



北京理工大学
Beijing Institute of Technology

本科生实验报告

课程名称：高级数字信号处理

实验名称：功率谱实验

任课教师：	田黎育			实验教师：	田黎育
实验日期：	2025 年 9 月 29 日			实验地点：	
实验类型：	<input checked="" type="checkbox"/> 原理验证 <input type="checkbox"/> 综合设计 <input type="checkbox"/> 自主创新				
学生姓名：	闫子易	班级：	13212302	学号：	1120230621
学 院：	集成电路与电子学院			专 业：	电子科学与技术
组 号：		同 组 同 学：			
成 绩：					



集成电路与电子学院
SCHOOL OF INTEGRATED CIRCUITS
AND ELECTRONICS

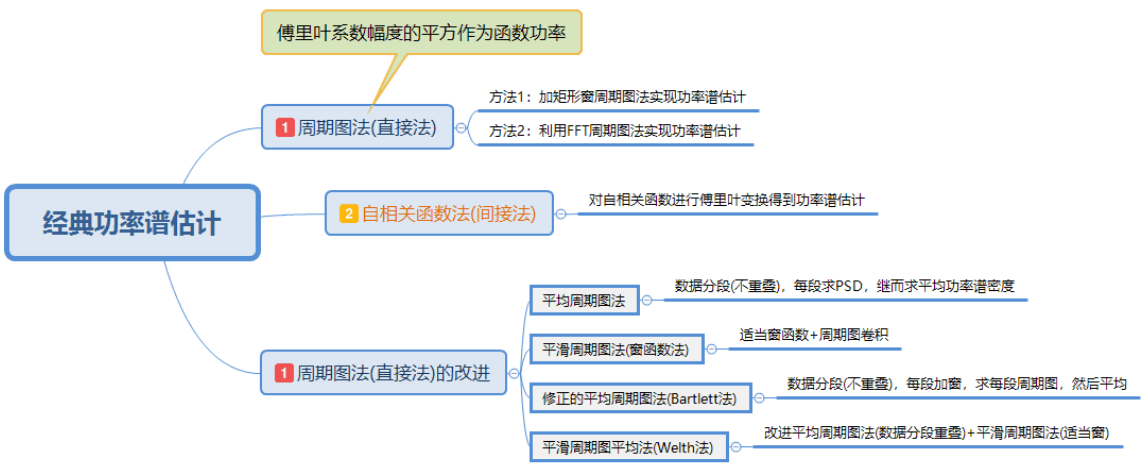
一、实验目的

- ① 理解自相关函数的估计、运算与卷积的联系与区别
- ② 对比两种求功率谱的方法，如运算时间、性能

二、实验原理

经典功率谱估计采用的是传统傅里叶变换分析方法（又称线性谱估计），主要分为自相关法（间接法）和周期图法（直接法）两种。自相关法在 1985 年提出，先估计自相关函数，再计算功率谱。周期图法直接对观测数据进行快速傅里叶变换，得到功率谱。优点是可以直接 FFT 快速计算，所以应用比较广泛。

经典谱估计优点是计算效率高，缺点是频率分辨率低，常用于频率分辨率要求不高的场合。



周期图法（直接法）

周期图法就是把 N 点观测数据看做能量有限的信号，直接对其进行傅里叶变换，然后取其模值的平方，并除以 N ，得到观测数据真实的功率谱的估计。

由于随机过程 $u(n)$ 的 N 个观测值 $u_N(n)$ 是确定信号，对其进行傅里叶变换，得

$$U_N(\omega) = \sum_{n=0}^{N-1} u_N(n) e^{-j\omega n}$$

根据帕塞瓦尔定理，上式的模的平方是确定信号 $u_N(n)$ 的能量谱，对能量谱除以

持续时间 N ，其结果应是 $u_N(n)$ 的功率谱估计，将其作为随机过程 $u(n)$ 的功率谱的估计，表示为

$$\hat{S}_{\text{PER}}(\omega) = \frac{1}{N} |U_N(\omega)|^2$$

该方法称为功率谱估计的周期图法(periodogram)。因为这种功率谱估计方法是直接通过观测数据的傅里叶变换求得的，称之为直接法。

(1) 周期图法是功率谱的有偏估计。产生偏移的原因一方面是局部平均中主瓣的模糊租用，模糊的结果使谱估计的分辨率下降；另一方面是由于旁瓣的泄露。

(2) 频率分辨率低。原因是傅里叶变换域是无限长的，而观测数据是有限长的，相当于将信号在时域添加了矩形窗，在频域真正功率谱卷积一个抽样函数 sinc。

接下来验证一下周期图法不满足无偏性

$$\begin{aligned} E[I_N(w)] &= E \left[\frac{1}{N} |X_N(e^{jw})|^2 \right] \\ &= E \left[\frac{1}{N} X_N(e^{jw}) X_N^*(e^{jw}) \right] \quad \text{写为共轭式方便分析} \\ &= \frac{1}{N} E \left[\sum_{n=0}^{N-1} x(n) e^{-jwn} \sum_{k=0}^{N-1} x(k) e^{jwk} \right] \\ &= \frac{1}{N} E \left[\sum_{n=-\infty}^{\infty} d(n) x(n) e^{-jwn} \sum_{k=-\infty}^{\infty} d(k) x(k) e^{jwk} \right] \quad d(\cdot) \text{ 为一个矩形窗} \\ &= \frac{1}{N} E \left[\sum_n \sum_k d(n) d(k) x(n) x(k) e^{-jw(n-k)} \right] \end{aligned}$$

$$d(n) = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & \text{else} \end{cases}$$

令 $m = n - k$ ，则上式可变为

$$E[I_N(w)] = \sum_m \frac{1}{N} R_{dd}(m) R_{xx}(m) e^{-jwm}$$

我们知道 $x(n)$ 的功率谱是对 $R_{xx}(m)$ 做傅里叶变换

$$P_{xx}(w) = \sum_m R_{xx}(m) e^{-jwm}$$

很显然 $E[I_N(w)]$ 多了一项 $\frac{1}{N} R_{dd}(m)$ ，且这一项不为1，所以

$$\hat{P}_{xx}(w) = E[I_N(w)] \neq P_{xx}(w)$$

实际上这个 $\frac{1}{N}R_{dd}(m)$ 也是有物理意义的，不妨令 $q_N(m) = \frac{1}{N}R_{dd}(m)$ ，则

$$\begin{aligned} q_N(m) &= \frac{1}{N}R_{dd}(m) \\ &= \frac{1}{N}[d(m) * d(-m)] \quad \text{门函数的卷积} \\ &= \begin{cases} 1 - \frac{|m|}{N}, & |m| \leq N-1 \\ 0, & \text{else} \end{cases} \end{aligned}$$

$q_N(m)$ 实际上是一个三角窗，也就是说，用周期图法估计功率谱相当于在时域上给自相关函数加上了一个三角窗后再做傅里叶变换得到估计的功率谱。

自相关函数法（间接法）

一、时域自相关函数 (Autocorrelation Function)

- 数学定义

对于能量信号（总能量有限，比如一个有限的时间序列），自相关函数定义为：

$$R_{xx}(\tau) = \int_{-\infty}^{\infty} x(t) \cdot x^*(t + \tau) dt$$

其中， τ 是时间延迟（Time Lag）， $*$ 表示取复共轭。

对于功率有限的随机过程（总能量无限，但平均功率有限，如噪声），我们无法用单一实现的积分来定义，必须使用统计平均（期望值）。对于一个宽平稳随机过程，其自相关函数定义为：

$$R_{xx}(\tau) = E[\mathbf{x}(t) \cdot \mathbf{x}^*(t + \tau)]$$

宽平稳意味着信号的统计特性（如均值、方差）不随时间平移而变化，因此 $R_{xx}(\tau)$ 只与时间差 τ 有关，与绝对时间 t 无关。

- 自相关函数的重要性质

1. 共轭对称性： $R_{xx}(\tau) = R_{xx}^*(-\tau)$ 。对于实信号，它是偶函数： $R_{xx}(\tau) = R_{xx}(-\tau)$ 。
2. 最大值在原点： $|R_{xx}(\tau)| \leq R_{xx}(0)$ 。在 $\tau = 0$ 时，自相关函数取得最大值。
3. $R_{xx}(0)$ 的物理意义：

- 对于能量信号， $R_{xx}(0) = \int |\mathbf{x}(t)|^2 dt$ ，就是信号的总能量。

- 对于功率信号， $R_{xx}(0) = E[|x(t)|^2]$ ，就是信号的平均功率。
4. 周期性：如果信号是周期性的，其自相关函数也将保持同样的周期性。
5. 衰减性：对于随机信号，通常 $|\tau|$ 越大，相关性越弱，即 $\lim_{|\tau| \rightarrow \infty} R_{xx}(\tau) = \mu_x^2$ (μ_x 是信号的均值)。如果均值为 0，则自相关函数完全衰减到 0。

二、连续信号的维纳辛钦定理

对于宽平稳随机过程，功率谱密度是自相关函数的傅里叶变换+：

$$S_{xx}(f) = \int_{-\infty}^{\infty} R_{xx}(\tau) e^{-j2\pi f\tau} d\tau$$

详细推导

步骤 1：截断信号的功率谱

考虑截断信号 $\mathbf{x}_T(t) = \mathbf{x}(t) \cdot \mathbf{rect}(t/T)$ ，其傅里叶变换为：

$$X_T(f) = \int_{-T/2}^{T/2} x(t) e^{-j2\pi f t} dt$$

截断信号的功率谱估计为：

$$\hat{S}_T(f) = \frac{1}{T} |X_T(f)|^2$$

步骤 2：展开功率谱表达式

$$\hat{S}_T(f) = \frac{1}{T} \int_{-T/2}^{T/2} \int_{-T/2}^{T/2} x(t) x^*(s) e^{-j2\pi f(t-s)} dt ds$$

令 $\tau = t - s$ ，则 $t = s + \tau$ ，雅可比行列式为 1：

$$\begin{aligned} & \cdot \tau \in [-T, T] \\ & \cdot s \in [-T/2, T/2] \\ & \cdot t = s + \tau \in [-T/2, T/2] \Rightarrow s \in [-T/2 - \tau, T/2 - \tau] \end{aligned}$$

所以 s 的实际范围是这两个区间的交集。

求交集

交集区间的左端点： $\max(-T/2, -T/2 - \tau)$

交集区间的右端点： $\min(T/2, T/2 - \tau)$

对于全部 $\tau \in [-T, T]$ ，这个交集区间的长度是 $T - |\tau|$ 。

由于区间长度是 $T - |\tau|$ ，对于全部 $\tau \in [-T, T]$ ，中心点均为 $-\tau/2$ ，所以：

$$s \in \left[-\frac{\tau}{2} - \frac{T - |\tau|}{2}, -\frac{\tau}{2} + \frac{T - |\tau|}{2} \right]$$

简化后：

$$s \in \left[-\frac{T + \tau - |\tau|}{2}, \frac{T - \tau - |\tau|}{2} \right]$$

原始积分为：

令 $s' = s + \frac{\tau}{2}$ ，则 $s = s' - \frac{\tau}{2}$ 。代入 $s = s' - \frac{\tau}{2}$ ：

- 积分限变为: $s' \in \left[-\frac{T - |\tau|}{2}, \frac{T - |\tau|}{2} \right]$

被积函数: $x(s' + \frac{\tau}{2})x^*(s' - \frac{\tau}{2})$

此时积分形式变为

$$\hat{S}_T(f) = \frac{1}{T} \int_{-T}^T \left[\int_{-\frac{T - |\tau|}{2}}^{\frac{T - |\tau|}{2}} x(s' + \frac{\tau}{2})x^*(s' - \frac{\tau}{2}) ds' \right] e^{-j2\pi f\tau} d\tau$$

步骤 3：取数学期望和极限

$$\mathbb{E}[\hat{S}_T(f)] = \int_{-T}^T \left(\frac{1}{T} \int_{-\frac{T - |\tau|}{2}}^{\frac{T - |\tau|}{2}} \mathbb{E}[x(s' + \frac{\tau}{2})x^*(s' - \frac{\tau}{2})] ds' \right) e^{-j2\pi f\tau} d\tau$$

由于宽平稳性， $\mathbb{E}[x(s' + \frac{\tau}{2})x^*(s' - \frac{\tau}{2})] = R_{xx}(\tau)$ 与 s' 无关。因此积分简化为：

$$\mathbb{E}[\hat{S}_T(f)] = \int_{-T}^T \left(1 - \frac{|\tau|}{T} \right) R_{xx}(\tau) e^{-j2\pi f\tau} d\tau$$

取极限 $T \rightarrow \infty$

$$S_{xx}(f) = \lim_{T \rightarrow \infty} \mathbb{E}[\hat{S}_T(f)] = \int_{-\infty}^{\infty} R_{xx}(\tau) e^{-j2\pi f\tau} d\tau$$

证毕。

逆关系

由傅里叶逆变换可得：

$$R_{xx}(\tau) = \int_{-\infty}^{\infty} S_{xx}(f) e^{j2\pi f\tau} df$$

特别地，当 $\tau = 0$ 时：

$$R_{xx}(0) = \mathbb{E}[|x(t)|^2] = \int_{-\infty}^{\infty} S_{xx}(f) df$$

这表示总功率等于功率谱密度在全频域的积分。

三、有限长度离散信号的维纳辛钦定理

信号定义

- 有限长信号: $x[n], n = 0, 1, \dots, N-1$
- 采样间隔: $\Delta t = 1/f_s$
- 信号时长: $T = N\Delta t$

自相关函数（有偏估计）

$$R_{xx}[m] = \frac{1}{N} \sum_{n=\max(0, -m)}^{\min(N-1, N-1-m)} x[n]x^*[n+m]$$

对于 $|m| \geq N, R_{xx}[m] = 0$

功率谱密度（周期图法）（对 PSD 进行频域积分而非求和可以得到信号功率）

$$PSD[k] = \frac{1}{N \cdot f_s} \left| \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn} \right|^2 = \frac{1}{N \cdot f_s} |X[k]|^2$$

维纳辛钦定理指出：

功率谱密度（PSD）是自相关函数的傅里叶变换。

对于长度为 N 的离散信号 $\mathbf{x}[n]$ ，定理表示为：

$$PSD[k] = DFT(R_{xx})$$

$$\frac{1}{N \cdot f_s} \left| \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn} \right|^2 = \sum_{m=-(N-1)}^{N-1} R_{xx}[m] e^{-j\frac{2\pi}{N}km}$$

自相关与卷积

自相关的直观含义就是：把一个信号平移一段距离，跟原来有多相似。

于是就有了自相关的定义：

$$R(\tau) = \int_{-\infty}^{\infty} x(t)x(t-\tau)dt$$

它代表了“移、乘、积”这三步操作。

如果只谈自相关，其实到此就可以结束了。

卷积跟自相关的定义很相似，包含了“卷、移、乘、积”四步操作：

$$(x * y)(\tau) = \int_{-\infty}^{\infty} x(t)y(\tau - t)dt$$

左边有时也写作 $x(t) * y(t)$ ，表示这个函数是由 $x(t)$ 和 $y(t)$ 卷积而得的，但它的自变量是 τ 。

我们发现卷积比自相关多了一步“卷”的操作，也就是自相关时要对信号进行时间反转（翻转）再卷积，为了去掉这个多余的操作，我们先把原信号自己卷一下，就可以抵消掉卷积中的“卷”操作了。这就是自相关与卷积的关系：

$$R(\tau) = x(t) * x(-t)$$

对于确定性信号，无统计平均含义，其**卷积和自相关**存在如下关系：

1. 连续信号

$$R_f(\tau) = f(\tau) * f(-\tau)$$

2. 离散序列

$$R_h(m) = h(m) * h(-m)$$

现证明如下：

1. 连续信号

• 卷积

$$\begin{aligned} f(t) * f(t) &= \int_{-\infty}^{\infty} f(\tau)f(t - \tau)d\tau \\ f(t) * f(-t) &= \int_{-\infty}^{\infty} f(\tau)f(-(t - \tau))d\tau \\ &= \int_{-\infty}^{\infty} f(\tau)f(\tau - t)d\tau \\ &= \int_{-\infty}^{\infty} f(\tau - t)f(\tau)d\tau \end{aligned}$$

$$f(\tau) * f(-\tau) = \int_{-\infty}^{\infty} f(t - \tau)f(t)dt$$

• 自相关

$$R_f(\tau) = \int_{-\infty}^{\infty} f(t)f(t - \tau)dt$$

由此可得, $R_f(\tau) = f(\tau) * f(-\tau)$

2. 离散序列

• 卷积

$$h(n) * h(n) = \sum_{m=-\infty}^{\infty} h(m)h(n-m)$$

$$h(n) * h(-n) = \sum_{m=-\infty}^{\infty} h(-m)h(n-m)$$

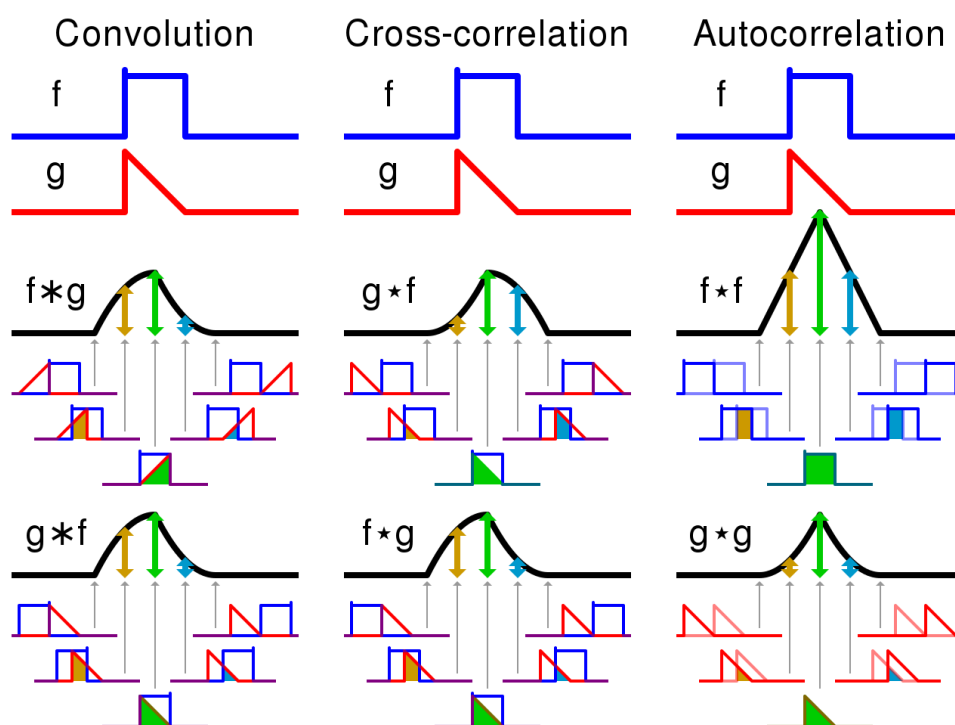
$$= \sum_{m=-\infty}^{\infty} h(m)h(n+m)$$

$$h(m) * h(-m) = \sum_{n=-\infty}^{\infty} h(n)h(n+m)$$

自相关

$$R_h(m) = \sum_{n=-\infty}^{\infty} h(n)h(n+m)$$

由此可得, $R_h(m) = h(m) * h(-m)$



那么自相关与卷积有什么区别呢？

自相关函数与“信号自己和自己卷积”的关键区别在于：**自相关需要对信号进行时间反转**，而直接卷积没有反转。这一差别导致自相关函数总是偶函数，与功率谱密度直接相关，而自卷积则更多体现信号形状的“扩展”。

$$(x * y)[n] = \sum_{k=-\infty}^{+\infty} x[k] y[n-k]$$

$$R_{xx}[m] = \sum_{n=-\infty}^{+\infty} x[n] x[n+m]$$

注意，这里自相关没有时间反转。

特性	自相关 $R_{xx}(\tau)$	自己与自己卷积 $(x * x)(t)$
是否翻转	有： $x(-t)$	无
对称性	总是偶函数： $R_{xx}(\tau) = R_{xx}(-\tau)$	不一定对称
物理意义	衡量信号在不同延迟下的相似性	信号能量在时间上的“扩展”
频域意义	傅里叶变换 = 功率谱密度 (Wiener - Khinchin 定理)	傅里叶变换 = 频谱的平方 $(X(f))^2$ ，包含相位信息
典型结果	峰值在 $\tau = 0$ ，值等于信号能量	峰值位置与信号形状相关，不一定在 $t = 0$

此外，还有四种常见的**改进周期图法**。周期图法相当于一个矩形窗对时域信号的加窗，然后分辨率大约于数据长度 N 成反比。

平均周期图法

对一个随机变量观测时，得到 L 组独立数据，每组数据长为 M ，对每一组求 PSD，之后将 L 个均值加起来求平均。这样得到的均值，其方差是原来的 $1/L$ 。

平均周期图法仍然是有偏估计，偏移和每组数据长度 M 有关。由于每段 FFT 的长度变为 M ，分辨率更低 (F_s/M)，因此，平均周期图法以分辨率的降低换取了估计方差的减少。

窗函数法

窗函数法是将长度为 N 的观测数据乘以同一长度的数据窗 w ，数据加窗后，谱估计值的数学期望等于真实谱值与窗谱函数的平方卷积，因而是有偏估计。

质量分析：

选择低旁瓣的数据窗可使得杂散响应减少，但旁瓣的降低必然使得主瓣加宽，从而降低分辨率。

数据加窗后，谱估计值的方差 \geq 谱估计值的平方，且不随数据长度增大而减少。数据加窗可以降低谱估计值的旁瓣，但是降低了谱估计的分辨率。

修正的周期图平均法

又叫 Bartlett 法，首先把长度为 N 的数据分成 L 段，每段数据长为 M ，则 $N=LM$ 。然后把窗函数 w 加到每段数据上，求出每段的周期图，之后对每段周期图进行评价。

质量分析：

Bartlett 法在计算周期图前，先对各数据段加窗，是平均周期图法的估计方差减少，但是分辨率降低。

加权交叠平均法

又称 Welch 法。对 Bartlett 法的改进。首先，分段时相邻两段可以重叠，其次，窗函数使用汉宁窗或汉明窗，通过改进，达到进一步减小方差的目的。

三、运行代码及实验内容

% 自写周期图与自相关法功率谱估计（含峰值标注优化）

```
clear; clc; close all;
```

```
f1 = 300; f2 = 310; fs = 1000; % 采样率
```

```
dt = 1/fs;
```

```
Td = 0.3; % 信号时间长度
```

```
t = 0:dt:Td;
```

```

x = cos(2*pi*t*f1) + cos(2*pi*t*f2) + randn(size(t));
nfft = 2^11;
fprintf('f1 = %d, f2 = %d, fs = %d \n', f1,f2,fs );
fprintf('Td = %f, nfft = %d \n', Td, nfft );

% ----- 周期图功率谱估计（计时） -----
tic;
[Pwd1, ff1] = myPeriodogram(x, nfft, fs);
time_periodogram = toc;
fprintf('Periodogram computation time: %.6f seconds\n', time_periodogram);

% ----- 自相关功率谱估计（计时） -----
tic;
[Pwd3, ff3] = myAutocorrPSD(x, nfft, fs);
time_autocorr_fft = toc;
fprintf('Autocorrelation+FFT computation time: %.6f seconds\n',
time_autocorr_fft);

% ----- 绘图 -----
fUB = fs/2; fLB = 0;
upperB = -5; lowerB = -50;

figure;

%% ----- (1) 周期图功率谱 -----
subplot(2,1,1);
plot(ff1, 10*log10(Pwd1), 'b'); hold on;
title('周期图功率谱估计（自写函数）');
xlabel('频率 (Hz)');
ylabel('功率谱密度 (dB/Hz)');
xlim([fLB fUB]); ylim([lowerB upperB]); grid on;

% 寻找最大两个峰值
[pks1, locs1] = findpeaks(10*log10(Pwd1), ff1, 'SortStr', 'descend');
num_peaks = min(2, length(pks1));
pks1 = pks1(1:num_peaks);
locs1 = locs1(1:num_peaks);

% 定义颜色（第一个红，第二个绿）
colors = {'r','g'};

% 标注峰值

```

```

for i = 1:num_peaks
    plot(locs1(i), pks1(i), 'o', 'MarkerFaceColor', colors{i}, ...
        'MarkerEdgeColor', colors{i}, 'MarkerSize', 4);

    if num_peaks > 1 && abs(locs1(i) - mean(locs1)) < 10
        text(locs1(i)+5, pks1(i)+(-1)^(i)*3, ...
            sprintf('%.1f Hz, %.1f dB', locs1(i), pks1(i)), ...
            'Color', colors{i}, 'FontSize', 9, 'HorizontalAlignment', 'center');
    else
        text(locs1(i)+5, pks1(i), ...
            sprintf('%.1f Hz, %.1f dB', locs1(i), pks1(i)), ...
            'Color', colors{i}, 'FontSize', 9, 'HorizontalAlignment', 'center');
    end
end
end

```

%% ----- (2) 自相关功率谱 -----

```

subplot(2,1,2);
plot(ff3, 10*log10(Pwd3), 'b'); hold on;
title('自相关功率谱估计 (自写函数)');
xlabel('频率 (Hz)');
ylabel('功率谱密度 (dB/Hz)');
xlim([fLB fUB]); ylim([lowerB upperB]); grid on;

```

% 寻找最大两个峰值

```

[pks3, locs3] = findpeaks(10*log10(Pwd3), ff3, 'SortStr', 'descend');
num_peaks = min(2, length(pks3));
pks3 = pks3(1:num_peaks);
locs3 = locs3(1:num_peaks);

```

% 定义颜色 (第一个红, 第二个绿)

```

colors = {'r', 'g'};

```

% 标注峰值

```

for i = 1:num_peaks
    plot(locs3(i), pks3(i), 'o', 'MarkerFaceColor', colors{i}, ...
        'MarkerEdgeColor', colors{i}, 'MarkerSize', 4);

    if num_peaks > 1 && abs(locs3(i) - mean(locs3)) < 10
        text(locs3(i)+5, pks3(i)+(-1)^(i)*3, ...
            sprintf('%.1f Hz, %.1f dB', locs3(i), pks3(i)), ...
            'Color', colors{i}, 'FontSize', 9, 'HorizontalAlignment', 'center');
    end
end

```

```

else
    text(locs3(i)+5, pks3(i), ...
        sprintf('%0.1f Hz, %0.1f dB', locs3(i), pks3(i)), ...
        'Color', colors{i}, 'FontSize',9,'HorizontalAlignment','center');
end
end
end

```

```

%% ===== 子函数 =====

```

```

% ----- 周期图法 -----

```

```

function [Pxx, f] = myPeriodogram(x, nfft, fs)
    N = length(x);
    X = fft(x, nfft);           % FFT
    Pxx = (1/(N*fs)) * abs(X).^2; % 功率谱密度
    Pxx = Pxx(1:nfft/2+1);      % 单边谱
    Pxx(2:end-1) = 2*Pxx(2:end-1); % 单边能量补偿
    f = (0:nfft/2) * fs/nfft;   % 频率轴
end

```

```

% ----- 自相关法 -----

```

```

function [Pxx, f] = myAutocorrPSD(x, nfft, fs)
    N = length(x);
    % 手动计算有偏自相关
    R = zeros(1, 2*N-1);
    for k = -(N-1):(N-1)
        idx = k + N; % shift index
        if k >= 0
            R(idx) = sum(x(1:N-k) .* x(1+k:N)) / N;
        else
            R(idx) = sum(x(1-k:N) .* x(1:N+k)) / N;
        end
    end
    end
    % FFT 得到功率谱
    S = fft(R, nfft);
    Pxx = abs(S)/fs;
    Pxx = Pxx(1:nfft/2+1);      % 单边谱
    Pxx(2:end-1) = 2*Pxx(2:end-1); % 单边能量补偿
    f = (0:nfft/2) * fs/nfft;   % 频率轴
end

```

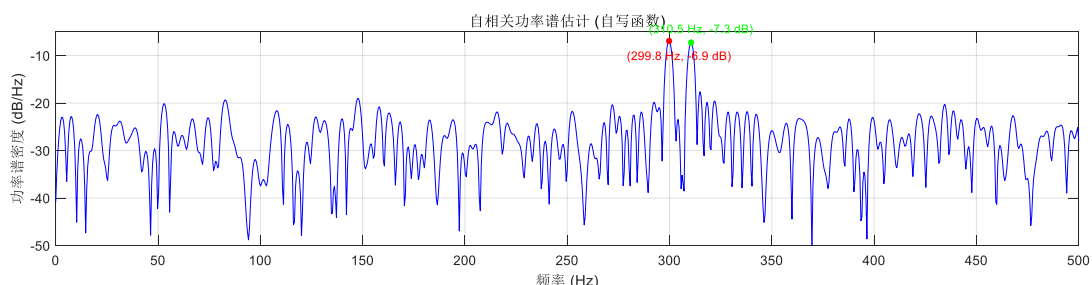
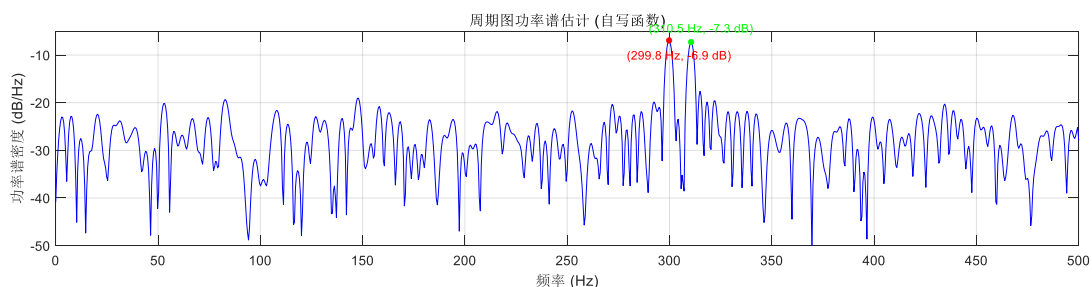
首先，由于要对比两种求功率谱方法的运算时间和性能，所以我在代码中加入了tic tok，来得到两种运算的所需时间。

```
tic;
[Pwd1, ff1] = myPeriodogram(x, nfft, fs);
time_periodogram = toc;
```

同时，我们最开始设置的信号是两个信号叠加，一个 300Hz，一个 310Hz，当然这两个频率可以自己改变，那么在功率谱上就应该明显地显示出有两个突出的谱线（或者说峰），实际运行结果也确实如此。所以我把图中最高的两个极大点用一红一绿两个点标注了出来，同时在旁边附上了它们的坐标，这样可以一目了然地看见初始信号的两个频率是多少，也方便验证后续的实验。

```
% 标注峰值
for i = 1:num_peaks
    plot(locs1(i), pks1(i), 'o', 'MarkerFaceColor', colors{i}, ...
        'MarkerEdgeColor', colors{i}, 'MarkerSize', 4);

    if num_peaks > 1 && abs(locs1(i) - mean(locs1)) < 10
        text(locs1(i)+5, pks1(i)+(-1)^i*3, ...
            sprintf('%.1f Hz, %.1f dB', locs1(i), pks1(i)), ...
            'Color', colors{i}, 'FontSize',9,'HorizontalAlignment','center');
    else
        text(locs1(i)+5, pks1(i), ...
            sprintf('%.1f Hz, %.1f dB', locs1(i), pks1(i)), ...
            'Color', colors{i}, 'FontSize',9,'HorizontalAlignment','center');
    end
end
```



但在测周期图法和自相关法的运算时间时，我发现一个问题。理论上基于自相关的功率谱（先算自相关再做 FFT）涉及 $O(N^2)$ 的直接自相关或 $O(N\log N)$ 的用 FFT 加速的实现，周期图（直接对信号做 FFT）一般是 $O(N\log N)$ 。但实际命令行里输出的时候，很多时候是自相关法的运算时间小于周期图法。

Periodogram computation time: 0.025519 seconds

Autocorrelation+FFT computation time: 0.013021 seconds

Periodogram computation time: 0.004148 seconds

Autocorrelation+FFT computation time: 0.003240 seconds

Periodogram computation time: 0.000857 seconds

Autocorrelation+FFT computation time: 0.000300 seconds

Periodogram computation time: 0.001201 seconds

Autocorrelation+FFT computation time: 0.000458 seconds

可以看到，我运行了几次程序，自相关法的运算时间都小于周期图法，这显然有悖于理论，于是我问了 AI 什么原因。

内置函数的不同实现路径 MATLAB 的 `xcorr`/`xcorrbiased`、`periodogram` 等函数内部实现不同。许多自相关实现会检测到可以用 FFT 去计算自相关（利用 Wiener-Khinchin 定理），因此实际复杂度接近 FFT 的 $O(N\log N)$ 。如果 `xcorr` 已被高度优化并且调用了 FFTW、多线程或矢量化实现，时间会很短。

FFT 长度和零填充 (nfft) 差异 你对自相关结果做 FFT 时，通常是对 $2N-1$ 长度的自相关做 nfft；而 `periodogram` 直接对信号做 nfft。不同的 nfft（尤其是是否为 2^k ）会显著影响 FFT 的速度。更短或更“友好”的 nfft 会更快。

通俗地来理解，就是 Matlab 自己检测到我的自相关方法有更简便的方法来算，

就自动给我优化了算法，导致两个方法的时间复杂度一样。

所以为了能看到真实的自相关法和周期图法的运算性能的比较，我自己手动写了两个方法的子函数，不调用 Matlab 自带的高级函数，把每一步都拆成 for 循环来实现。

```
%% ===== 子函数 =====

% ----- 周期图法 -----
function [Pxx, f] = myPeriodogram(x, nfft, fs)
    N = length(x);
    X = fft(x, nfft);           % FFT
    Pxx = (1/(N*fs)) * abs(X).^2; % 功率谱密度
    Pxx = Pxx(1:nfft/2+1);      % 单边谱
    Pxx(2:end-1) = 2*Pxx(2:end-1); % 单边能量补偿
    f = (0:nfft/2) * fs/nfft;   % 频率轴
end

% ----- 自相关法 -----
function [Pxx, f] = myAutocorrPSD(x, nfft, fs)
    N = length(x);
    % 手动计算有偏自相关
    R = zeros(1, 2*N-1);
    for k = -(N-1):(N-1)
        idx = k + N; % shift index
        if k >= 0
            R(idx) = sum(x(1:N-k) .* x(1+k:N)) / N;
        else
            R(idx) = sum(x(1-k:N) .* x(1:N+k)) / N;
        end
    end
    % FFT 得到功率谱
    S = fft(R, nfft);
    Pxx = abs(S)/fs;
    Pxx = Pxx(1:nfft/2+1);      % 单边谱
    Pxx(2:end-1) = 2*Pxx(2:end-1); % 单边能量补偿
    f = (0:nfft/2) * fs/nfft;   % 频率轴
end
```

几次运行后命令行结果如下：

$f1 = 300, f2 = 310, fs = 1000$

$Td = 0.300000, nfft = 2048$

Periodogram computation time: 0.000513 seconds

Autocorrelation+FFT computation time: 0.004881 seconds

$f1 = 300, f2 = 310, fs = 1000$

$Td = 0.300000, nfft = 2048$

Periodogram computation time: 0.000480 seconds

Autocorrelation+FFT computation time: 0.001347 seconds

$f1 = 300, f2 = 310, fs = 1000$

$Td = 0.300000, nfft = 2048$

Periodogram computation time: 0.000427 seconds

Autocorrelation+FFT computation time: 0.000794 seconds

$f1 = 300, f2 = 310, fs = 1000$

$Td = 0.300000, nfft = 2048$

Periodogram computation time: 0.000152 seconds

Autocorrelation+FFT computation time: 0.000894 seconds

可以看到，在自己写两个方法的子函数后，运行时间上自相关法明显大于周期图法，与理论一致。周期图法在运算时间上通常更快，因为它只需要一次快速傅里叶变换和平方操作；自相关函数法则需要先计算自相关序列，再进行傅里叶变换，整体计算量更大，因此在性能上往往比周期图法慢。

从计算复杂度的角度来看，周期图法的核心就是对长度为 NN 的信号做一次 FFT，复杂度大约是 $O(N\log N)$ ，再加上取模平方的操作，几乎可以忽略不计。这意味着当数据量很大时，周期图法依然保持较高的效率，尤其是在现代硬件上，FFT 已经高度优化，因此它在性能上几乎是最优的直接估计方法。

相比之下，自相关函数法需要先估计自相关函数。如果采用最直接的定义式去计算自相关，复杂度是 $O(N^2)$ ，这在大样本情况下会非常耗时。虽然可以通过 FFT 来加速自相关的计算（利用卷积定理，将自相关转化为频域乘积再逆变换），这样复杂度也能降到 $O(N\log N)$ ，但这实际上相当于多做了一次 FFT 和一次 IFFT，再加上后续的傅里叶变换步骤，整体运算量比周期图法要多一倍甚至更多。

此外，内存占用上也有差异。周期图法只需要存储原始信号和其频谱，而自相关函数法需要额外存储自相关序列，长度可能接近 $2N-1$ ，这在大规模数据下会增加内存压力。再加上自相关函数法通常还要对自相关序列加窗或截断，这些操作虽然在理论上改善了估计的统计特性，但在性能上进一步增加了开销。因此，从工程实现的角度看，周期图法在性能和速度上更具优势，而自相关函数法则更偏向于在需要平滑谱估计时使用，属于用计算时间换取估计质量的策略。

随着 $nfft$ 的增加，观察两个方法的运算时间。

$f1 = 300, f2 = 310, fs = 1000$

$Td = 0.300000, nfft = 2048$

Periodogram computation time: 0.000331 seconds

Autocorrelation+FFT computation time: 0.001378 seconds

$f1 = 300, f2 = 310, fs = 1000$

$Td = 0.300000, nfft = 8192$

Periodogram computation time: 0.001435 seconds

Autocorrelation+FFT computation time: 0.002948 seconds

$f1 = 300, f2 = 310, fs = 1000$

$Td = 0.300000, nfft = 32768$

Periodogram computation time: 0.001361 seconds

Autocorrelation+FFT computation time: 0.004778 seconds

$f1 = 300, f2 = 310, fs = 1000$

$Td = 0.300000, nfft = 131072$

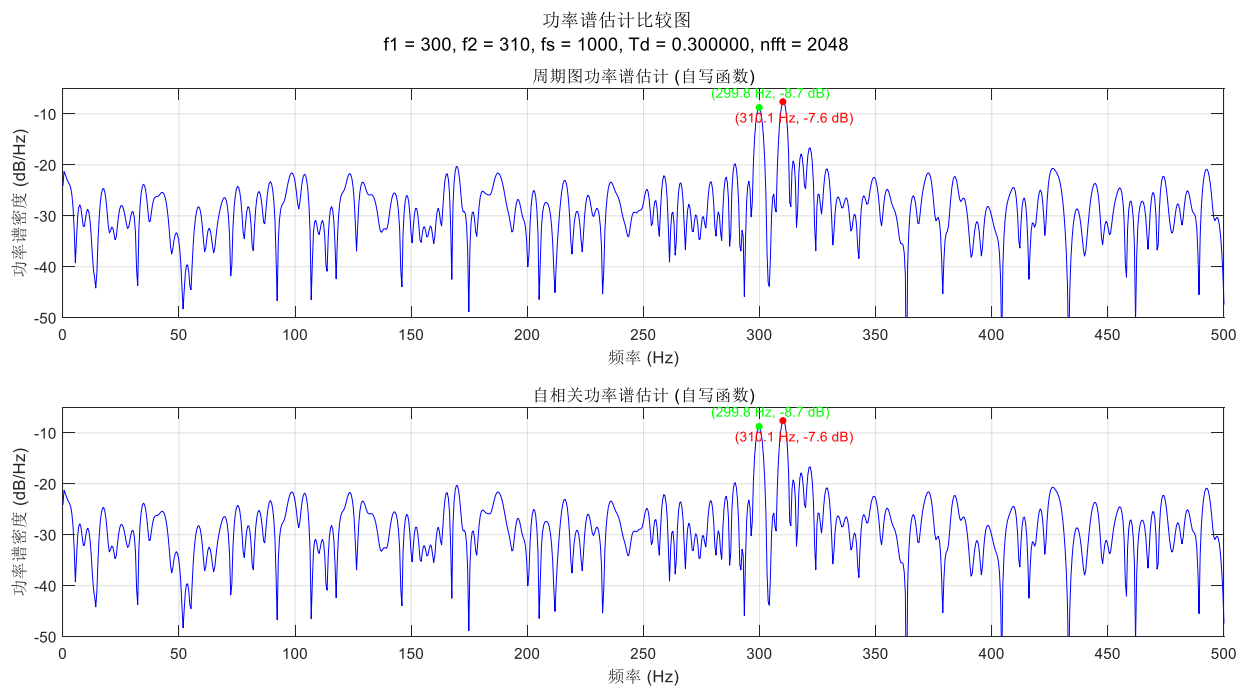
Periodogram computation time: 0.002672 seconds

Autocorrelation+FFT computation time: 0.004200 seconds

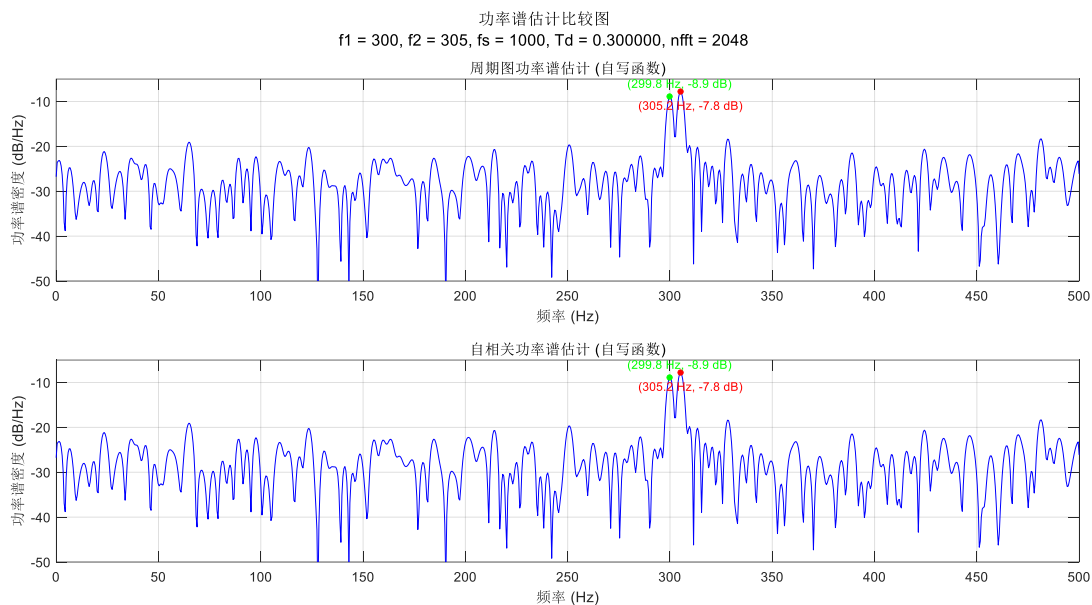
可以看到，随着 $nfft$ 的增加，周期图法和自相关法的运行时间整体上呈上升趋势，且自相关法始终比周期图法所需的运算时间要长，消耗大。

观察功率谱对两个频率分量的分辨

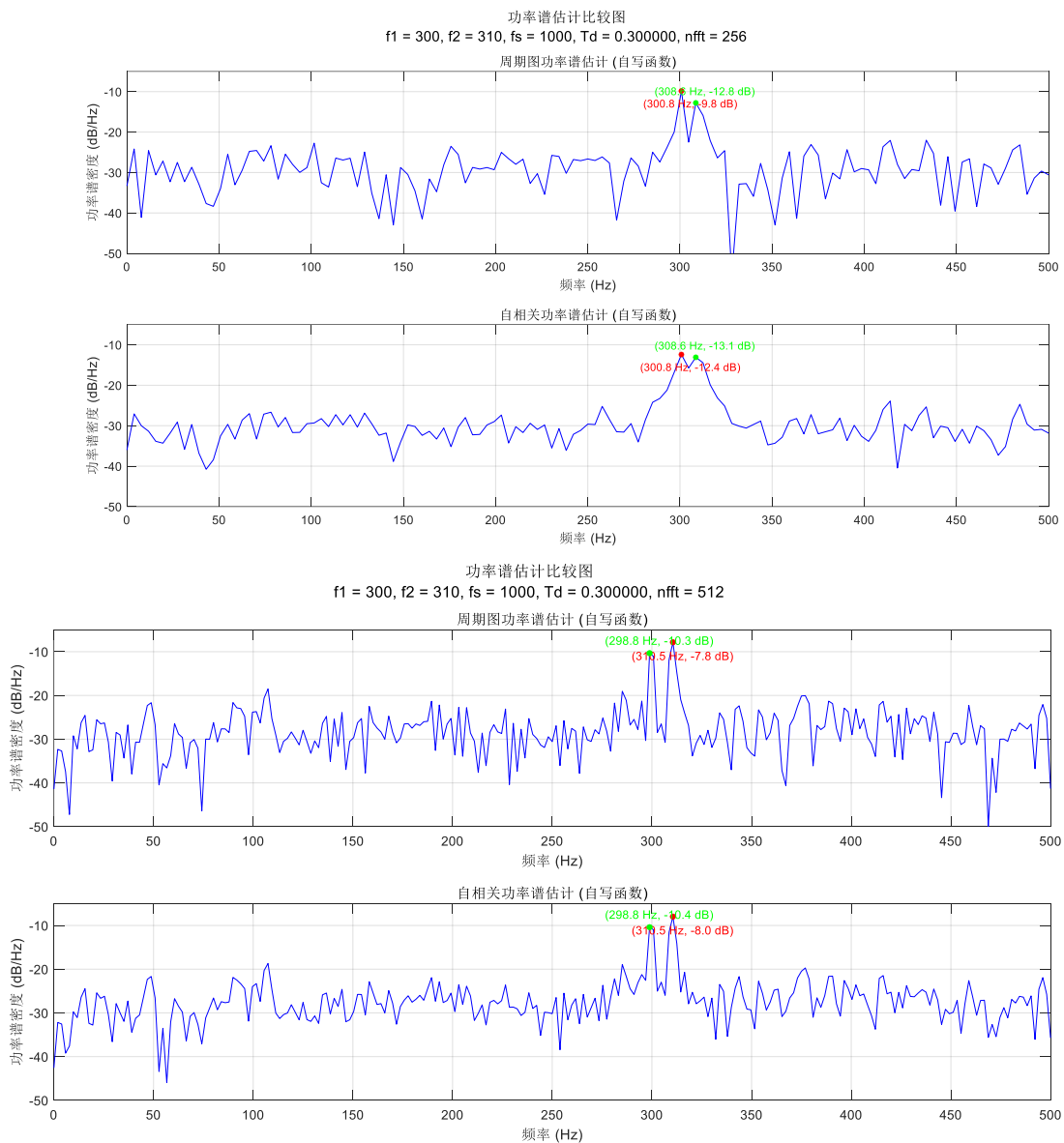
固定 $fs=1000, Td=0.300000, nfft=2048$ ，当 $f1=300, f2=310$ 时，可以在功率谱图中清晰地看见两个突起的峰，且峰值的坐标显示，两个峰值对应的横坐标几乎就是 $f1$ 和 $f2$ 。

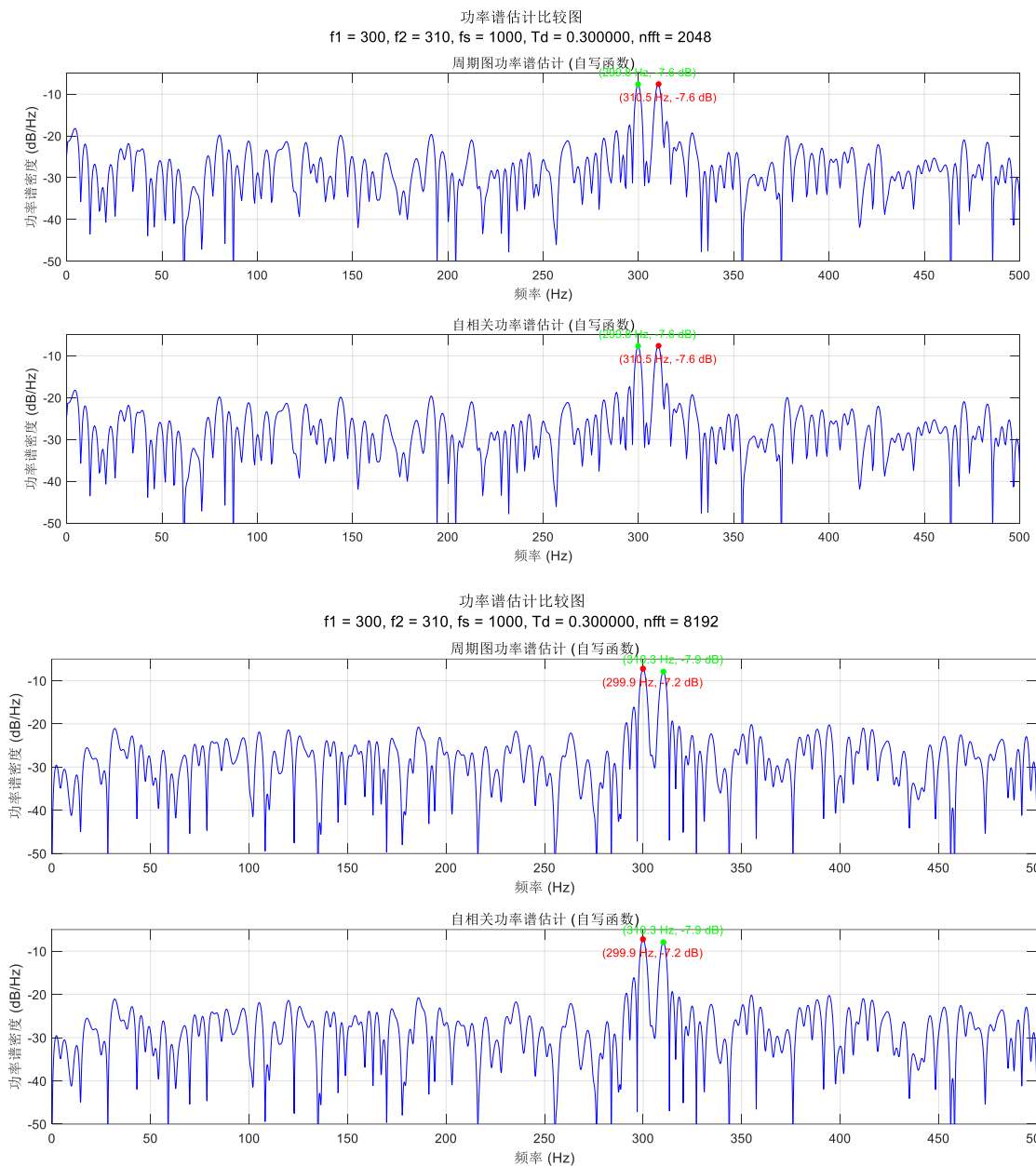


同样的，我们把 $f1$ 和 $f2$ 改为 300Hz 和 305Hz，也能轻松地分辨出来。

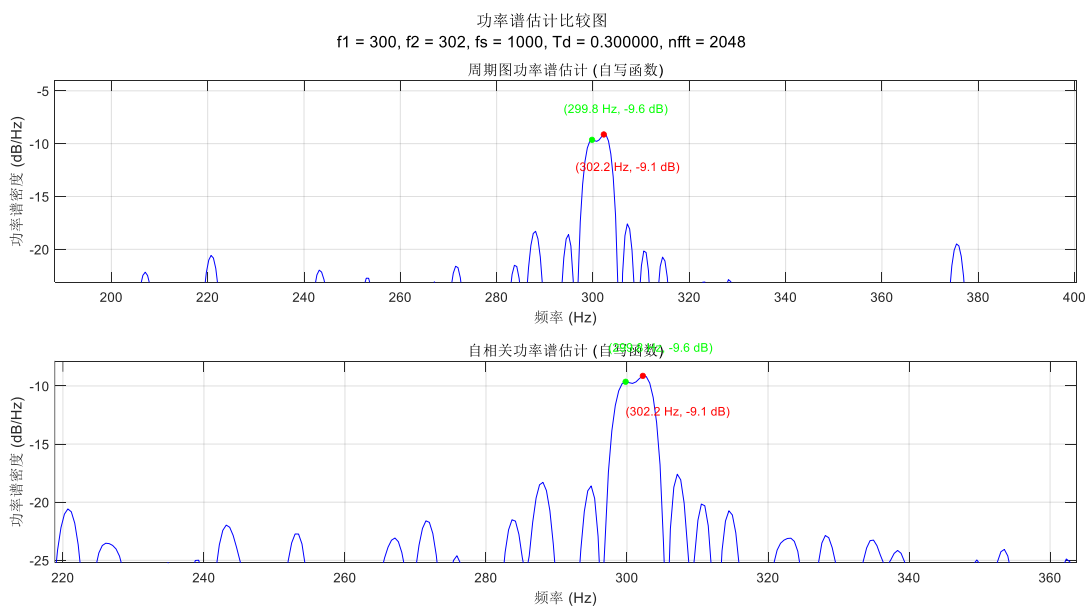


我们再改变 `nfft` 的值，将其从小到大变化。



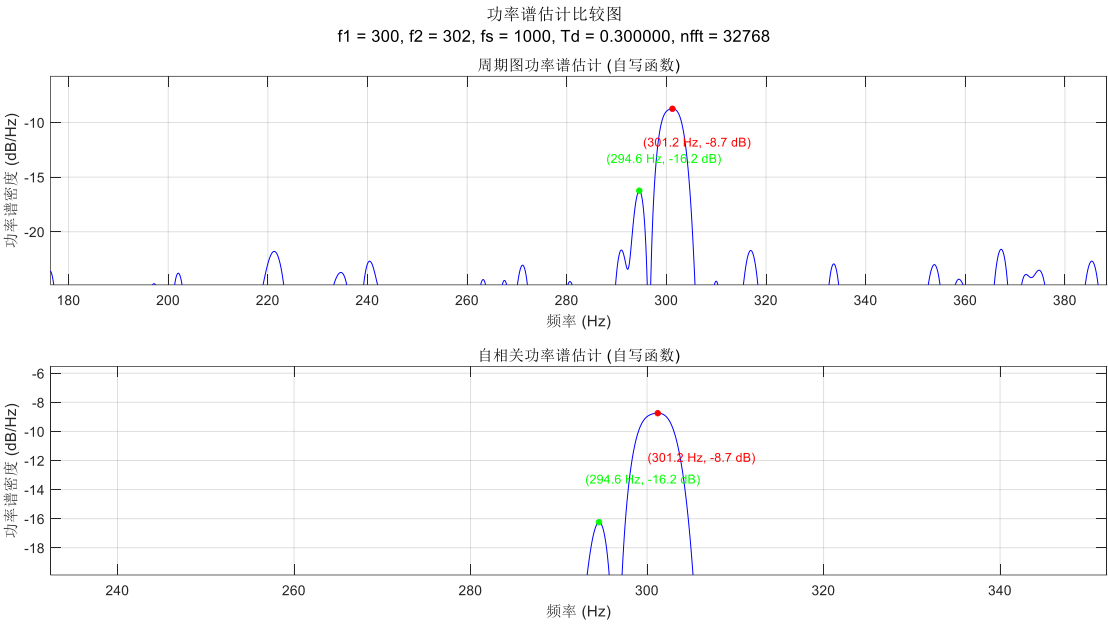


以上四幅图时 $nfft$ 从 2^8 到 2^{13} , 每次乘 4, 可以看到, 谱线越来越密集和清晰, 最高的两个峰值对应的横坐标也越来越接近 300Hz 和 310Hz, 即使在 $nfft$ 很小的时候, 也能明显辨认出两个峰值。

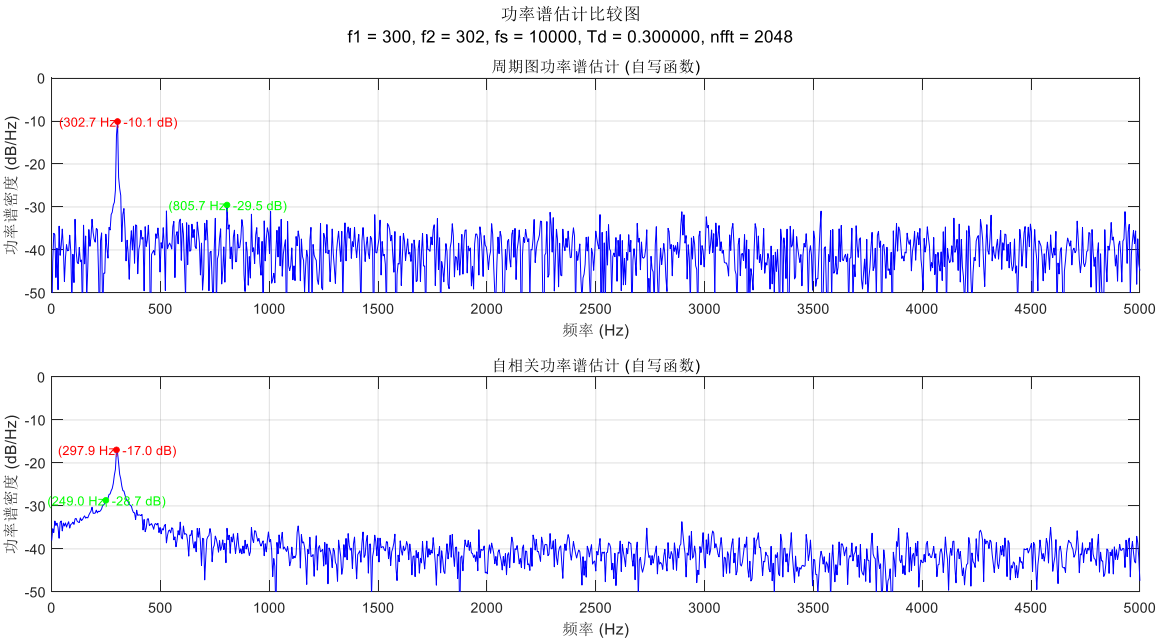


但当我们把 f_1 和 f_2 改为 300Hz 和 302Hz 时，会发现一个现象，两个峰值重叠在一起了，谱线中不能突显原信号的两个频率的合成。

即便增加 $nfft$ 的值，两个峰也还是“融合”在一起分不开。下图中时 $nfft$ 为 2^{15} 时的谱线，可以看到两个最大值点对应的横坐标为 294Hz 和 301.2Hz，并不是原信号的 300Hz 和 302Hz，也就是分辨失败了



同样，增大 fs 也没有用。



这是为什么呢？我们知道，`nfft` 其实做的就是“补零”的操作。FFT 点数越多，频率轴划分越细，曲线看起来更平滑。观测信号长度 $T_d=0.3s$ ，自然频率分辨率 $=1/T_d=3.33Hz$ ，补零到 2048 点后，虽然本质分辨率没变，但插值后能更精确地看到峰值落在哪个频率点，但补零不能突破 3.33Hz 这个极限。

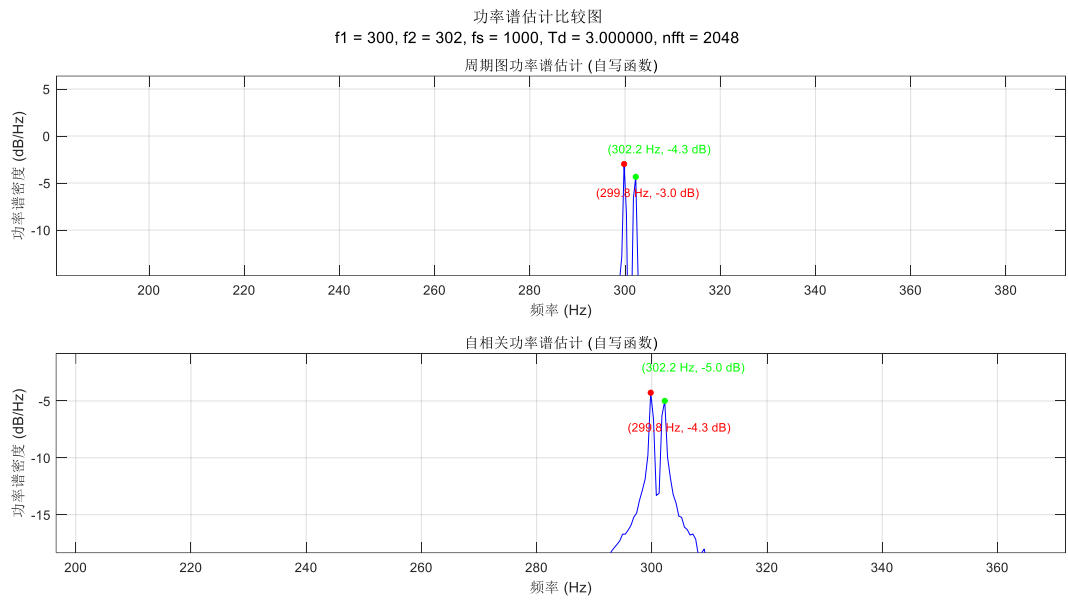
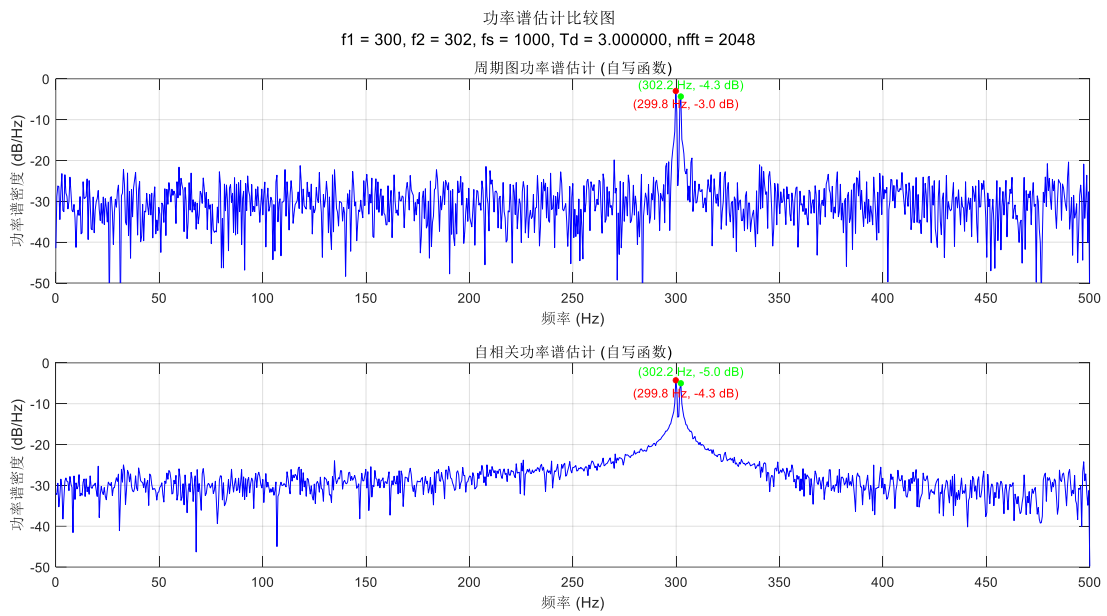
补零在周期图法和自相关函数法中主要作用是提高频率采样的密度，从而让功率谱曲线看起来更平滑、更细致，但它并不会真正提升频率分辨率。补零不会降低谱估计的方差，抖动依然存在。

在周期图法中，功率谱估计是通过对有限长信号做离散傅里叶变换再取模平方得到的。由于信号长度有限，频率采样点数也有限，得到的谱往往比较粗糙。如果在时域信号后面补零，相当于延长了序列长度，使得 FFT 的点数增加，频域采样间隔变小，于是功率谱曲线在频率轴上看起来更连续、更平滑。这种效果常常被误解为分辨率提高，但实际上主瓣宽度并没有缩小，真正的频率分辨率仍然由原始数据长度决定。

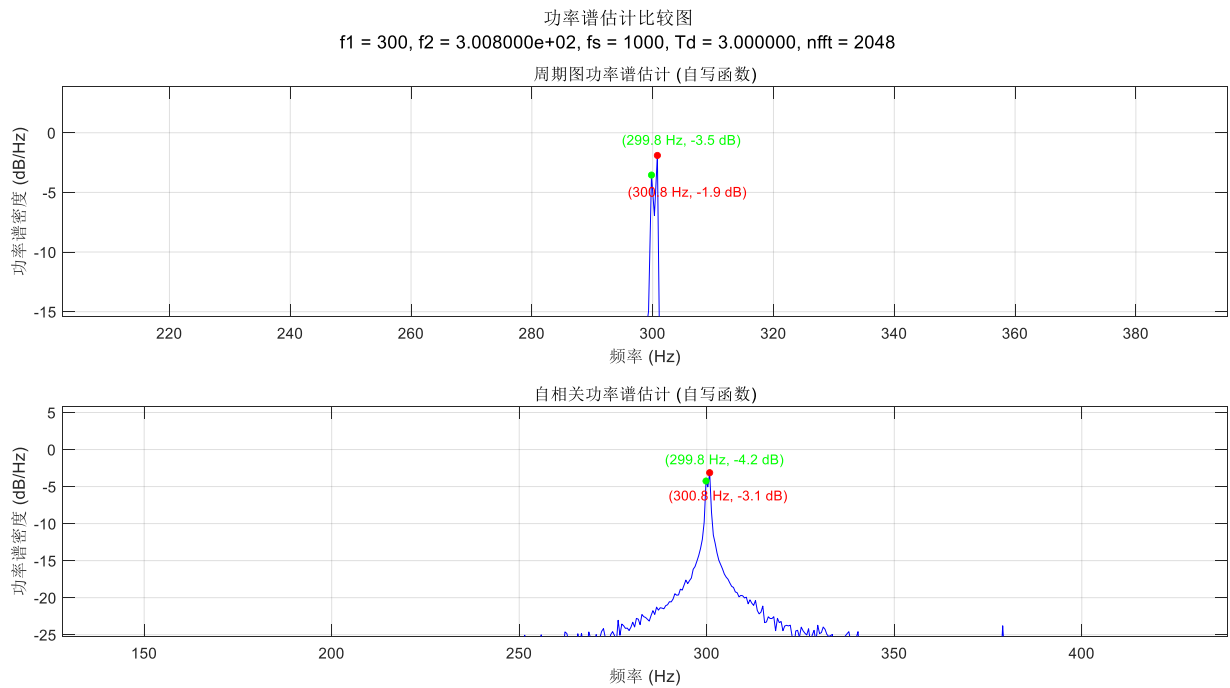
在自相关函数法中，功率谱估计是通过先估计有限长的自相关函数，再对其做傅里叶变换得到的。有限长的自相关函数在频域上对应的是一个粗糙的谱，同样存在采样点有限的问题。如果在自相关函数序列后补零，就能在频域上得到更密集的采样点，使得谱曲线更平滑。这里的逻辑与周期图法类似：补零并不会改变自相关函数本身所包含的信息量，也不会增加有效的时域样本长度，因此频率分辨率依旧受限于原始数据长度。但补零可以让傅里叶变换的结果在频率轴上更细致，从而更直观地展示谱的形状。

因此，补零的本质作用是**插值**，它改善的是显示效果和数值计算的便利性，而不是估计本身的精度。对于周期图法，补零让谱曲线更光滑，便于观察峰值位置；对于自相关函数法，补零让傅里叶变换的结果更细腻，便于分析谱的整体形态。在实际应用中，研究者往往会在 FFT 前补零，以便得到更清晰的谱图，但他们也清楚这并不能突破由数据长度决定的分辨率极限。换句话说，补零是“看得更清楚”，而不是“分得更开”。

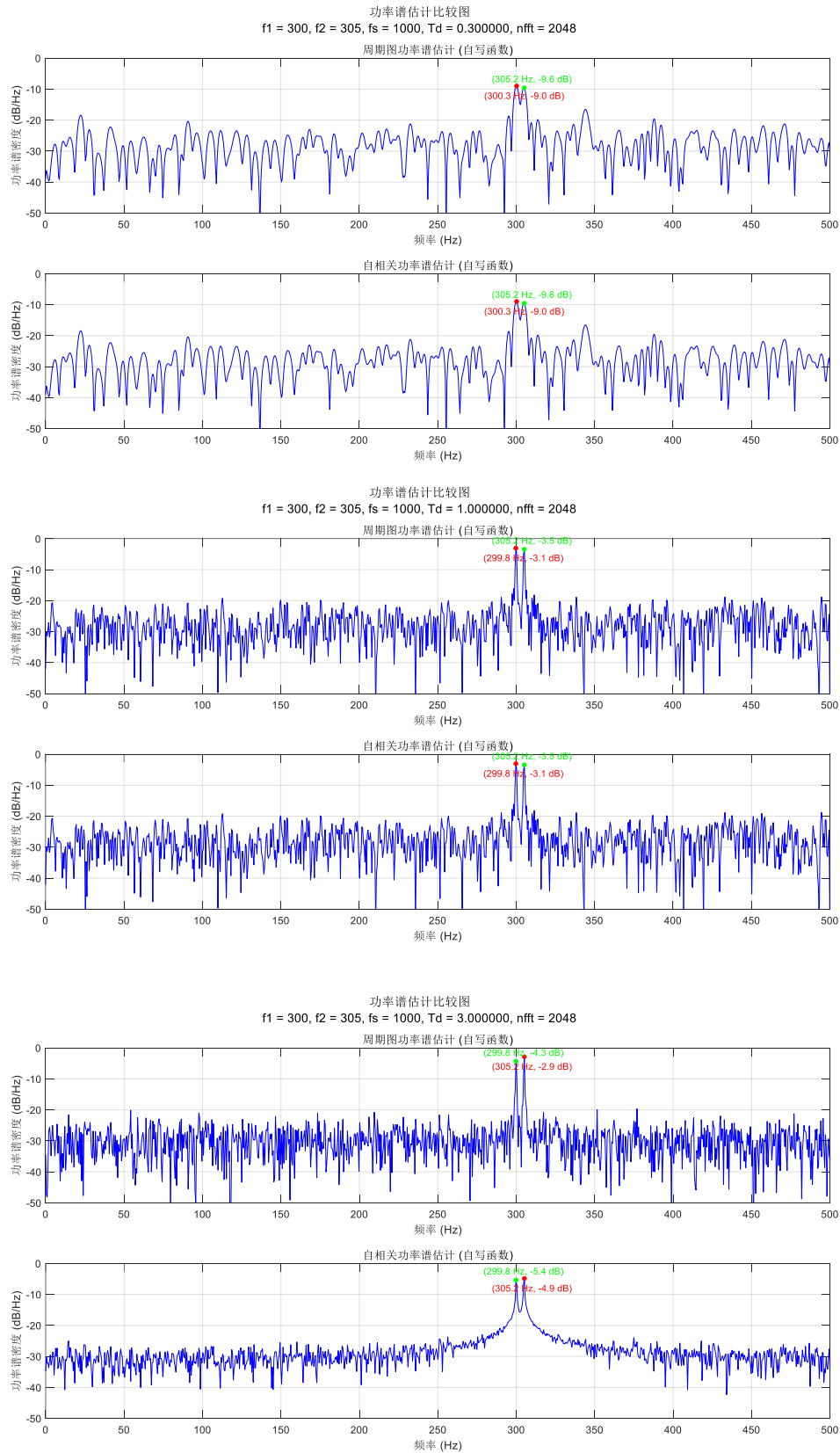
因此，与分辨率真正有关的是 T_d ，我们把 T_d 从 0.3 改为 3.0，让分辨率变为 0.333Hz，这时候，可以清晰地分辨出来 300 和 302Hz 两条谱线。

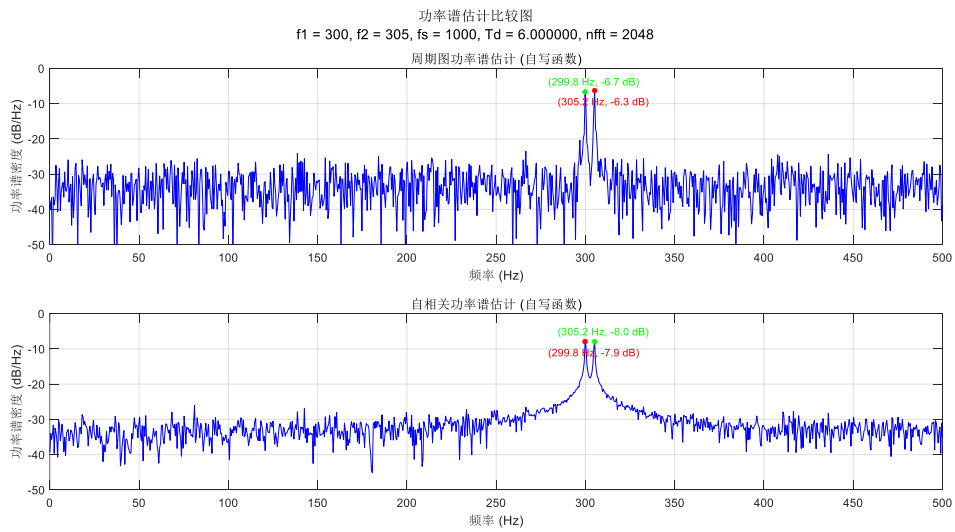


把 f1,f2 改为 300Hz 和 300.8Hz，在分辨率为 0.33Hz 的情况下也能分辨清楚。



下面，我们来观察随着 T_d 的增加，也就是分辨率 $1/T_d$ 变小的过程中，两条谱线有什么变化和区别。





可以看到，当信号持续时间 T_d 增加时，等效的频率分辨率 $1/T_d$ 逐渐变小，这意味着在频域上能够更清晰地区分相邻的频率分量。对于你这里的 $f_1=300\text{Hz}$ 和 $f_2=305\text{Hz}$ ，两条谱线之间的间隔只有 5Hz ，如果 T_d 较短，那么周期图和自相关法得到的谱峰会显得比较宽，两个峰值可能会部分重叠甚至难以区分，看起来像是一条模糊的单峰。随着 T_d 逐渐拉长，主瓣宽度收窄，旁瓣结构也更细致，两个峰值之间的凹陷逐渐显现出来，最终在频谱上能够清楚地看到两条独立的谱线。

因此，随着 T_d 增大，两种方法都会表现出分辨率提升的趋势，谱峰逐渐分离，频率估计更准确。区别在于周期图更敏感，峰值**锐化**得快，但在噪声下可能抖动明显；自相关法则更稳健，谱线更**圆滑**，抗噪性强，但分辨率提升的速度稍慢一些。最终在足够长的数据下，两者都会把 300Hz 和 305Hz 的谱线分开，只是呈现出来的细节和峰值形态有所差异。这个运行结果与前面的分析相匹配。

四、实验收获和体会

本次功率谱实验让我对周期图法与自相关法的差异有了更直观的认识。通过自行编写函数避免 MATLAB 内置优化，我验证了理论复杂度与实际运算时间的一致性：周期图法仅需一次 FFT，复杂度约为 $O(N\log N)$ ，在性能上明显优于自相关法；而自相关法需先计算长度约 $2N-1$ 的相关序列，再做 FFT，计算量更大，耗时更长。实验还清楚展示了频率分辨率与观测时长 T_d 的关系，分辨率受限于 $1/T_d$ ，补零操作只能增加频率采样点密度，使谱线更平滑，却无法突破分辨率极限。当 f_1 与 f_2 间隔小于 $1/T_d$ 时，即便增大 $nfft$ 也无法分辨，只有延长信号时长才能真正区分相邻频率分量。对比两种方法的谱形态，周期图法峰值尖锐，定位灵敏，但在噪声下抖动较大；

自相关法谱线更平滑，抗噪性强，但分辨率提升较慢。这说明周期图法适合快速频率检测，而自相关法更适合平滑估计和噪声环境。通过本实验，我不仅掌握了两种方法的实现细节和性能差异，更加深了对维纳辛钦定理、自相关与卷积关系以及频率分辨率本质的理解，体会到理论与实践结合的重要性。