5　基于 MATLAB 的信号频谱分析

5.1　快速傅里叶变换（FFT）

按照被变换的输入信号类型不同，傅立叶变换可以分为 4 种类型：

- 非周期性连续信号傅立叶变换（Fourier Transform）
- 周期性连续信号傅立叶级数（Fourier Series）
- 非周期性离散信号离散时域傅立叶变换（Discrete Time Fourier Transform）
- 周期性离散信号离散傅立叶变换（Discrete Fourier Transform，DFT）

因为计算机只能处理离散的和有限长度的数值信号，对于连续信号要先离散化。对于离散信号的变换只有离散傅立叶变换（DFT）才能被适用，对于其它的变换类型只有在数学演算中才能用到。

快速傅里叶变换（Fast Fourier Transform，FFT）是 DFT 的一种快速算法。DFT 运算过程为

$$X(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)\mathrm{e}^{-\mathrm{j}\pi nt/N} \tag{5.1}$$

可见，在计算机上进行的 DFT，使用的输入值是经过 ADC（Analog-to-Digital Conversion）后采集到的采样值，也就是时域的信号值，输入采样点的数量决定了转换的计算规模。**变换后的频谱输出包含同样数量的采样点，但是其中有一半的值是冗余的，通常不会显示在频谱中，所以真正有用的信息是 $N/2+1$ 个点。**

FFT 是 1965 年由库力（James W. Cooley，1926—2016）和图基（John W. Tukey，1915—2000）提出的，采用这种算法能使计算机计算离散傅里叶变换所需要的乘法次数大为减少，特别是被变换的抽样点数 N 越多，FFT 算法计算量的节省就越显著。

5.2　MATLAB 中 FFT 的使用方法

5.2.1　语法说明

```
Y = fft(X)
```

说明：用快速傅里叶变换（FFT）算法计算 X 的离散傅里叶变换（DFT）。

- 如果 X 是向量，则 `fft(X)` 返回该向量的傅里叶变换。
- 如果 X 是矩阵，则 `fft(X)` 将 X 的各列视为向量，并返回每列的傅里叶变换。
- 如果 X 是一个多维数组，则 `fft(X)` 将沿大小不等于 1 的第一个数组维度的值视为向量，并返回每个向量的傅里叶变换。

```
Y = fft(X,n)
```

说明：返回 n 点 DFT。如果未指定任何值，则 Y 的大小与 X 相同。

- 如果 X 是向量且 X 的长度小于 n，则为 X 补上尾零以达到长度 n。
- 如果 X 是向量且 X 的长度大于 n，则对 X 进行截断以达到长度 n。
- 如果 X 是矩阵，则每列的处理与在向量情况下相同。
- 如果 X 为多维数组，则大小不等于 1 的第一个数组维度的处理与在向量情况下相同。

5.2.2 示例

下面的示例说明如何使用 FFT 函数进行频谱分析。FFT 的一个常用场景是确定一个时域含噪信号的频率分量。

指定信号的参数，采样频率为 1 kHz，信号持续时间为 1.5 秒。

```
Fs = 1000;          % 采样频率
T = 1/Fs;           % 采样周期
N = 1500;           % 信号长度
t = (0:N-1)*T;      % 时间轴，从t=0至1.5，步长为0.001，共N个采样点
```

构造一个信号，其中包含幅值为 0.7 的 50 Hz 正弦量和幅值为 1 的 120 Hz 正弦量。

```
S = 0.7*sin(2*pi*50*t) + sin(2*pi*120*t);
```

在时域中绘制这个信号，如图 5.1 所示。

```
plot(t(1:250),S(1:250))          % 绘图，横轴t、纵轴S，绘制前250个点
xlabel('Time (s)')               % 添加横轴标签
ylabel('Amplitude')              % 添加纵轴标签
title('Time Domain Signal')      % 添加标题
```

用均值为零、方差为 4 的白噪声扰乱该信号。

```
X = S + 2*randn(size(t));
% randn(size(t))是返回一个和t有同样维数大小的随机数组
```

在时域中绘制含噪信号，如图 5.2 所示。通过查看信号 X(t) 很难确定频率分量。

```
plot(t(1:250),X(1:250))
xlabel('Time (s)')
ylabel('Amplitude')
title('Signal Corrupted with Zero-Mean Random Noise')
```

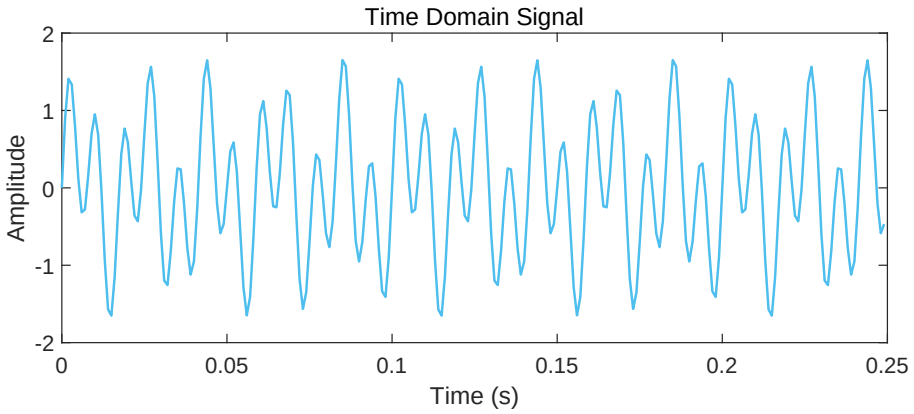

图 5.1　50 Hz 和 120 Hz 两个频率正弦波叠加的信号

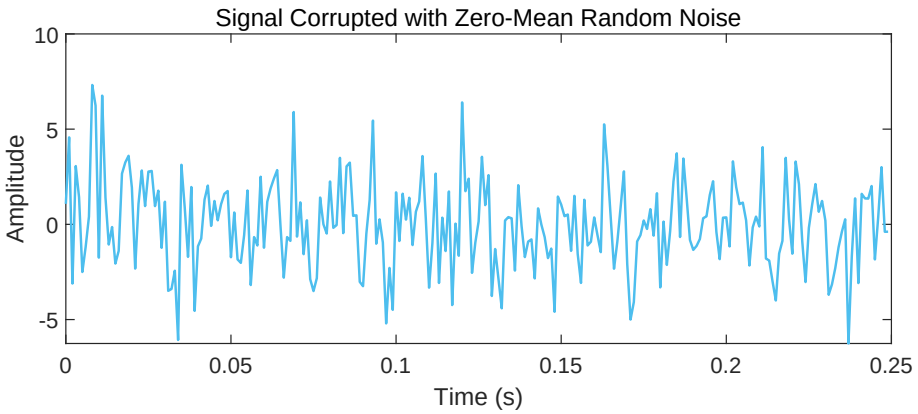

图 5.2　含噪信号

要想得到含噪信号 y 的离散傅里叶变换很容易，执行快速傅里叶变换 (FFT) 即可实现。

```
Y = fft(X);          % 变换后的频谱输出包含同样数量的采样点
```

计算双侧频谱 P2。然后基于 P2 和偶数信号长度 N 计算单侧频谱 P1。这是由于变换后的频谱输出有一半的值是冗余的，真正有用的信息是 N/2+1 个点。

```
P2 = abs(Y/N);                    % 计算谱密度
P1 = P2(1:N/2+1);                 % 计算单侧频谱
P1(2:end-1) = 2*P1(2:end-1);      % 双侧频谱是对称的，变单侧后，除两端外幅
                                  值加倍
```

定义频域 f 并绘制单侧幅值频谱 P1，如图 5.3 所示。与预期相符，由于增加了噪声，幅值并不精确等于 0.7 和 1。一般情况下，较长的信号会产生更好的频率近似值。

```
f = Fs*(0:(N/2))/N;               % Fs为采样频率，N为采样点数
plot(f,P1)
title('Single-Sided Amplitude Spectrum of X(t)')
xlabel('frequency (Hz)')
ylabel('|P1(f)|')
```

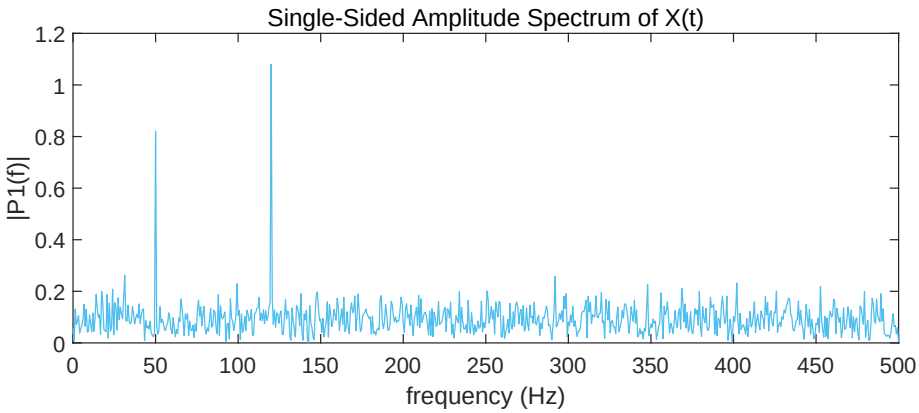

图 5.3　含噪信号的频谱

采用原始的、未加干扰的信号 S(t) 进行傅里叶变换，可获得精确的幅值 0.7 和 1.0，如图 5.4 所示。

```
Y = fft(S);
P2 = abs(Y/N);
P1 = P2(1:N/2+1);
P1(2:end-1) = 2*P1(2:end-1);
f = Fs*(0:(N/2))/N;
plot(f,P1)
title('Single-Sided Amplitude Spectrum of X(t)')
xlabel('frequency (Hz)')
ylabel('|P1(f)|')
```

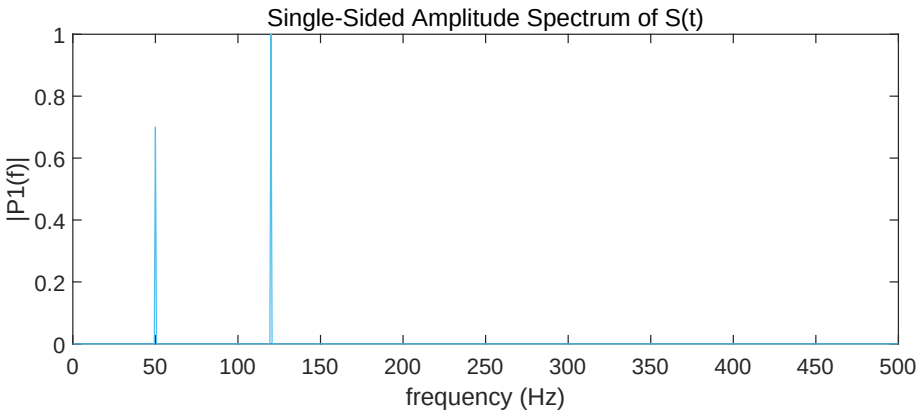

图 5.4　无噪信号的频谱

5.3　快速傅里叶逆变换（iFFT）

5.3.1　语法说明

```
X = ifft(Y)
```

说明：使用快速傅里叶变换算法计算 Y 的逆离散傅里叶变换。X 与 Y 的大小相同。

- 如果 Y 是向量，则 `ifft(Y)` 返回该向量的逆变换。
- 如果 Y 是矩阵，则 `ifft(Y)` 返回该矩阵每一列的逆变换。
- 如果 Y 是多维数组，则 `ifft(Y)` 将大小不等于 1 的第一个维度上的值视为向量，并返回每个向量的逆变换。

```
X = ifft(Y,n)
```

说明：通过用尾随零填充 Y 以达到长度 n，返回 Y 的 n 点傅里叶逆变换。

5.3.2　示例

时空采样数据与频率采样数据间的傅里叶变换及其逆变换。创建一个向量并计算其傅里叶变换。

```
X = [1 2 3 4 5];
Y = fft(X)
```

```
Y = 1×5 complex
15.0000+0.0000i    -2.5000+3.4410i    -2.5000+0.8123i
-2.5000-0.8123i    -2.5000-3.4410i
```

计算 Y 的逆变换，结果与原始向量 X 相同。

```
ifft(Y)
```

```
ans = 1×5
1     2     3     4     5
```

前面的例子中，无噪声信号 S(t)（图 5.1）的频谱如图 5.4 所示，有两个频率。现在我们对 Y 进行傅里叶逆变换，得到信号 W(t)，并绘出曲线，如图 5.5 所示。可见，图 5.5 和图 5.1 基本一致。

```
W = ifft(Y);           % Inverse Fast Fourier transform
plot(t(1:250),W(1:250))
xlabel('Time (s)')
ylabel('W(t)')
title('Inverse Fast Fourier transform')
```

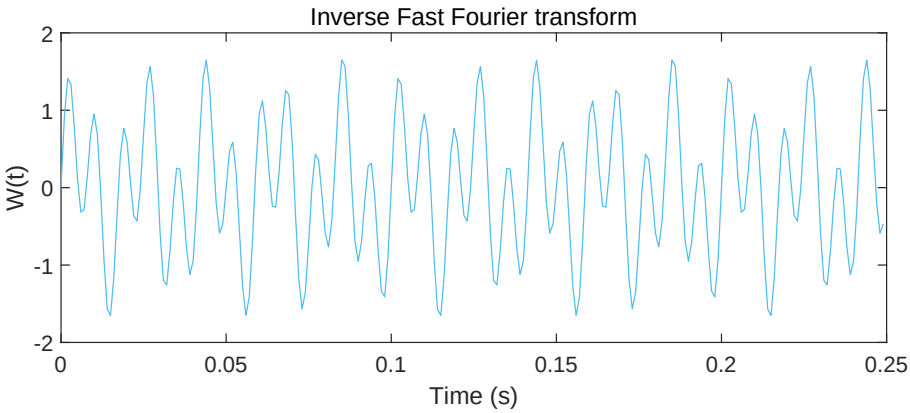

图 5.5　傅里叶逆变换后得到的信号

如果我们对频谱作些处理，去掉频率小于 100 Hz 的值。然后再做 iFFT，则信号变为如图 5.6 所示，只剩下 120 Hz 的正弦信号。

频谱范围是 0 ~ 500 Hz（见图 5.4），1500 个点。由于变换后的频谱输出有一半的值是冗余的，真正有用的信息是 $N/2+1$ 个点，即 751 个点。100 Hz 约为第 151 个点处（$751/500\times100$）。

```
Y(1:151)=0;                          % 频谱中去掉小于100Hz的部分
Y(1350:1500)=0;                      % 由于Y是对称的，两端都要去掉
W = ifft(Y);                         % Inverse Fast Fourier transform
W = real(W);                         % 取实部
plot(t(1:250),W(1:250))              % 绘图，横轴t、纵轴W
xlabel('Time (s)')                   % 添加横轴标签
ylabel('W(t)')                       % 添加纵轴标签
title('After Signal Processing')     % 添加标题
```

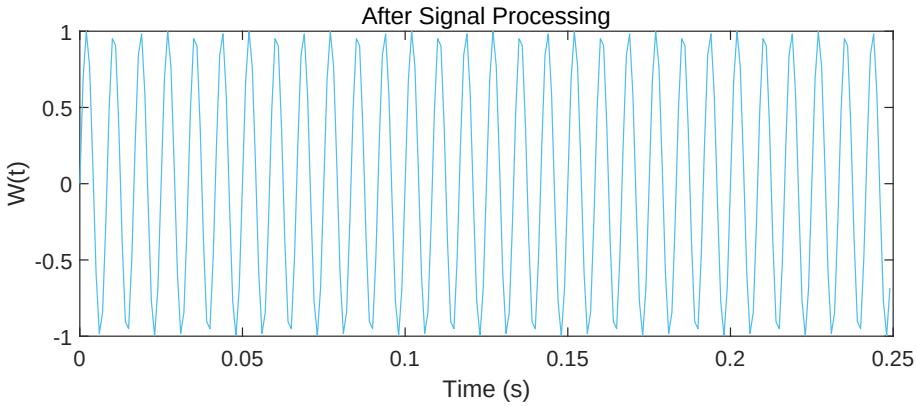

图 5.6　经频谱处理后得到的信号

5.4　声音信号频谱

本节介绍如何借助 MATLAB 分析声音信号的频谱，结合对应的时域波形图，再听一听声音，可以更直观地理解频谱的概念。

5.4.1　常用函数

```
[y,fs]=audioread('filename')
```

说明：从名为 `filename` 的文件中读取数据，并返回样本数据 `y` 以及该数据的采样率 `Fs`。`y` 一般是两列（立体声音）。

```
sound(y,Fs)
```

说明：以采样率 `Fs` 向扬声器发送音频信号 `y`。

5.4.2　示例

选择一段语音，首先用 audioread 函数读出文件中的数据，并用 sound 函数播放。

```
clear;                                  % 从工作区中删除项目、释放系统内存
clc;                                    % 清空命令行窗口
[y,fs]=audioread('sywj.wav');           % 将声音文件数据读入matlab
info=audioinfo('sywj.wav')              % 得到声音的格式、位数、频率等信息
sound(y,fs)                             % 将信号数据矩阵转换为声音（播放声音）
```

画出时域图，如图 5.7 所示。

```
T=1/fs;                                 % 采样时间
N=length(y);                            % 信号长度
t=(0:N-1)*T;                            % 时间
yleft=y(:,1);                           % 取左声道
figure(1);                              % 创建图窗窗口，Figure 1
plot(t,yleft);                          % 输入信号时域曲线
axis([0,t(N),-1.1,1.1]);                % 设置坐标轴范围
title('Time Domain Signal');
xlabel('Time (s)');
ylabel('Amplitude');
```

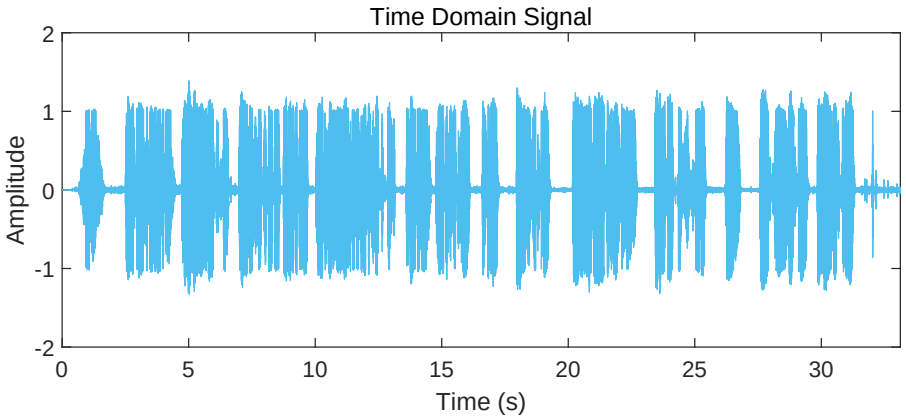

图 5.7　一段声音的时域图

然后对声音数据做 FFT，画出幅度频谱。

```
Yf=fft(yleft);                          % 对N点进行傅里叶变换到频域
P2 = abs(Yf/N);                         % 计算谱密度
P1 = P2(1:N/2+1);                       % 计算单侧频谱
P1(2:end-1) = 2*P1(2:end-1);            % 双侧频谱变单侧后，除两端外幅值加倍
```

```
f = fs*(0:(N/2))/N;                    % fs为采样频率，N为采样点数
figure(2)                              % 创建图窗窗口，Figure 2
plot(f,P1);                            % 左声道频谱图
axis([0,7000,0,0.025]);                % 设置坐标轴范围
title('Single-Sided Amplitude Spectrum');
xlabel('frequency (Hz)');
ylabel('H(f)');
grid on                                % 显示坐标区网格线
```

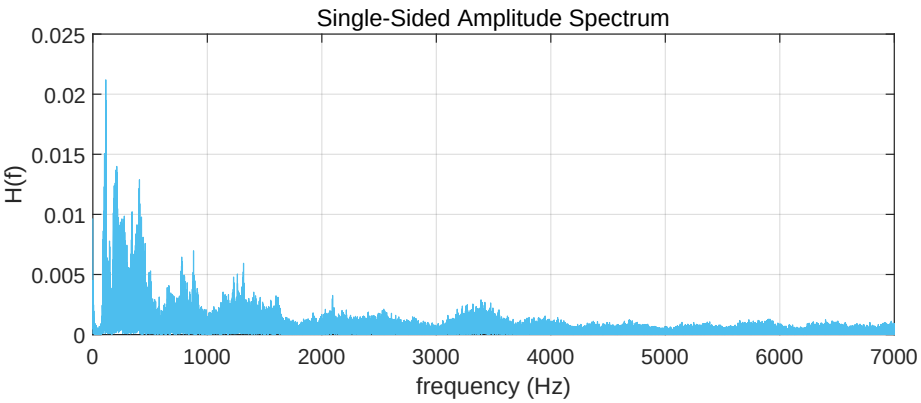

图 **5.8**　一段声音的频域图