

浙江大学实验报告

姓名：胡东东 专业：信息工程 学号：3240102873

课程名称：信息与电子工程导论 任课老师：回晓楠

实验名称：基于 MATLAB 的信号频谱分析 实验日期：2025 年 12 月 1 日

1 实验目的和要求

1.1 实验目的

- 学会使用 MATLAB 的 FFT 与 iFFT 对信号进行频谱分析与重构。
- 通过对比验证傅里叶变换在时域与频域转换中的准确性与可逆性。
- 理解随机噪声对信号频谱的影响，并掌握通过频域滤波进行信号去噪的思路与方法。
- 观察并理解采样频率与采样点数对频谱分析结果的影响。

1.2 实验要求

- 构建信号：使用 MATLAB 生成一个包含至少三个频率分量的周期信号。同时，构造一个在周期信号基础上叠加随机噪声的非周期信号，模拟实际信号传输中的干扰情形。
- 频谱分析：利用 MATLAB 中的快速傅里叶变换（FFT）分别对上述周期信号和非周期信号进行频谱分析，获取并观察其频域特性，包括各频率成分的幅值与位置。
- 频谱处理与重构：对周期信号的频谱进行傅里叶逆变换（iFFT），验证变换后时域信号与原始信号的一致性；对含噪非周期信号的频谱进行滤波处理，尝试去除或抑制噪声频段；将滤波后的频谱进行 iFFT 重构，观察并对比处理前后时域波形的变化。
- 分析参数影响：通过调整采样频率与采样点数，观察其对频谱分辨率与信号重构效果的影响，并简要分析原因。

2 实验原理

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、周期性连续信号傅立叶级数、非周期性离散信号离散时域傅立叶变换、周期性离散信号离散傅立叶变换。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。

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

算过程为： $X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n)e^{-j\pi nl/N}$ 。在计算机上进行的 DFT，使用的输入值是经过 ADC 后采集到的采样值。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。

3 实验内容

对无噪声信号执行 FFT 和 iFFT:

(1) 构建一个含有 3 个频率分量的周期信号，其中包含幅值为 0.7V 的 50Hz 正弦量、幅值为 1.2V 的 120Hz 正弦量和幅值为 1.6V 的 150Hz 正弦量。指定信号的参数，设置采样频率为 1kHz，信号持续时间为 1.5 秒，并在时域中绘制这个信号，如图 1 所示；

(2) 执行快速傅里叶变换（FFT）对原始的、未加干扰的信号 $S(t)$ 进行傅里叶变换，得到其频谱图，如图 2 所示；

(3) 对无噪声信号 $S(t)$ 的频谱进行傅里叶逆变换（iFFT），得到其时域信号 $W(t)$ ，绘制曲线如图 3 所示，并与原始信号 $S(t)$ 进行比较；

对含噪信号执行 FFT，滤波后执行 iFFT:

(4) 引入均值为零、方差为 4 的白噪声扰乱原始的无噪声信号 $S(t)$ ，得到含噪信号 $X(t)$ 并在时域中绘制这个信号，如图 4 所示；

(5) 执行快速傅里叶变换（FFT）对含噪信号 $X(t)$ 进行傅里叶变换，得到其频谱图，如图 5 所示；

(6) 对含噪信号 $X(t)$ 的频谱进行滤波处理，去掉频率大于 200Hz 的值，得到去除部分噪声的信号的频谱，如图 6 所示。然后再对滤波后的信号频谱做傅里叶逆变换（iFFT），得到时域信号如图 7 所示，观察去除效果；

利用含噪信号研究采样频率、采样点数的影响:

(7) 将采样频率由 1000Hz 改为 500Hz，对比观察改动前后含噪信号的频谱；

(8) 将采样点数由 1500 改为 500，对比观察改动前后含噪信号的频谱；

4 实验结果和分析

(1) 对无噪声信号执行 FFT 和 iFFT:

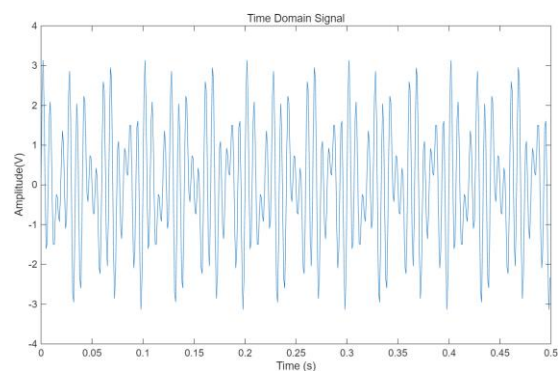


图 1 三个频率正弦波叠加的原始无噪声信号

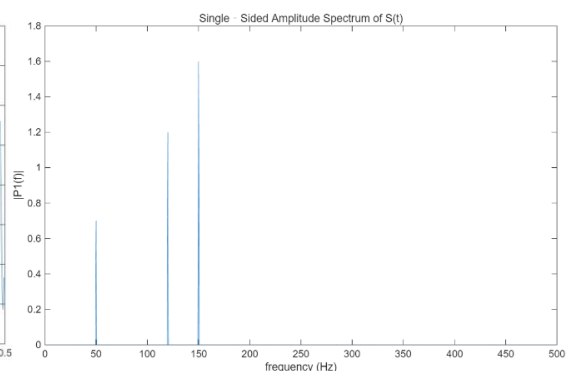


图 2 无噪声信号的频谱

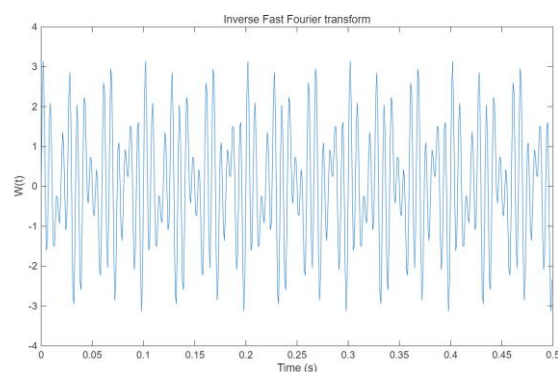


图 3 傅里叶逆变换后得到的无噪声信号

对比经傅里叶逆变换重构的信号时域图与原始无噪声信号的时域图，发现两者波形几乎完全吻合。这表明，利用 FFT 和 iFFT 这一对变换，能够在不丢失主要信息的前提下，高保真地实现信号在时域与频域之间的相互转换。

(2) 对含噪信号执行 FFT，滤波后执行 iFFT:

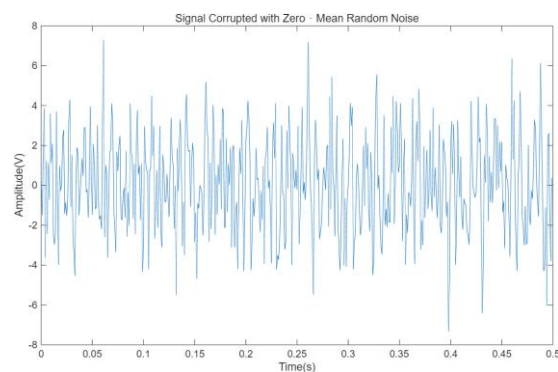


图 4 利用白噪声干扰原始信号得到的含噪信号

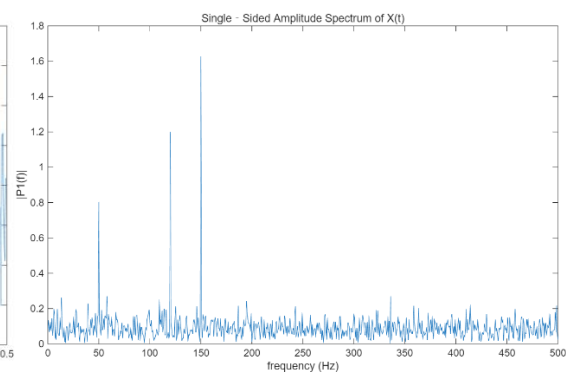


图 5 含噪信号的频谱

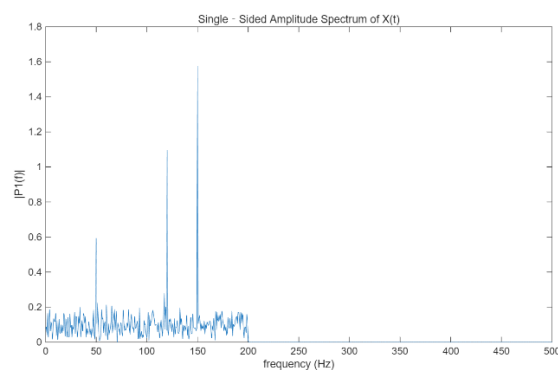


图 6 经滤波后的含噪信号频谱

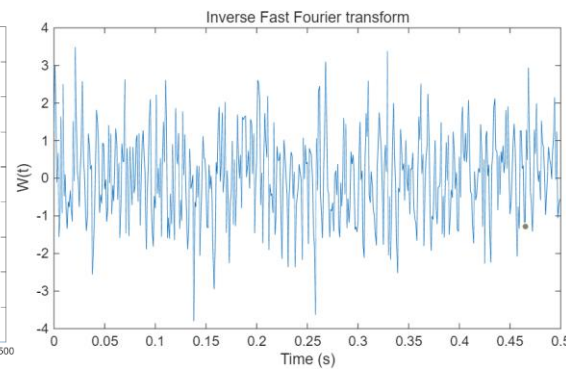


图 7 傅里叶逆变换后重构的信号

实验表明，含噪信号经 FFT 变换后，其频谱中的噪声能量广泛分布在整个频带（图 5）。通过有选择地将大于 200Hz 的高频成分置零（图 6），再进行 iFFT 重构，所得时域信号（图 7）的噪声起伏显著降低，波形更逼近原始信号（图 1）。这表明针对本实验信号的特点，所采用的简单高通滤波能有效抑制高频随机噪声，削弱随机噪声的影响，提升信号质量。

（3）利用含噪信号研究采样频率、采样点数的影响：

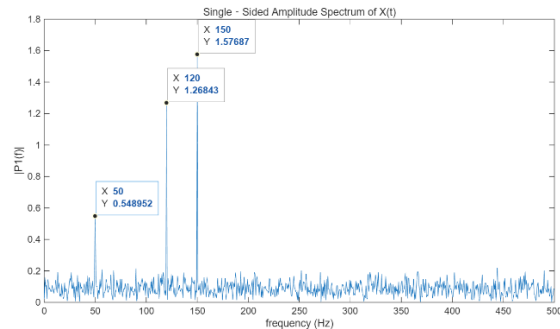


图 8 采样频率 1000Hz，采样点数 1500 的信号频谱

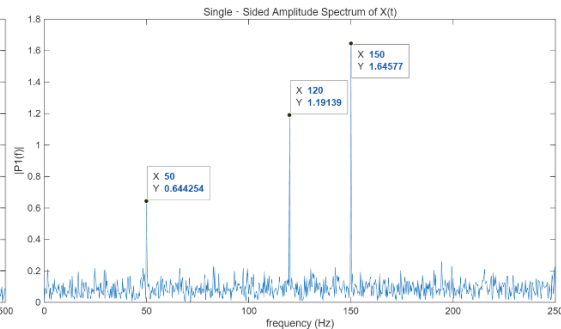


图 9 采样频率 500Hz，采样点数 1500 的信号频谱

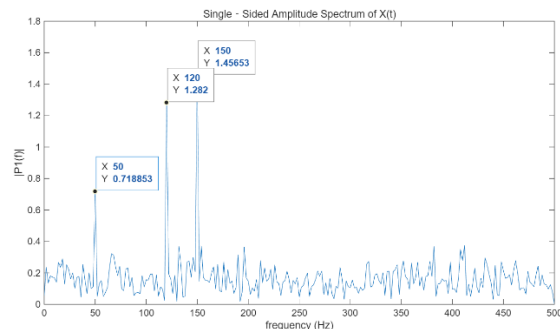


图 10 采样频率 1000Hz，采样点数 500 的信号频谱

采样频率对频谱分析的影响：

保持采样点数 1500 不变，将采样频率从 1000Hz 降低至 500Hz 时：频谱显示范围由 0 - 500Hz 缩减为 0 - 250Hz，符合奈奎斯特采样定理；原有频率分量仍可识别，但幅值有所变化，说明采样频率降低会影响频谱幅值的估计精度。

采样点数对频谱分析的影响

保持采样频率 1000 Hz 不变，将采样点数从 1500 减少至 500 时：频率分辨率显著降低，频谱图明显变稀疏；各频率分量的幅值出现波动，幅值估计准确性下降。

5 实验结论

（1）针对周期信号，FFT 与 iFFT 具有高精度与高效性：通过对无噪声周期信号的 FFT 与 iFFT 处理，重构信号与原始信号高度一致。这表明对于成分明确的周期信号，使用快速傅里叶（逆）变换不仅能获得准确结果，还可大幅降低运算量，提高处理效率。

（2）频域滤波可有效抑制噪声，但存在局限性：对含噪信号进行频域滤波后，信号波形改

善，更接近原始无噪信号。这说明针对噪声主要分布在高频段的情形，简单频域滤波具有实用价值。然而，由于噪声具有随机性与宽频特性，难以完全分离与信号频段重叠的噪声成分，因此滤波后信号仍会残留部分干扰。

(3) 采样参数显著影响频谱分析质量：采样频率决定了频谱的有效范围，频率过低会导致信号高频成分丢失；采样点数直接影响频率分辨率与幅值估计的稳定性，点数不足会降低频谱精细度与可靠性。

6 源代码与分析

(1) 对无噪声信号执行 FFT 和 iFFT:

```
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
S = 0.7 * sin(2 * pi * 50 * t) + 1.2 * sin(2 * pi * 120 * t) + 1.6 * sin(2 * pi * 150 * t);
plot(t(1:500),S(1:500))      % 绘图, 横轴t、纵轴S, 绘制前500个点
xlabel('Time(s)')            % 添加横轴标签
ylabel('Amplitude(V)')      % 添加纵轴标签
title('Time Domain Signal')  % 添加标题
Y = fft(S);                  % 变换后的频谱输出包含同样数量的采样点
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); % 双侧频谱对称的, 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
plot(f,P1)
title('Single-Sided Amplitude Spectrum of S(t)')
xlabel('frequency (Hz)')
ylabel('|P1(f)|')
W = ifft(Y);                 % Inverse Fast Fourier transform
plot(t(1:500),W(1:500))
xlabel('Time (s)')
```

```
ylabel('W(t)')
```

```
title('Inverse Fast Fourier transform')
```

(2) 对含噪信号执行 FFT，滤波后执行 iFFT:

```
Fs = 500; % 采样频率
```

```
T = 1/Fs; % 采样周期
```

```
N = 500; % 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
```

```
S = 0.7 * sin(2 * pi * 50 * t) + 1.2 * sin(2 * pi * 120 * t) + 1.6 * sin(2 * pi * 150 * t);
```

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

```
plot(t(1:500),X(1:500)) % 绘图，横轴t、纵轴S，绘制前500个点
```

```
xlabel('Time(s)') % 添加横轴标签
```

```
ylabel('Amplitude(V)') % 添加纵轴标签
```

```
title('Signal Corrupted with Zero-Mean Random Noise') % 添加标题
```

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

```
%Y(302:751) = 0; % 将正频率部分大于200Hz的部分置零
```

```
%Y(751+301:end) = 0; % 将对称的负频率部分对应的高频成分也置零
```

```
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，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
```

```
plot(f,P1)
```

```
ylim([0 1.8]); % 设置y轴范围从0到1.8V
```

```
title('Single-Sided Amplitude Spectrum of X(t)')
```

```
xlabel('frequency (Hz)')
```

```
ylabel('|P1(f)|')
```

```
W = ifft(Y); % Inverse Fast Fourier transform
```

```
plot(t(1:500),W(1:500))
```

```
xlabel('Time (s)')
```

```
ylabel('W(t)')
```

```
title('Inverse Fast Fourier transform')
```