0 * n * Ts);
y1 = cos(6 * pi * f0 * n * Ts)
% 混合信号
y = y1 + y2 + 0.4 * randn(1,100);
wpt = wpdec(y,3,'db1')
cfs = wpcoef(wpt,[2,1])
figure(1)
plot(wpt)
figure(2)
subplot(2,1,1)
plot(y)
title('原始信号')
subplot(2,1,2)
plot(cfs)
title('小波包系数 s[2,1]')
```

运行结果如图 5-27 及图 5-28 所示。

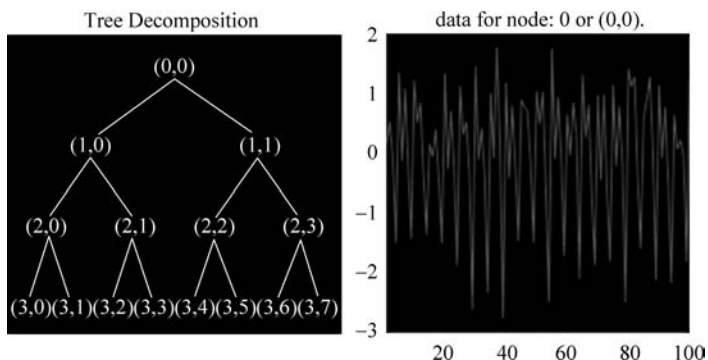
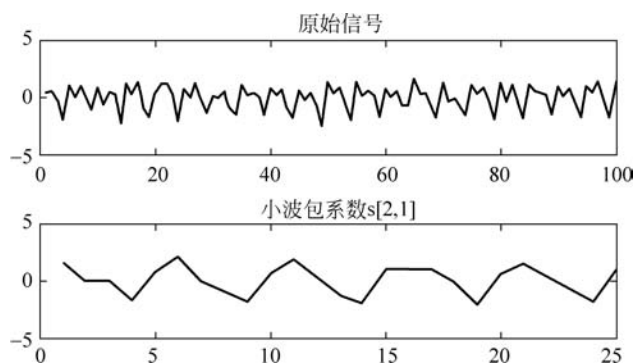


图 5-27 小波包树及原始信号

图 5-28 原始信号及小波包系数 $s[2,1]$

3. wpsplt 函数

格式如下：

```
[T, cA, cD] = wpsplt(T, N)
```

功能：返回节点的系数，其中，cA 为节点 N 的低频系数，cD 为节点 D 的高频系数。

4. wprcoef 函数

格式如下：

```
x = wprcoef(T, N)
```

功能：用于重构节点 N 的小波系数。

5. wprec 函数

格式如下：

```
x = wprec(T)
```

功能：用于一维小波包分解的重构。

5.3.3 小波包变换在信号处理中的应用

【例 5-6】 利用小波包分解法对碘钨灯管电弧数据进行处理。

数据显示代码如下：

```
% chapter5/test6.m
fid = fopen('G:\电弧数据\碘钨灯管\碘钨灯管\817.bin', 'r')
```

```
[a,count] = fread(fid,'int16');
e = a(500:3080)
plot(e)
```

运行结果如图 5-29 所示。

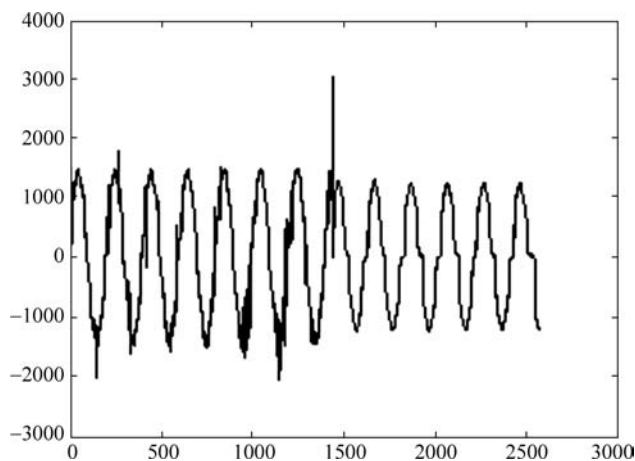


图 5-29 碘钨灯管电弧数据图

对数据进行截取处理,其中 500~2000 点为正常波形,2001~3000 点为电弧故障波形,分别求小波包,并求各系数能量。

小波包分解的近似部分,代码如下:

```
% db10 小波包进行 4 层分解
wpt = wpdec(e,4,'db6')
% 画出小波树节点的小波分解系数
c40 = wpcoef(wpt,[4 0])
c41 = wpcoef(wpt,[4 1])
c42 = wpcoef(wpt,[4 2])
c43 = wpcoef(wpt,[4 3])
c44 = wpcoef(wpt,[4 4])
c45 = wpcoef(wpt,[4 5])
c46 = wpcoef(wpt,[4 6])
c47 = wpcoef(wpt,[4 7])
c48 = wpcoef(wpt,[4 8])
c49 = wpcoef(wpt,[4 9])
c410 = wpcoef(wpt,[4 10])
c411 = wpcoef(wpt,[4 11])
c412 = wpcoef(wpt,[4 12])
c413 = wpcoef(wpt,[4 13])
c414 = wpcoef(wpt,[4 14])
c415 = wpcoef(wpt,[4 15])
```

```

figure(2)
subplot(8,1,1)
plot(c40)
subplot(8,1,2)
plot(c41)
subplot(8,1,3)
plot(c42)
subplot(8,1,4)
plot(c43)
subplot(8,1,5)
plot(c44)
subplot(8,1,6)
plot(c45)
subplot(8,1,7)
plot(c46)
subplot(8,1,8)
plot(c47)
[T] = wpdec(e,4,'db6');      % 小波包分解
for i = 1:16
y = wprcoef(T,i);           % 重构所有层节点的小波包系数
E(i) = wenergy(y,i);         % 求小波包能量
end
figure(4)
bar(E)

```

运行结果如图 5-30 所示。

小波包分解的细节信号,代码如下:

```

figure(3)
subplot(8,1,1)
plot(c48)
subplot(8,1,2)
plot(c49)
subplot(8,1,3)
plot(c410)
subplot(8,1,4)
plot(c411)
subplot(8,1,5)
plot(c412)
subplot(8,1,6)
plot(c413)
subplot(8,1,7)
plot(c414)
subplot(8,1,8)
plot(c415)

```

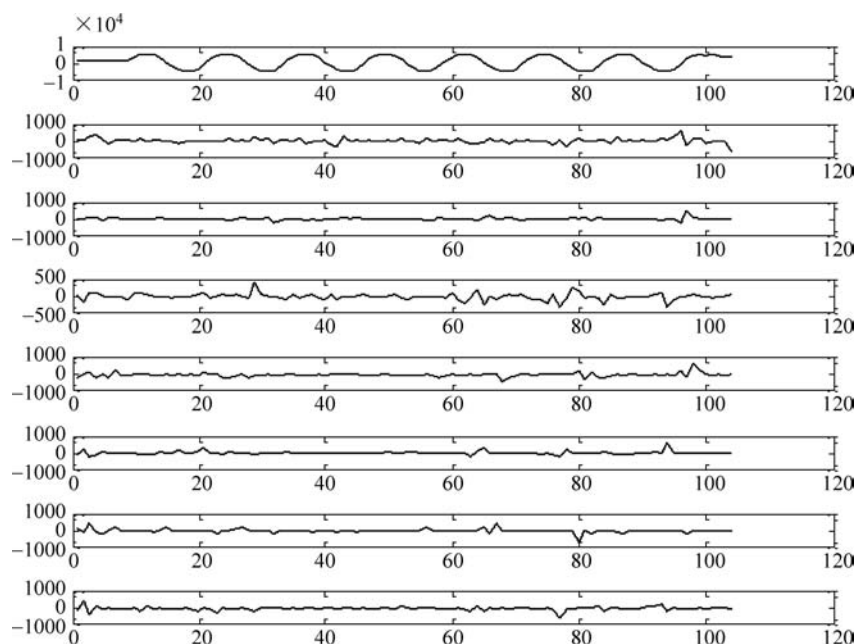


图 5-30 小波包分解的近似部分信号

运行结果如图 5-31 所示。

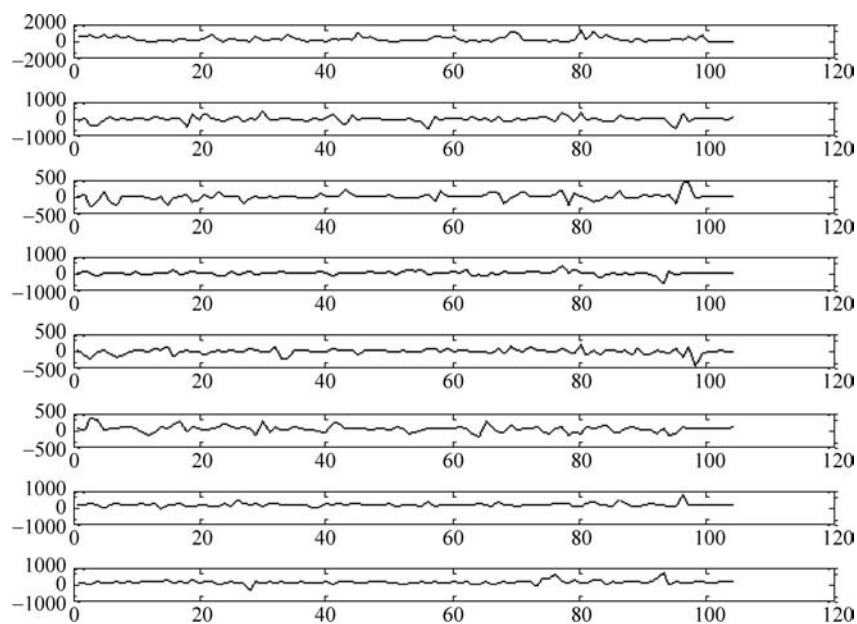


图 5-31 小波包分解的细节信号

小波包分解正常和故障能量值,代码如下:

```
[T] = wpdec(e,4,'db6');      % 小波包分解
for i = 1:16                  %
y = wprcoef(T,i);            % 重构所有层节点的小波包系数
E(i) = wenergy(y,i);          % 求小波包能量
end
figure(4)
bar(E)
```

运行结果如图 5-32 和图 5-33 所示,从能量值图可以看出,正常信号和故障信号能量值区分很明显。

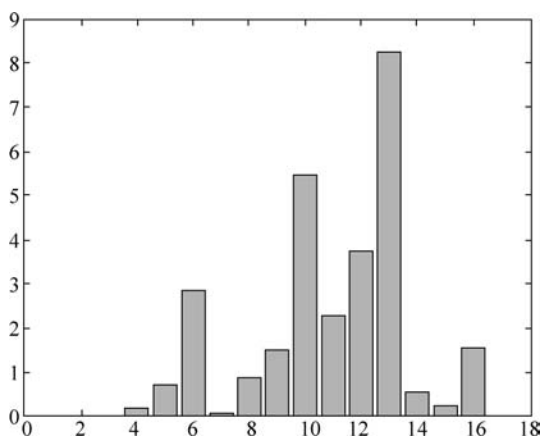


图 5-32 正常能量值

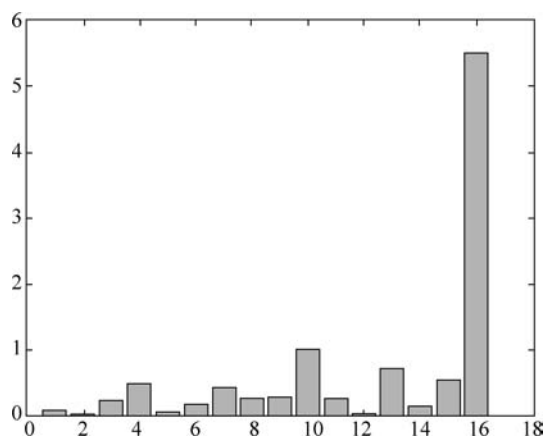


图 5-33 故障能量值

小波包在图像处理方面有广泛的应用,例如图像边缘检测等。小波包分解后得到的图像序列由近似部分和细节部分组成,近似部分是源图像对高频部分进行滤波所得的近似表示。经滤波后,近似部分去除了高频分量,因此能够检测到源图像中所检测不到的边缘。对近似图像进行边缘检测的结果和直接对原始图像进行边缘检测的结果相比,前一种方法的效果更好。

【例 5-7】 利用小波包分解法实现二维图像的边缘检测。

代码如下:

```
% chapter5/test10.m
x = imread('G:\3.jpg');
y1 = rgb2gray(x)                % 灰度化,将彩色图像转化为黑白
x1 = double(y1) + 20 * randn(size(y1)) % 图像添加均值为 0,标准差为 10,方差为 100 的高斯白噪声
subplot(2,2,1)
image(x1)
title('加噪后图像')
```

```

T = wpdec2(x1, 3, 'db4')           % 二维小波包函数
A = wprcoef(T, [1, 0])
subplot(2, 2, 2)
image(A)
title('图像近似部分 ')
by1 = edge(x1, 'prewitt')           % 边缘检测函数
edge(x1, 'canny', [0.04, 0.1], 3)
subplot(2, 2, 3)
imshow(by1)
title('源图像的边缘 ')
by2 = edge(A, 'prewitt')
subplot(2, 2, 4)
imshow(by2)
title('图像近似部分的边缘 ')

```

运行结果如图 5-34 所示,从图中可以看出,经过小波包去噪后的图像边缘效果要比未去噪的图像边缘效果好很多。

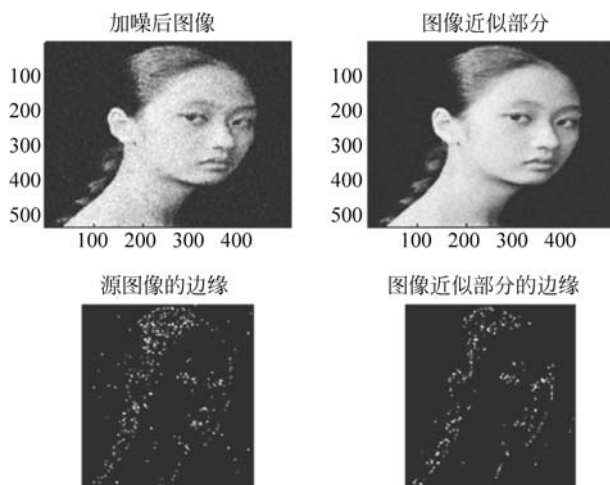


图 5-34 小波包边缘检测