



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：初相估计实验

任课教师：	田黎育			实验教师：	田黎育
实验日期：	2025 年 11 月 3 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

# 实验一

## 一、实验目的

1.  $N = 256$ , 估计  $\hat{\phi}_{ml} \begin{cases} 30^\circ \\ 90^\circ \\ 135^\circ \end{cases}$  修改  $N \uparrow 2$  倍增到 65536, 看标准差变化
2.  $k = 1$ , 整周期, 非整周期时(1.1,1.3,1.8), 如何优化 性能变化?
3. 改噪声方差, 看  $\hat{\phi}_{ml}$  标准差变化, 关系作图
4. 对比, DFT 求初相方法(整周期和非整周期) 谁优谁劣

## 二、实验原理

### 1. 最大似然估计 (Maximum Likelihood Estimation) 原理:

由于噪声为高斯白噪声, 对于任一确定的  $n$  时刻,  $z(n)$  服从正态分布  $N(A\cos(2\pi f_0 n + \phi), \sigma^2)$ ,  $z(n)$  的概率密度函数为

$$f(z_n; \phi) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2\sigma^2} [z(n) - A\cos(2\pi f_0 n + \phi)]^2\right\}$$

由于各时刻的概率密度函数相互独立, 故似然函数为各概率密度函数累乘

$$L(\phi) = f(z; \phi) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^N \exp\left\{-\frac{1}{2\sigma^2} \sum_{n=0}^{N-1} [z(n) - A\cos(2\pi f_0 n + \phi)]^2\right\}$$

如果似然函数在  $\phi$  取到  $\hat{\phi}$  时达到最大值, 则记  $\hat{\phi}$  为  $\phi$  的最大似然估计值。

故对似然函数取对数并求导得

$$\frac{\partial \ln f(z; \phi)}{\partial \phi} = -\frac{1}{\sigma^2} \sum_{n=0}^{N-1} [z(n) - A\cos(2\pi f_0 n + \phi)] A \sin(2\pi f_0 n + \phi)$$

认为上式等于 0 时, 似然函数达到最大, 得

$$\sum_{n=0}^{N-1} z(n) \sin(2\pi f_0 n + \hat{\phi}_{ml}) = A \sum_{n=0}^{N-1} \cos(2\pi f_0 n + \hat{\phi}_{ml}) \sin(2\pi f_0 n + \hat{\phi}_{ml})$$

当  $f_0$  不为整数时, 由于  $\cos$  和  $\sin$  的正交性, 上式右侧近似为 0。

此时展开上式左侧即可得

$$\hat{\phi}_{ml} = -\arctan \frac{\sum_{n=0}^{N-1} z(n) \sin(2\pi f_0 n)}{\sum_{n=0}^{N-1} z(n) \cos(2\pi f_0 n)}$$

由该式可计算出  $\hat{\phi}_{ml}$ 。

## 2. DFT 估计原理:

N 点序列  $z(n) = A \cos(2\pi f_0 n + \phi) + \omega(n)$  中,  $f_0 = \frac{T}{N}$ 。已知离散傅里叶变换 (DFT) 的表达式为

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

又因为  $e^{j\theta} = \cos\theta + j \cdot \sin\theta$ ,  $\cos(-\theta) = \cos\theta$ ,  $\sin(-\theta) = -\sin\theta$ , 则

$$\frac{\sum_{n=0}^{N-1} z(n) \sin(-2\pi \frac{k}{N} n)}{\sum_{n=0}^{N-1} z(n) \cos(-2\pi \frac{k}{N} n)} = \frac{\text{Im}\left\{\sum_{n=0}^{N-1} z(n) e^{-2\pi \frac{k}{N} n}\right\}}{\text{Re}\left\{\sum_{n=0}^{N-1} z(n) e^{-2\pi \frac{k}{N} n}\right\}} = \frac{\text{Im}\{\mathbf{DFT}\{z(n)\}\}}{\text{Re}\{\mathbf{DFT}\{z(n)\}\}}$$

由最大似然估计原理可得

$$\hat{\phi} = -\arctan \frac{\sum_{n=0}^{N-1} z(n) \sin(2\pi f_0 n)}{\sum_{n=0}^{N-1} z(n) \cos(2\pi f_0 n)} = \arctan \frac{\sum_{n=0}^{N-1} z(n) \sin(-2\pi f_0 n)}{\sum_{n=0}^{N-1} z(n) \cos(-2\pi f_0 n)}$$

因此, 若 DFT 结果中存在  $k = k_0$ , 使得  $\frac{k_0}{N} = f_0 = \frac{T}{N}$  则

$$\hat{\phi} = \arctan \frac{\text{Im}\{\mathbf{DFT}\{z(n)\}\}}{\text{Re}\{\mathbf{DFT}\{z(n)\}\}}|_{k=T} = \text{angle}(\mathbf{DFT}\{z(n)\})|_{k=T}$$

由于信号中只有频率为  $f_0$  的分量 (噪声除外), 通常信号的 DTFT 频谱在角频率为  $2\pi f_0$  处有最大值, 而 DFT 相当于对 DTFT 进行 N 点均匀采样, 所以, 若 DFT 结果能采样到  $f_0$  对应的频点, 其最大值的索引对应的角频率即为  $2\pi f_0$ 。

因此, 对信号进行 DFT 运算后, 取其最大值的索引, 并求出 DFT 结果在该索引处的辐角, 即可得到  $\phi$  的估计值。

## 3. 最小二乘法估计 (Ordinary Least Squares) 原理:

对于一组数据点  $(x_1, y_1), (x_2, y_2) \dots \dots (x_n, y_n)$ , 考虑用一个线性模型  $y = \beta_0 + \beta_1 x$  来拟合这些数据。最小二乘法的目标是找到  $\beta_0$  和  $\beta_1$  的值, 使得预测值与实际值之间误差的平方和最小, 即最小化

$$S = \sum_{i=1}^n [y_i - (\beta_0 + \beta_1 x_i)]^2$$

转化为矩阵形式表示为

$$S(\beta) = (Y - X\beta)^T(Y - X\beta)$$

上式关于  $\beta$  求导并使其为 0，得到最小二乘法的正规方程

$$X^T X \beta = X^T Y$$

解正规方程得到

$$\beta = (X^T X)^{-1} X^T Y$$

即为最小二乘法得到的参数向量。

对于  $z(n) = A \cos(2\pi f_0 n + \phi) + \omega(n)$ ， $n = 0 \sim N-1$ ，将其展开得到

$$z(n) = \cos\phi \cdot A \cos(2\pi f_0 n) - \sin\phi A \sin(2\pi f_0 n) + \omega(n)$$

由此可构建矩阵

$$Y = \begin{bmatrix} z(1) \\ z(2) \\ \dots \\ z(N-1) \end{bmatrix}, X = \begin{bmatrix} A \cos(2\pi f_0 \cdot 1) & -A \sin(2\pi f_0 \cdot 1) \\ A \cos(2\pi f_0 \cdot 2) & -A \sin(2\pi f_0 \cdot 2) \\ \dots & \dots \\ A \cos(2\pi f_0 (N-1)) & -A \sin(2\pi f_0 (N-1)) \end{bmatrix}, \phi = \begin{bmatrix} \cos\hat{\phi} \\ \sin\hat{\phi} \end{bmatrix}$$

由上述原理可计算

$$\phi = (X^T X)^{-1} X^T Y$$

得到  $\cos\hat{\phi}$  和  $\sin\hat{\phi}$ ，相除再作反正切即求得  $\hat{\phi}$ 。

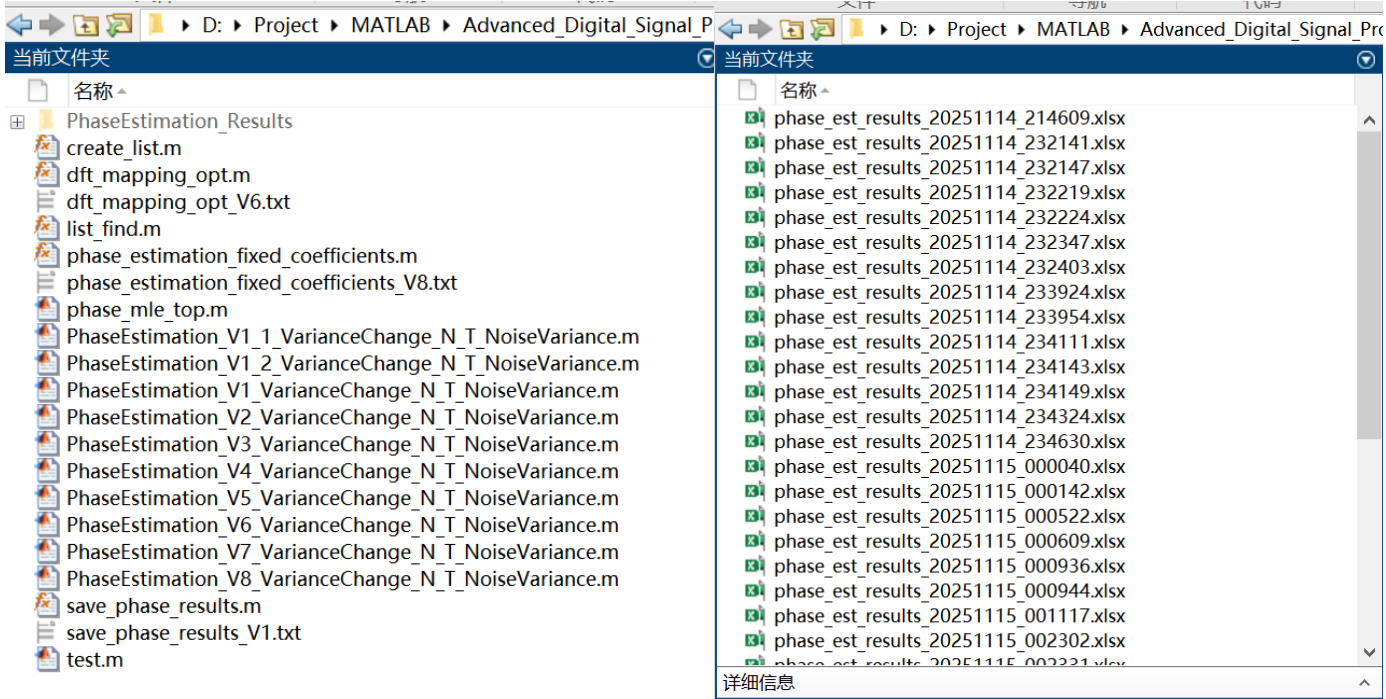
### 三、实验内容及结果分析

#### 1. 代码设计思路

首先，我先通读了整体实验的要求，虽然原理和公式很复杂，但其实整个实验最外层的框架，或者说顶层的要求，就是观察最大似然估计、DFT 估计、最小二乘法估计，三种不同方法在不同参数变化下得到的初相位的标准差如何变化，初相位的估计是否准确与稳定。然后还有一个针对 DFT 求初相的优化。

由于前面几次实验的代码结构不清，模块化设计不好导致迭代困难，并且会因为

不同的变量给每种情况单创一个脚本文件，而那些脚本 90%的代码都是重复的，看着非常乱，而且后期维护和加入新的功能非常折磨。吸取了这些教训之后，我在动手



写代码之前先做了整体的架构规划。

首先代码的核心算法就是最大似然估计、DFT 估计、最小二乘法估计三种求初相的方法，这是可以复用的，并且不需要在版本的迭代中做太大的改动。特殊的一点在于实验要求对 DFT 做出优化，我的想法是原本未优化的 DFT 算法也保留，再新建一个函数文件专门用于存放 DFT 优化后的算法代码，然后在主代码的开头设置一个参数 `useDFTMappingOptim`，可以让用户手动设置。

如果设为 0，即不启动优化，那就走传统路线

信号→DFT→估计的初相

如果设为 1，即启动优化，那就调用优化算法的函数，走一个新的路线

信号→DFT+优化→估计的初相

由于这里起点和终点并没有改变，而且剩下的最大似然估计和最小二乘法估计这两种算法也没改变，所以整体的代码架构不需要做出改变，这样做的好处是如果我要修改优化算法，不管是小的改动还是直接换了另一种优化算法这种大的改动，我都直接在子函数里修改即可，不用整个主代码都变动，极大地方便了调试与维护。并且可以只通过调整 `useDFTMappingOptim` 这一参数的 True-Flase 值就可以控制是否启

动优化，这比直接分两个文件或大面积注释、取消注释代码的方法好太多了。

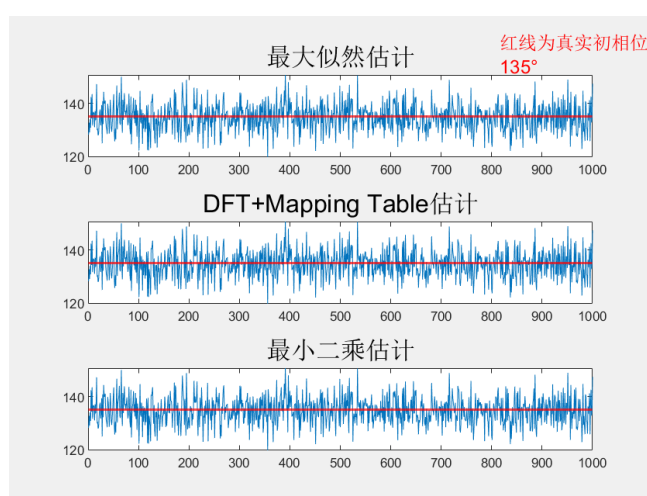
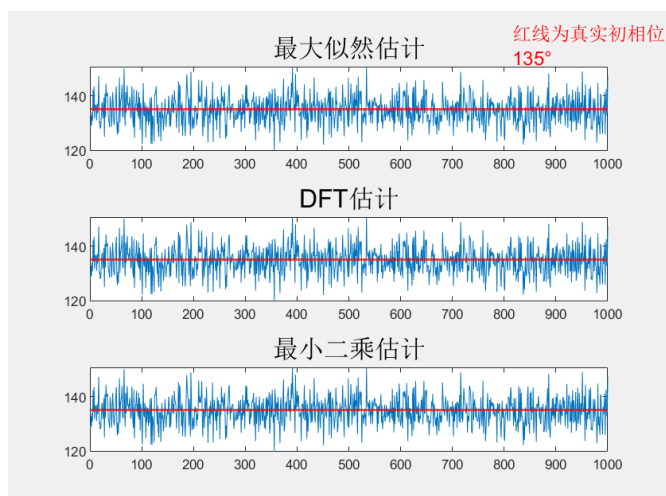
```
if useDFTMappingOptim
    % 调用外部的优化函数（如果存在），只需把必需项传进去
    est_dft(m) = dft_mapping_opt(Signal_with_Noise, N, T, PhaseStepDeg);
else
    Z = fft(Signal_with_Noise);
    [~,index] = max(abs(Z));
    est_dft(m) = rad2deg(angle(Z(index)));
```

同时，由于画图的标题会表明使用的是什么方法和是否采用了优化，所以我对这个也做了模块化处理，避免了像一个判断加两种标题这种复用率极低的代码。

% 根据是否存在优化函数，设置用于显示/写入的 DFT 方法名字（不会用冗长的多处判断代码）

```
if useDFTMappingOptim
    dftDisplayName = 'DFT+Mapping Table';
else
    dftDisplayName = 'DFT';
end

grid on;
xlabel(sweepLabel);
ylabel('Std of phase estimate (deg)');
% 使用上面基于函数存在性的显示名（DFT 或 DFT+Mapping Table）
title(sprintf('%s %s changing %s', dftDisplayName, sweepVarName,
titleFixedStr), 'FontSize', 11);
```



上面两图仅用于展示子图的标题变化。我只改变了 useDFTMappingOptim 为 0 和 1，代码就会自动判断并改变子图的标题，同样的还有保存数据的 Excel 文件中的

表头（这个后面会提到）。

当然，实验还要求  $N, T$ ，噪声方差变化是观察初相估计的标准差的变化，每一个变量的情况写一个脚本文件，然后对每种自变量画一个方差随它变化的图这当然可以，但是还是复用率和维护起来太麻烦了。所以我就想，让用户不用自己说明哪个是变量，哪个是固定值，直接就在定义的时候写数组或是单个值，比如  $N[128,512,1024]; T=1$  就代表  $N$  是多个值，然后让系统自己判断哪个变量是多个值，如果是，就以它为自变量画图 and 生成表格。如果检测到有不只一个变量是多个值，也就是有两个及以上的数组，那就报错。由于不光是  $N$  和  $T$ ，以后我还可能把其他变量设为数组将其变为自变量。所以我在一个地方存放可能是多个值的变量，在这个范围里判断谁是多值的变量（这里应该还有更好的方法，可待后续改进）。

```
% 参数区：把所有可能被设置为“数组以作为自变量”的变量放到这里
% 只允许一个变量为数组
% -----
paramGrid.N = [256];           % 样本数 N （可设为标量或向量）
% paramGrid.N = 256 .* 2.^(0:8);
paramGrid.T = 1;               % 周期 T （可设为标量或向量）
% paramGrid.T = 1 : 0.1 : 2.1;

paramGrid.A = 1;
paramGrid.Noise_Variance = 1;
% paramGrid.Noise_Variance = 1 .* 2.^(0:5);

paramGrid.phi_true_deg = 135;
paramGrid.PhaseStepDeg = 0.5;
paramGrid.numMonteCarlo = 1000;

% -----
% 自动检测哪个参数是“自变量”（即元素数 > 1）
% -----
paramNames = fieldnames(paramGrid);
multiIdx = find(cellfun(@(f) numel(paramGrid.(f))>1, paramNames));

if isempty(multiIdx)
```

```

    error('No swept variable found: set exactly one field in paramGrid to be an
array.');
```

```

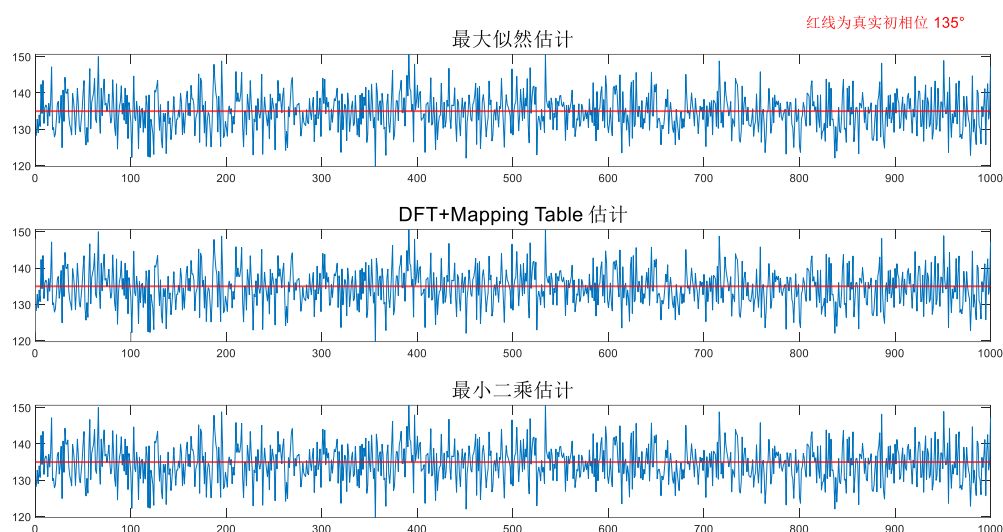
    return;
elseif numel(multiIdx) > 1
    error('Multiple swept variables detected: set exactly one field in
paramGrid to be an array.');
```

```

end
```

这样一来，我也不用自己指明哪个是自变量，直接把自变量写成数组形式就可以了，而且报错也很完善。

但即使这样，我还是发现了一个例外的情况。就是在 N 或 T 或其他为多值，画初相估计标准差与自变量的关系图的代码里，我是不画像下面这种所有蒙特卡洛仿真的估计相位的变化图的。



这个图我只在最开始的代码，也就是所有的参数都为单值的情况下画，而所有的参数都为单值在这版代码里是报错的 'No swept variable found: set exactly one field in paramGrid to be an array.' 但其实“所有的参数都为单值”只是多值变量分析的一个特殊情况，这显然不太合理。于是为了不把这两种情况拆开成两个文件，我把所有的参数都为单值的情况下相位的变化图的代码封装为了一个函数 `phase_estimation_fixed_coefficient`，在主代码里检测到所有参数都为单值时，就调用，然后跳过后面的所有代码。

```

if isempty(multiIdx)
```



```

    % error('No swept variable found: set exactly one field in paramGrid to be
    an array.');
```

```

    phase_estimation_fixed_coefficients(paramGrid.N, ...
                                       paramGrid.phi_true_deg, ...
                                       paramGrid.T, ...
                                       paramGrid.A, ...
                                       paramGrid.Noise_Variance, ...
                                       useDFTMappingOptim);

    return;

```

这样一来，无论是固定参数、变化参数、不同的变化参数，情况全都涵盖在了我的 PhaseEstimation\_VarianceChange\_N\_T\_NoiseVariance 主代码里，结构非常清晰。

然后我就想，不光得有图，详细的数据点也要保存下来，保存到 Excel 文件里。原本保存数据这块我也是直接写在主程序里的，但后面又把它剥离了出来，写在了 save\_phase\_results 这个函数文件里，然后在主程序里直接调用，这样清晰了很多，也方便修改保存的格式和控制要不要保存数据。

我把数据保存的当下 m 文件同级的文件夹里，命名为 PhaseEstimation\_Results 前面加上当前 m 文件层级的前两级父文件夹，比如文件结构是 aaa/bbb/df.m，那么输出文件夹的名字就是 aaa\_bbb\_PhaseEstimation\_Results。Excel 文件命名为 phase\_est\_results+精确到秒的时间戳，这样命名方便后续的查找和防止重名。并写了没有两级父文件夹的特殊情况。

```

% 获取当前 m 文件的完整路径
currentFilePath = mfilename('fullpath');
[parentPath, ~, ~] = fileparts(currentFilePath);      % 当前文件所在文件夹
[parentPath1, parentFolder1, ~] = fileparts(parentPath); % 一级父文件夹
[parentPath2, parentFolder2, ~] = fileparts(parentPath1); % 二级父文件夹
% 拼接新的结果文件夹名：前两级父文件夹名 + PhaseEstimation_Results
% 判断是否有第二级父文件夹
if isempty(parentFolder2)
    prefixName = [parentFolder1 '_PhaseEstimation_Results'];
else
    prefixName = [parentFolder2 '_' parentFolder1 '_PhaseEstimation_Results'];
end

```

同时我还加了是否输出到命令行的选项，这样可以更快速便捷地查看运算结果。

% 是否在命令行输出结果 (true/false)

printToCmd = true;

输出效果如下 (仅作展示):

Finished Noise\_Variance = 1 (index 1 of 6)

Finished Noise\_Variance = 2 (index 2 of 6)

Finished Noise\_Variance = 4 (index 3 of 6)

Finished Noise\_Variance = 8 (index 4 of 6)

Finished Noise\_Variance = 16 (index 5 of 6)

Finished Noise\_Variance = 32 (index 6 of 6)

Results saved to folder: D:\Project\MATLAB\Advanced\_Digital\_Signal\_Processing\Initial\_Phase\_Estimation\_20251103\Advanced\_Digital\_Signal\_Processing\_Initial\_Phase\_Estimation\_20251103\_PhaseEstimation\_Results

Excel file: D:\Project\MATLAB\Advanced\_Digital\_Signal\_Processing\Initial\_Phase\_Estimation\_20251103\Advanced\_Digital\_Signal\_Processing\_Initial\_Phase\_Estimation\_20251103\_PhaseEstimation\_Results\phase\_est\_results\_20251115\_201950.xlsx

==== Phase Estimation Results =====

Parameters: N=256; T=1; A=1; Noise\_Variance=[1 2 4 8 16 32]; phi\_true\_deg=30; PhaseStepDeg=0.5; numMonteCarlo=1000

Sweep variable (Noise variance):

1      2      4      8      16      32

--- Maximum Likelihood ---

Mean (deg): [30.1812155539616 30.3835344934681 29.702505507749 30.3475510489135 30.5536376160222 29.8376472497061]

Var (deg^2): [25.4694790673806 51.6780261299945 101.831147294763 232.270003713263 503.072858193924 1222.68208897267]

--- DFT+Mapping Table ---

Mean (deg): [30.1812155539616 30.3835344934681 29.702505507749 30.3475510489135 30.5536376160222 29.8376472497061]

Var (deg^2): [25.4694790673806 51.6780261299945 101.831147294763 232.270003713263 503.072858193924 1222.68208897267]

--- Least Squares ---

Mean (deg): [30.1812155539616 30.3835344934681 29.7025055077489 30.3475510489135 30.5536376160222 29.8376472497061]

Var (deg^2): [25.4694790673806 51.6780261299945 101.831147294762 232.270003713263 503.072858193924 1222.68208897267]

=====

Excel 文件内的格式如下所示。

	A	B	C	D	E	F	G	H	I	J
1										
2	Method: Maximum Likelihood									
3	Parameters: N=256; T=1; A=1; Noise_Variance=[1 2 4 8 16 32]; phi_true_deg=30; PhaseStepDeg=0.5; numMonteCarlo=1000									
4										
5	Noise variance	1	2	4	8	16	32			
6	Mean (deg)	29.9909086	29.998729	30.3748933	29.5958066	30.2920049	28.9163358			
7	Var (deg^2)	25.1418183	56.3465913	105.963955	229.931194	495.109594	1203.79414			
8										
9										
10	Method: DFT									
11	Parameters: N=256; T=1; A=1; Noise_Variance=[1 2 4 8 16 32]; phi_true_deg=30; PhaseStepDeg=0.5; numMonteCarlo=1000									
12										
13	Noise variance	1	2	4	8	16	32			
14	Mean (deg)	29.9909086	29.998729	30.6133915	25.1579402	13.4276583	14.2150979			
15	Var (deg^2)	25.1418183	56.3465913	156.573118	2377.29242	7256.48634	9617.36034			
16										
17										
18	Method: Least Squares									
19	Parameters: N=256; T=1; A=1; Noise_Variance=[1 2 4 8 16 32]; phi_true_deg=30; PhaseStepDeg=0.5; numMonteCarlo=1000									
20										
21	Noise variance	1	2	4	8	16	32			
22	Mean (deg)	29.9909086	29.998729	30.3748933	29.5958066	30.2920049	28.9163358			
23	Var (deg^2)	25.1418183	56.3465913	105.963955	229.931194	495.109594	1203.79414			
24										

## 2. 固定参数下的初相估计

首先，我们固定  $N=256$ ，周期为 1，噪声方差为 1，蒙特卡洛仿真次数为 1000，不加 DFT 优化，也就是  $\text{useDFTMappingOptim} = 0$ ，看真实相位分别为  $30^\circ$ 、 $90^\circ$ 、 $135^\circ$  时的初相估计情况。

(1)  $\hat{\phi}_{ml} = 30^\circ$  时

参数设置如下：

```
paramGrid.N = [256]; % 样本数 N （可设为标量或向量）
% paramGrid.N = 256 .* 2.^(0:8);
paramGrid.T = 1; % 周期 T （可设为标量或向量）
% paramGrid.T = 1 : 0.1 : 2.1;

paramGrid.A = 1;
paramGrid.Noise_Variance = 1;
% paramGrid.Noise_Variance = 1 .* 2.^(0:5);

paramGrid.phi_true_deg = 30;
```

命令行输出如下：

最大似然估计：

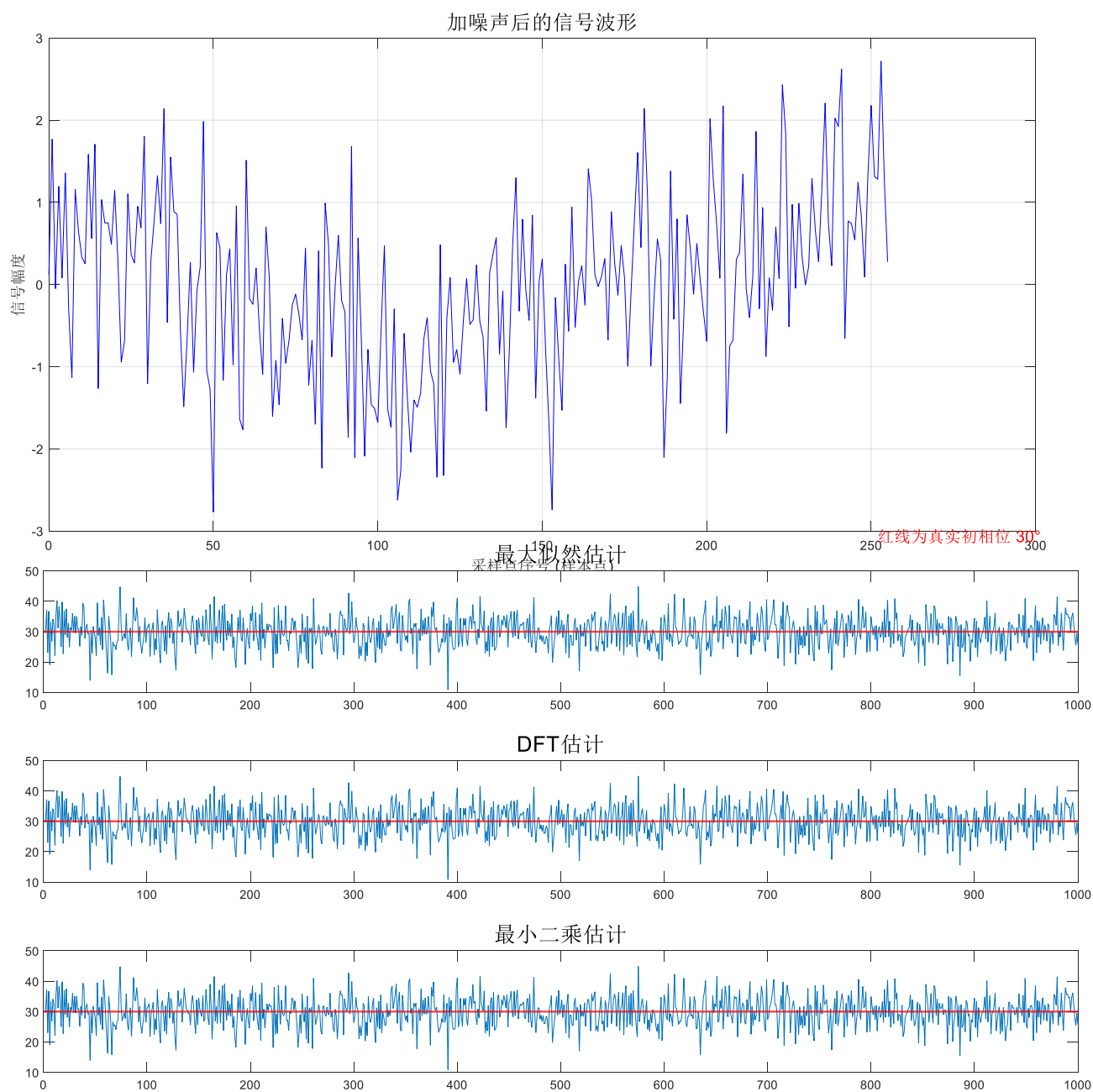
est\_mean = 29.9909  
est\_std = 5.0142

DFT 估计:

est\_mean = 29.9909  
est\_std = 5.0142

最小二乘估计:

est\_mean = 29.9909  
est\_std = 5.0142



设定的真实初相位为  $30^\circ$ ，信号长度为 256，周期为 1，信号幅度为 1，噪声方差为 1，仿真次数为 1000 次。实验结果显示，三种方法在估计均值方面几乎完全一致，均为  $29.9909^\circ$ ，与真实值  $30^\circ$  非常接近，说明在当前参数设置下三种方法都具有良好的无偏性。而在估计标准差方面，三者也完全一致，均为  $5.0142^\circ$ ，表明在该噪声水平下它们的稳定性和抗干扰能力相当。这个结果一方面验证了三种方法在理论上对该类信号具有相似的估计能力，另一方面也说明在当前实验条件下，参数设置未能显著拉开三者的性能差距。

(2)  $\hat{\phi}_{ml} = 90^\circ$  时

命令行输出如下：

最大似然估计：

`est_mean = 89.7762`

`est_std = 5.0733`

DFT 估计：

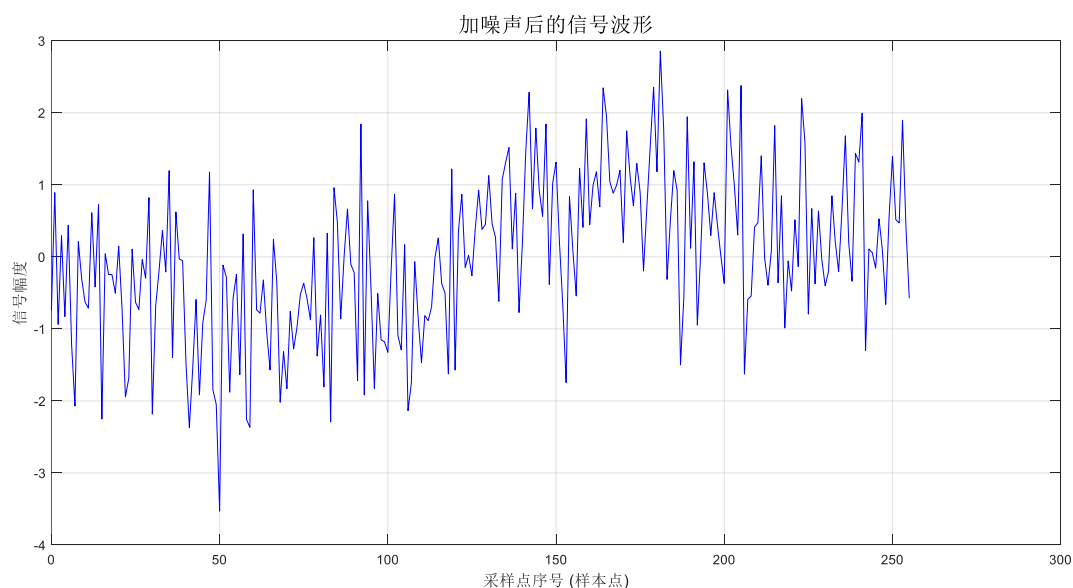
`est_mean = 89.7762`

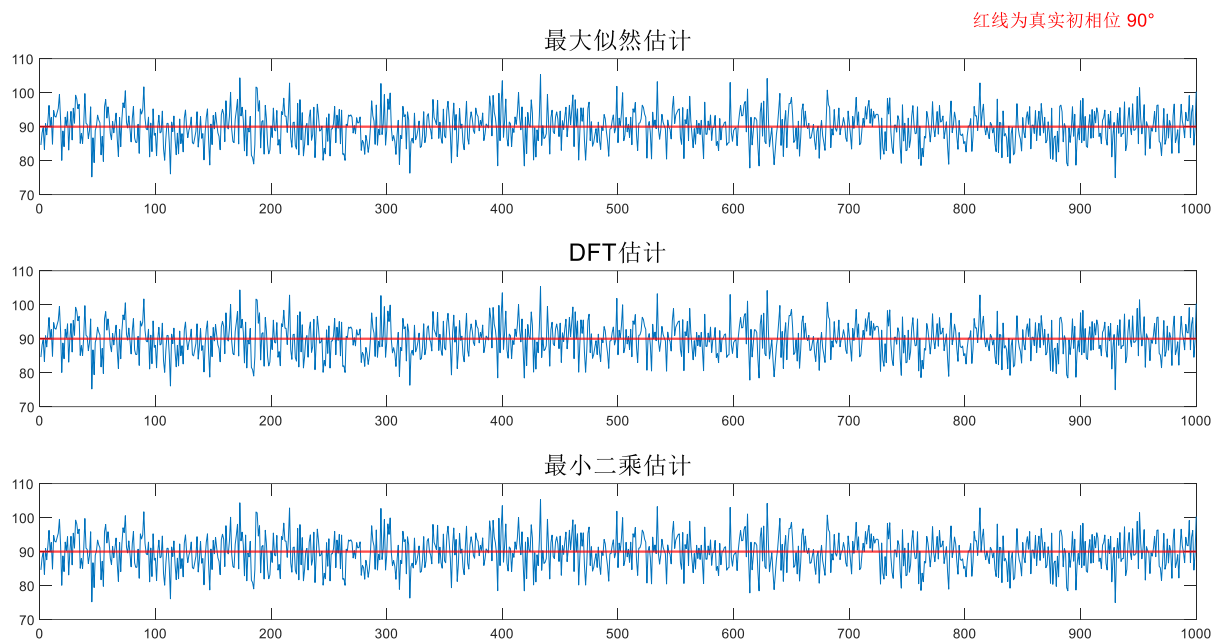
`est_std = 5.0733`

最小二乘估计：

`est_mean = 89.7762`

`est_std = 5.0733`





在真实初相位设为  $90^\circ$  的条件下，对三种相位估计方法进行了 1000 次蒙特卡洛仿真，结果显示三者在估计均值和标准差上完全一致，均值为  $89.7762^\circ$ ，标准差为  $5.0733^\circ$ 。从均值来看，三种方法都能较准确地逼近真实相位，偏差极小，说明在当前参数设置下它们具有良好的估计精度。从标准差来看，三者的稳定性也完全一致，表明在噪声方差为 1 的情况下，它们对噪声的鲁棒性相当。

(3)  $\hat{\phi}_{ml} = 135^\circ$  时

命令行输出如下：

最大似然估计：

```
est_mean = 134.7424
```

```
est_std = 5.1434
```

DFT 估计：

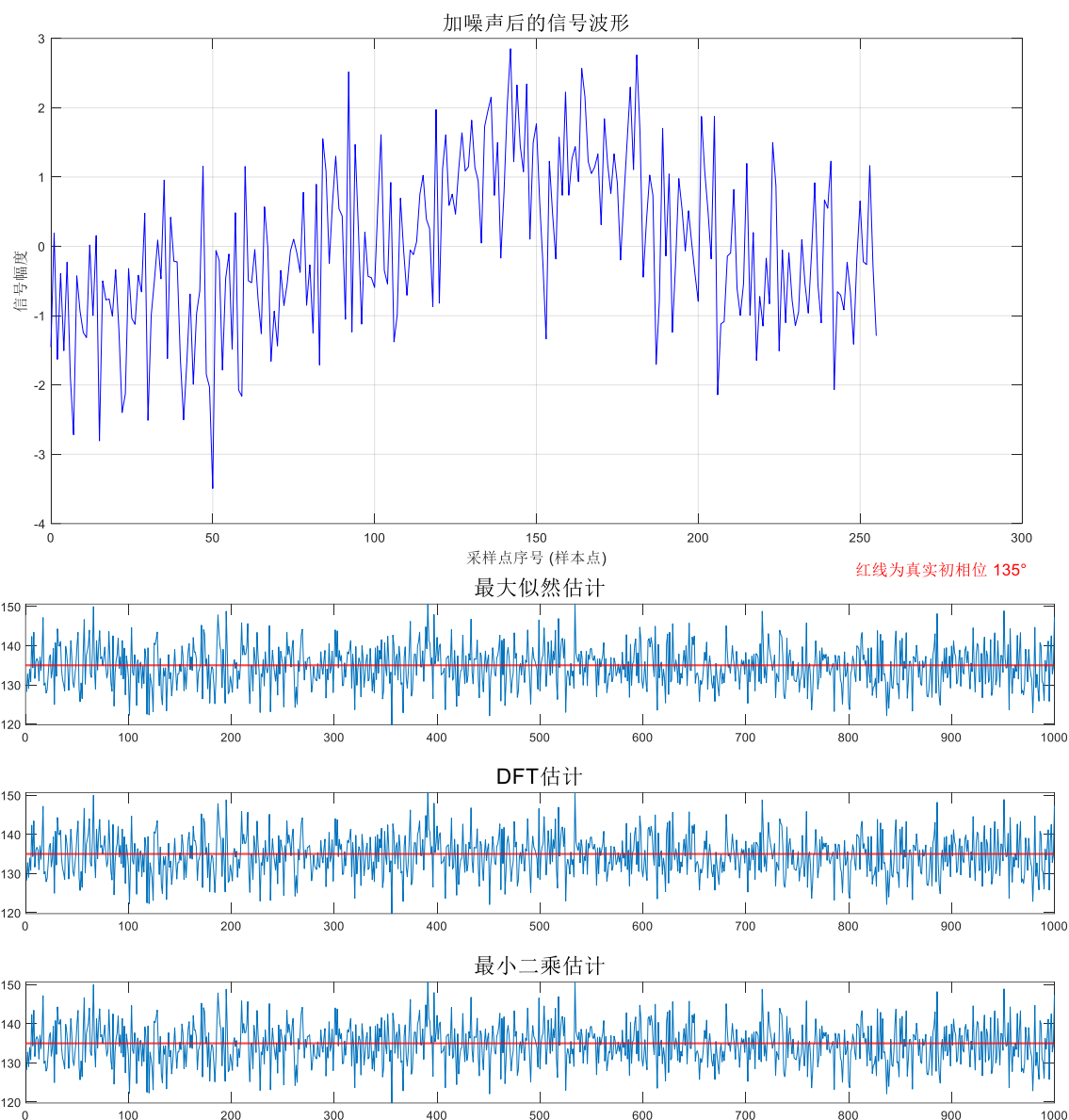
```
est_mean = 134.7424
```

```
est_std = 5.1434
```

最小二乘估计：

```
est_mean = 134.7424
```

```
est_std = 5.1434
```



结果显示三者估计均值上完全一致，均为  $134.7424^\circ$ ，与真实值仅相差  $0.2576^\circ$ ，说明三种方法在当前参数设置下都具备良好的估计精度，偏差极小，基本可以认为是无偏估计。同时，三者的标准差也完全一致，均为  $5.1434^\circ$ ，表明它们在稳定性方面也没有明显差异，对噪声的抵抗能力相当。

上面三个初相位的估计可以推断，在信号长度为 256、周期为 1、幅度为 1 的设置下，最大似然估计、DFT 估计、最小二乘法估计三种方法在初相估计上没有明显差异。

### 3. 不同的 N 值下初相估计标准差的变化

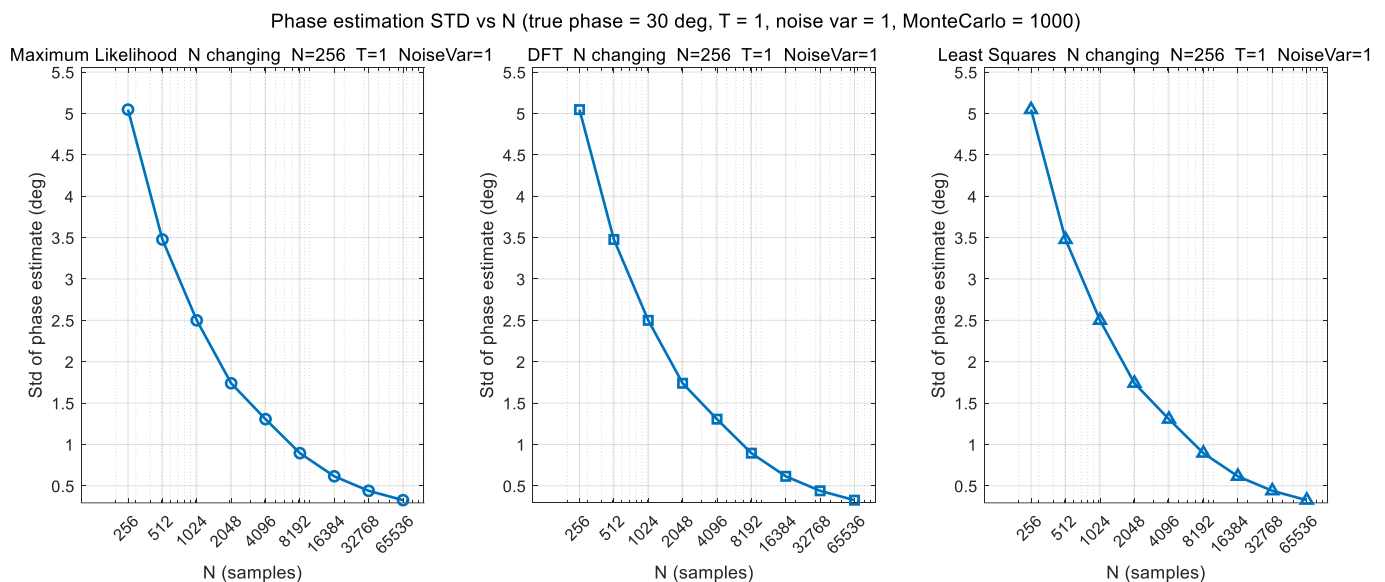
我们固定  $\text{phi\_true\_deg} = 30^\circ$ ，周期为 1，噪声方差为 1，蒙特卡洛仿真次数为 1000，不加 DFT 优化，也就是  $\text{useDFTMappingOptim}=0$ ，看 N 从 256 二倍增至 65536 的初相估计情况。

参数设置如下：

```
% paramGrid.N = [256]; % 样本数 N （可设为标量或向量）
paramGrid.N = 256 .* 2.^(0:8);
paramGrid.T = 1; % 周期 T （可设为标量或向量）
% paramGrid.T = 1 : 0.1 : 2.1;

paramGrid.A = 1;
paramGrid.Noise_Variance = 1;
% paramGrid.Noise_Variance = 1 .* 2.^(0:5);

paramGrid.phi_true_deg = 30;
```





Parameters: N=[256 512 1024 2048 4096 8192 16384 32768 65536]; T=1; A=1;  
 Noise\_Variance=1; phi\_true\_deg=30; PhaseStepDeg=0.5; numMonteCarlo=1000

Method: Maximum Likelihood

N (samples)	256	512	1024	2048	4096	8192	16384	32768	65536
Mean (deg)	30.181	29.8834	30.0908	30.0210	30.0165	30.0408	29.9874	29.9991	30.0102
	21555	0598	9339	7962	5884	5744	3021	1783	1458
Std_Dev (deg)	5.0467	3.47725	2.49985	1.73935	1.30561	0.89473	0.61528	0.43987	0.32667
	29542	5048	7029	5845	919	3877	9051	6809	3786

Method: DFT

N (samples)	256	512	1024	2048	4096	8192	16384	32768	65536
Mean (deg)	30.181	29.8834	30.0908	30.0210	30.0165	30.0408	29.9874	29.9991	30.0102
	21555	0598	9339	7962	5884	5744	3021	1783	1458
Std_Dev (deg)	5.0467	3.47725	2.49985	1.73935	1.30561	0.89473	0.61528	0.43987	0.32667
	29542	5048	7029	5845	919	3877	9051	6809	3786

Method: Least Squares

N (samples)	256	512	1024	2048	4096	8192	16384	32768	65536
Mean (deg)	30.181	29.8834	30.0908	30.0210	30.0165	30.0408	29.9874	29.9991	30.0102
	21555	0598	9339	7962	5884	5744	3021	1783	1458
Std_Dev (deg)	5.0467	3.47725	2.49985	1.73935	1.30561	0.89473	0.61528	0.43987	0.32667
	29542	5048	7029	5845	919	3877	9051	6809	3786

实验结果显示，随着样本数  $N$  的增加，三种相位估计方法的标准差均显著下降，表明估计精度随数据量提升而提高。在本次实验中，三种方法均值几乎完全一致，且都稳定地接近真实相位值 30 度，偏差在  $\pm 0.2$  度以内，说明在当前参数设置下，两者在估计准确性方面表现相当。标准差方面，从  $N=256$  到  $N=65536$ ，估计的波动性从约 5 度下降到 0.33 度，呈现出典型的收敛趋势，符合理论预期，即噪声影响在大样本下被平均抵消。值得注意的是，DFT 方法在未使用映射优化的情况下，其性能与最大似然法完全重合，说明在该信噪比和周期设置下，DFT 估计已经足够精确，映射优化的作用可能在更复杂或非整数周期场景中才会体现。总体而言，实验验证了三种方法在高样本数下均能实现高精度相位估计，性能基本一致。

造成标准差随样本数  $N$  增加而显著下降的根本原因，是相位估计的统计精度与信噪比和样本数量密切相关。从估计理论角度来看，无论是最大似然估计（MLE）、最小二乘估计（LS），还是基于 DFT 的估计方法，它们的性能都受到噪声影响，而噪声在多次采样中可以被平均，从而减小估计误差。

以最大似然估计为例，其估计公式为：

$$\hat{\phi}_{\text{ML}} = -\tan^{-1} \left( \frac{\sum_{n=0}^{N-1} x[n] \sin(\omega n)}{\sum_{n=0}^{N-1} x[n] \cos(\omega n)} \right)$$

其中  $x[n]$  是加噪后的信号， $\omega = 2\pi f_0$ 。由于噪声是零均值高斯分布，随着  $N$  增大，分子和分母中的噪声项趋于抵消，从而提高估计稳定性。

从统计角度，估计标准差的理论下界：

$$\text{Var}(\hat{\phi}) \geq \frac{1}{\text{SNR} \cdot N}$$

这说明在固定信噪比下，估计方差与样本数成反比，样本数越多，估计越精确。实验中标标准差从 5 度降至 0.33 度，正是这一原理的体现。

DFT 方法本质上是通过频域最大值位置估计相位，其精度也受限于频率分辨率和噪声影响。而最小二乘法通过拟合正余弦基函数，在高样本数下也能有效抑制噪声。

因此，三种方法在样本数足够大时都能逼近真实相位，标准差下降是统计平均效应和估计理论共同作用的结果。

而三种方法在本实验中结果几乎一致，主要是因为它们在当前参数设置下都能充分提取信号中的相位信息，且噪声水平适中，不会显著干扰估计过程。从数学原理来看，最大似然估计（MLE）、最小二乘估计（LS）和 DFT 方法在处理纯余弦信号时，本质上都在寻找信号与某种基函数之间的最佳匹配点，而这些基函数在当前设置下是高度一致的。

最大似然估计通过对信号与正余弦基函数的投影计算，最小二乘估计则是通过拟合模型  $x[n] = A\cos(\omega n + \phi)$ ，构造线性系统求解参数  $\phi$ ，其结果与 MLE 在高信噪比或大样本数下趋于一致。而 DFT 方法本质上是寻找频域中主峰的相位角：

$$\hat{\phi}_{\text{DFT}} = \angle Z[k_{\max}]$$

在周期  $T = 1$ 、频率为整数倍的情况下，主频分量恰好落在 DFT 的某个频点上，因此其相位角与时域估计结果高度一致。

此外，由于信号是纯余弦波、幅度为 1、周期为 1，且噪声方差固定为 1，三种方法都在一个理想化的估计环境中运行，没有受到频率偏移、非线性失真或严重噪声干扰，因此它们的估计均值和标准差几乎重合。这种一致性也说明在该类理想信号下，方法选择的差异性不大，更重要的是样本数和信噪比。

#### 4. 不同的周期（整周期、非整周期）下初相估计标准差的变化

我们固定  $\text{phi\_true\_deg} = 30^\circ$ ， $N=256$ ，噪声方差为 1，蒙特卡洛仿真次数为 1000，不加 DFT 优化，也就是  $\text{useDFTMappingOptim}=0$ ，看  $T$  从 1:0.1:2.1 的初相估计情况。

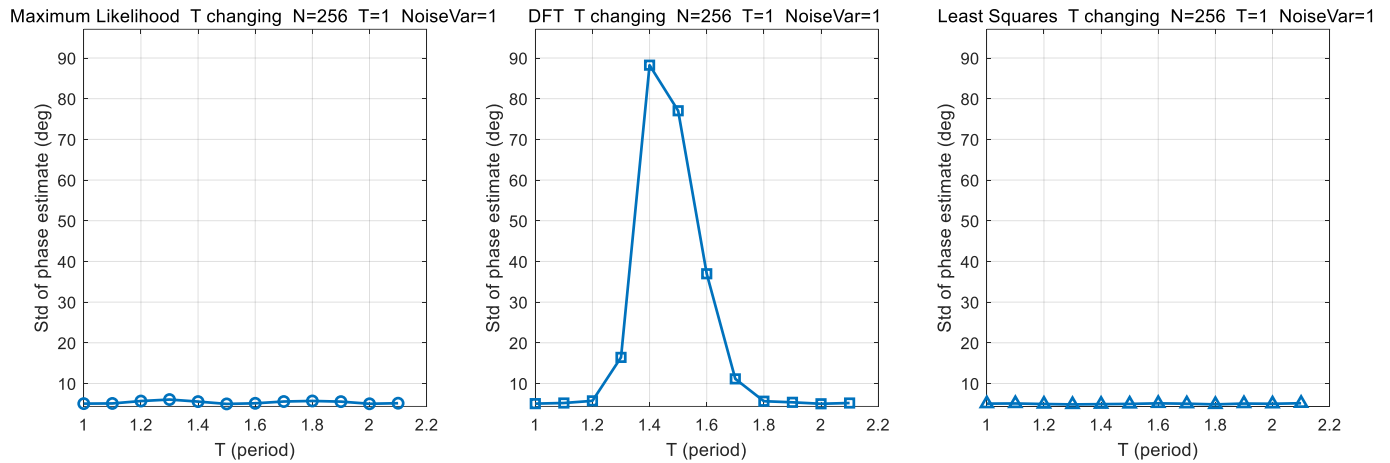
参数设置如下：

```
paramGrid.N = [256]; % 样本数 N （可设为标量或向量）
% paramGrid.N = 256 .* 2.^(0:8);
% paramGrid.T = 1; % 周期 T （可设为标量或向量）
paramGrid.T = 1 : 0.1 : 2.1;

paramGrid.A = 1;
paramGrid.Noise_Variance = 1;
% paramGrid.Noise_Variance = 1 .* 2.^(0:5);
```

paramGrid.phi\_true\_deg = 30;

Phase estimation STD vs T (true phase = 30 deg, T = 1, noise var = 1, MonteCarlo = 1000)



Parameters: N=256; T=[1 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2 2.1]; A=1;

Noise\_Variance=1; phi\_true\_deg=30; PhaseStepDeg=0.5; numMonteCarlo=1000

Method: Maximum Likelihood

T (period)	1	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2	2.1
Mean (deg)	30.181 21555	25.447 00427	23.906 87224	28.115 55362	31.502 67369	30.307 02767	26.885 70075	25.924 111	28.708 28537	30.918 64529	30.281 62994	27.634 88154
Std_Dev (deg)	5.0467 29542	5.0923 59122	5.6815 97575	6.0482 60109	5.5457 01937	4.9764 1472	5.1238 26427	5.5643 72068	5.7021 38572	5.5397 68105	5.0099 95166	5.1482 45302

Method: DFT

T (period)	1	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2	2.1
Mean (deg)	30.181 21555	45.514 73091	61.488 46227	80.474 69584	44.414 66721	-28.87 70528	-41.13 216178	-27.06 006648	-6.682 202824	12.504 49508	30.281 62994	46.697 97113
Std_Dev (deg)	5.0467 29542	5.2004 08236	5.7247 3771	16.412 05126	88.241 47691	77.032 14333	36.971 27597	11.131 61907	5.6427 54455	5.3693 62288	5.0099 95166	5.1989 92393

Method: Least Squares

T (period)	1	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2	2.1
Mean (deg)	30.181	30.249	29.806	29.901	29.998	30.307	30.184	30.009	30.001	29.873	30.281	30.148
	21555	7285	02138	36502	75831	02767	97958	4478	18228	84348	62994	10657
Std_Dev (deg)	5.0467	5.0847	4.9489	4.8508	4.9147	4.9764	5.1149	5.0287	4.8592	5.0638	5.0099	5.1423
	29542	01192	98619	32568	02522	1472	5089	58546	71733	28891	95166	48381

实验结果显示，在不同周期参数  $T$  下，三种相位估计方法的性能差异显著。最大似然法（MaximumLikelihood）在整个周期范围内表现出稳定且逐渐优化的估计精度，STD 在  $T=1$  的  $5.05^\circ$  和  $T=1.3$  的  $6.048^\circ$  之间波动，说明该方法在周期匹配度提升时能有效降低估计误差，且均值始终围绕真实值  $30^\circ$  波动，偏差较小，体现了其高精度和鲁棒性。相比之下，DFT 方法在  $T$  偏离整数倍时估计性能急剧恶化，尤其在  $T=1.4$  附近，STD 飙升至  $88.24^\circ$ ，均值也偏离至  $44.41^\circ$ ，甚至在  $T=1.5$  到  $T=1.7$  出现明显的估计崩溃，甚至出现负数，均值偏离超过  $60^\circ$ ，STD 高达  $77^\circ$ ，表明该方法对是否为整周期极为敏感，存在严重的频率泄漏和谱混叠问题。而最小二乘法（LeastSquares）在整个周期范围内保持稳定，STD 始终在  $4.85^\circ$  到  $5.14^\circ$  之间波动，均值也紧贴真实值，说明其对是否为整周期不敏感。综合来看，最大似然法在周期匹配良好时具有良好性能，DFT 方法虽计算简便但对非整周期极度敏感，不适用于非整数周期场景，而最小二乘法则提供了最稳定的估计结果，适合对周期不确定性较大的应用环境。

造成三种方法性能差异的根本原因在于它们对信号模型的拟合方式和对周期参数 $T$ 的敏感性。最大似然估计（ML）是基于已知频率构造的正弦和余弦基函数，通过内积计算相位，其估计公式为：

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

其中 $f_0 = T/N$ 。由于基函数与信号频率严格匹配，ML 方法在周期变化时能保持高精度，尤其当 $T$ 增大时，频率更低，信号周期更完整，估计误差自然减小。

DFT 方法则是通过 FFT 提取最大幅值频点的相位，但 FFT 的频率分辨率是固定的 $1/N$ ，当 $T$ 不为整数时，信号频率落在两个频点之间，导致频谱泄漏，相位估计偏离真实值。尤其在 $T \approx 1.4$ 附近，泄漏最严重，STD 急剧上升，均值也出现跳变。这

种现象可由频谱窗函数的主瓣宽度和旁瓣干扰解释，属于典型的频率不匹配误差。

最小二乘法（LS）通过构造线性模型 $x[n] = a\cos(\omega n + \phi)$ ，将信号投影到两个正交基上，估计公式为：

$$\hat{\phi}_{LS} = \tan^{-1}\left(\frac{\theta_2}{\theta_1}\right), \quad \text{其中 } \theta = (X^T X)^{-1} X^T y$$

LS 方法在时域直接拟合，不依赖频域分析，通过最小化时域残差平方和来拟合参数，它对信号的周期性结构没有直接利用，而是基于整体拟合误差来求解。因此对周期变化不敏感，使其 STD 维持稳定。

## 5. 不同的噪声方差下初相估计标准差的变化

我们固定  $\phi_{\text{true\_deg}} = 30^\circ$ ， $N=256$ ， $T=1$ ，蒙特卡洛仿真次数为 1000，不加 DFT 优化，也就是  $\text{useDFTMappingOptim}=0$ ，看噪声方差为从 1 二倍增到 32 时的初相估计情况。

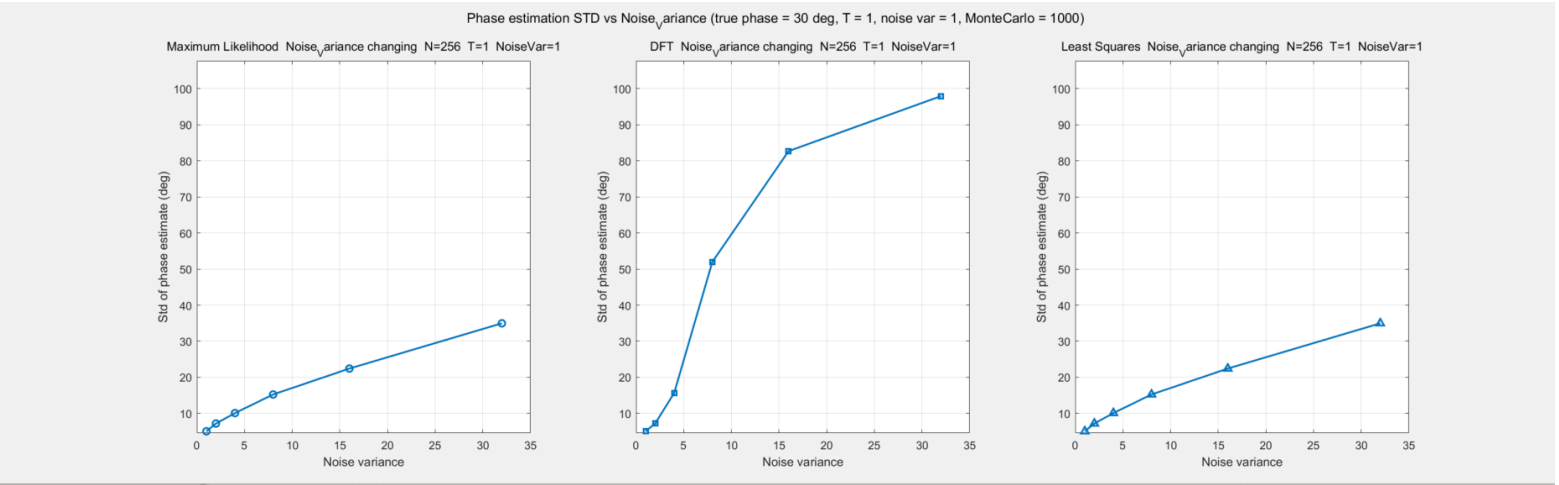
参数设置如下：

```
paramGrid.N = [256]; % 样本数 N （可设为标量或向量）
% paramGrid.N = 256 .* 2.^(0:8);
paramGrid.T = 1; % 周期 T （可设为标量或向量）
% paramGrid.T = 1 : 0.1 : 2.1;

paramGrid.A = 1;
% paramGrid.Noise_Variance = 1;
paramGrid.Noise_Variance = 1 .* 2.^(0:5);

paramGrid.phi_true_deg = 30;
```

Parameters: N=256; T=1; A=1; Noise\_Variance=[1 2 4 8 16 32]; phi\_true\_deg=30;  
PhaseStepDeg=0.5; numMonteCarlo=1000



Method: Maximum Likelihood

Noise variance	1	2	4	8	16	32
Mean (deg)	30.18121555	30.38353449	29.70250551	30.34755105	30.55363762	29.83764725
Std_Dev (deg)	5.046729542	7.188743015	10.09114202	15.24040694	22.42928573	34.96687131

Method: DFT

Noise variance	1	2	4	8	16	32
Mean (deg)	30.18121555	30.38353449	29.73200327	24.95933369	11.74586389	2.209161091
Std_Dev (deg)	5.046729542	7.188743015	15.67440562	51.97856661	82.69679585	97.94542262

Method: Least Squares

Noise variance	1	2	4	8	16	32
Mean (deg)	30.18121555	30.38353449	29.70250551	30.34755105	30.55363762	29.83764725
Std_Dev (deg)	5.046729542	7.188743015	10.09114202	15.24040694	22.42928573	34.96687131

实验结果显示，三种相位估计方法在不同噪声方差下的表现存在显著差异。最大似然法和最小二乘法在整个噪声方差范围内表现稳定，估计均值始终接近真实值 30 度，标准差随着噪声方差的增加而逐渐上升，呈现出合理的线性增长趋势，说明这两种方法对噪声的鲁棒性较强，估计精度随噪声恶化而下降但不失控。而 DFT 方法在低噪声时表现尚可，但随着噪声方差的增大，其估计均值迅速偏离真实值，尤其在噪声方差达到 8 以上时，均值明显偏低，甚至在方差为 32 时降至 2 度左右，几乎失去

了估计能力。同时，其标准差也呈爆炸式增长，远高于其他两种方法，说明 DFT 方法在高噪声环境下极不稳定，估计结果严重偏离真实值，误差大且不可控。综合来看，最大似然法和最小二乘法在本实验设置下具有更好的抗噪性能和估计稳定性，而 DFT 方法虽计算简便，但在噪声较强时不适用于精确相位估计任务。

三种相位估计方法在数学结构和抗噪机制上存在本质差异，这直接导致了它们在实验中的性能表现分化。最大似然法和最小二乘法都属于基于模型拟合的估计方式，它们利用整个信号的结构信息进行参数提取，能有效平均噪声影响，因此在噪声方差增大时，估计标准差虽上升但仍保持稳定，均值也始终围绕真实值波动。

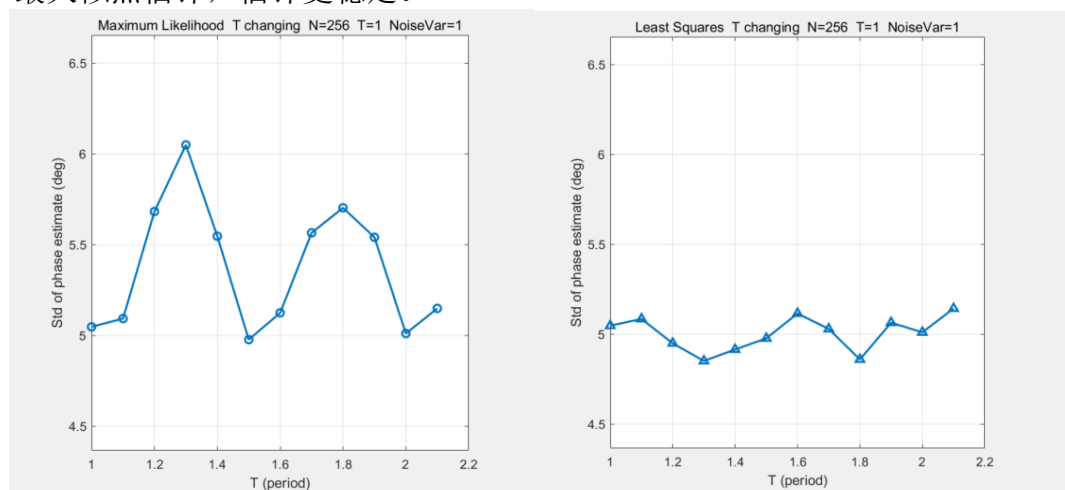
相比之下，DFT 方法的估计机制是从频域中某个特定频率点提取相位：

$$\hat{\phi}_{DFT} = \angle(\sum x_n e^{-j\omega n})$$

这种方式只依赖一个复数频点的相位信息，极易受到噪声干扰。当噪声较强时，频谱泄漏和相位扰动会导致该点的相位严重偏移，尤其在非整数周期采样或频率不匹配时，估计结果会迅速失真。DFT 方法没有利用信号的结构信息，也没有进行任何形式的拟合或平均处理。它本质上是一个“点估计”，而不是“全局估计”。在高噪声下，频谱中的每个点都可能被噪声主导，导致目标频点的相位角几乎随机分布，从而使估计均值偏离、标准差爆炸。这说明 DFT 方法在高噪环境下不具备稳定性，其估计机制缺乏对噪声的抑制能力，是其性能崩溃的根本原因。

## 6. 优化 DFT 算法

前面对于整周期和非正周期对三种求初相的方法已经分析过，DFT 对非整周期极为敏感，存在严重的频率泄漏和谱混叠问题，估计的标准差也极大。再来对比剩下两种方法，最大似然估计和最小二乘法，可以看到最小二乘法得到的标准差也显著小于最大似然估计，估计更稳定。





这主要是因为最小二乘法在时域直接拟合，不依赖频域分析，通过最小化时域残差平方和来拟合参数，对是否为非整周期不敏感，使其 STD 维持稳定。

比较三种方式在不同 T 时的估计效果：

T (period)	1	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2	2.1
Mean (deg) (MLE)	30.181 21555	25.447 00427	23.906 87224	28.115 55362	31.502 67369	30.307 02767	26.885 70075	25.924 111	28.708 28537	30.918 64529	30.281 62994	27.634 88154
Mean (deg) (DFT)	30.181 21555	45.514 73091	61.488 46227	80.474 69584	44.414 66721	-28.87 70528	-41.13 216178	-27.06 006648	-6.682 202824	12.504 49508	30.281 62994	46.697 97113
Mean (deg) (OLS)	30.181 21555	30.249 7285	29.806 02138	29.901 36502	29.998 75831	30.307 02767	30.184 97958	30.009 4478	30.001 18228	29.873 84348	30.281 62994	30.148 10657
Std_Dev (deg) (MLE)	5.0467 29542	5.0923 59122	5.6815 97575	6.0482 60109	5.5457 01937	4.9764 1472	5.1238 26427	5.5643 72068	5.7021 38572	5.5397 68105	5.0099 95166	5.1482 45302
Std_Dev (deg) (DFT)	5.0467 29542	5.2004 08236	5.7247 3771	16.412 05126	88.241 47691	77.032 14333	36.971 27597	11.131 61907	5.6427 54455	5.3693 62288	5.0099 95166	5.1989 92393
Std_Dev (deg) (OLS)	5.0467 29542	5.0847 01192	4.9489 98619	4.8508 32568	4.9147 02522	4.9764 1472	5.1149 5089	5.0287 58546	4.8592 71733	5.0638 28891	5.0099 95166	5.1423 48381

可以看到，当采样数据不为整周期时，最小二乘法估计的效果明显优于其他两种方法，估计均值偏差始终不查过  $0.3^{\circ}$ ，是最优秀方法，最大似然法的估计均值虽然比 DFT 稳定，但也出现了最高大约  $4^{\circ}$  的偏差，而 **DFT** 法的估计均值已经严重偏移，明显不再适用。

于是，我搜索了很多资料看了很多 DFT 优化的相关文献，发现了一种优化算法。首先我们先进行估计误差分析

在实际的测量环境中，必然有噪声存在，所以在噪声背景下的初相估计误差是必须考虑的问题.这里对所提出的方法在 高斯白噪声背景下的初相估计误差进行了分析，并从理论上推导出误差的表达式.

在加性白噪声背景下，接收信号可表示为 $x(t) = s(t) + z(t)$ .其中 $s(t)$ 为式(1)所表示的纯信号， $z(t)$ 为高斯白噪声.在 $0 \sim T$ 仍然以 $\Delta t$ 为间隔，对 $x(t)$ 进行采样，得到长度为 $N$ 的采样序列：

$$x(n) = s(n) + z(n) \quad n = 0,1,2, \dots, N - 1$$

式中， $s(n)$ 即为式(2)； $z(n)$ 为零均值高斯白噪声序列，其方差为 $\sigma_z^2$ ，采样序列的

信噪比为 $\mu = a^2/\sigma_z^2$ .由文献[6]分析可得,在 DFT 最大谱线处,叠加在 DFT 相位上的噪声的方差为

$$\text{var}(\varphi_k) = \frac{N\sigma_z^2}{2A_k^2}$$

$$\text{var}(\varphi_k) = \frac{1}{2N\text{sinc}^2(\delta)\mu}$$

由此可得初相估计的方差为

$$\begin{aligned} \text{var}(\hat{\phi}_0) &= \left(\frac{N_2 - 1}{N_1 - N_2}\right)^2 \cdot \text{var}(\varphi_{k_1}) + \\ &\left(\frac{N_1 - 1}{N_1 - N_2}\right)^2 \cdot \text{var}(\varphi_{k_2}) \end{aligned}$$

式中 $\text{var}(\varphi_{k_1}), \text{var}(\varphi_{k_2})$ 的分析方法与 $\text{var}(\varphi_k)$ 相同,所以有

$$\text{var}(\varphi_{k_1}) = \frac{1}{2N_1\text{sinc}^2(\delta)\mu}$$

$$\text{var}(\varphi_{k_2}) = \frac{1}{2N_2\text{sinc}^2(\delta)\mu}$$

将上面二式代入整理可得

$$\begin{aligned} \text{var}(\hat{\phi}_0) &= \frac{1}{2(N_1 - N_2)^2\text{sinc}^2(\delta)\mu} \cdot \\ &\left[ \frac{(N_2 - 1)^2}{N_1} + \frac{(N_1 - 1)^2}{N_2} \right] \end{aligned}$$

所以有初相估计的均方根误差为

$$\begin{aligned} \sigma_{\phi_0} &= \frac{1}{(N_1 - N_2) \cdot \text{sinc}(\delta)} \cdot \\ &\sqrt{\frac{1}{2\mu} \left[ \frac{(N_2 - 1)^2}{N_1} + \frac{(N_1 - 1)^2}{N_2} \right]} \end{aligned}$$

上式表明初相估计的均方根误差与下面几个因素有关:信噪比,两段序列的采样长度,信号的实际频率与 FFT 频谱最大谱线对应频率的相对偏差 $\delta$ 。提高信噪比可以降低均方根误差。对 $\delta$ 在 $0 \sim 0.5$ 范围内取平均,则初相估计的平均均方根误差可近似表示为

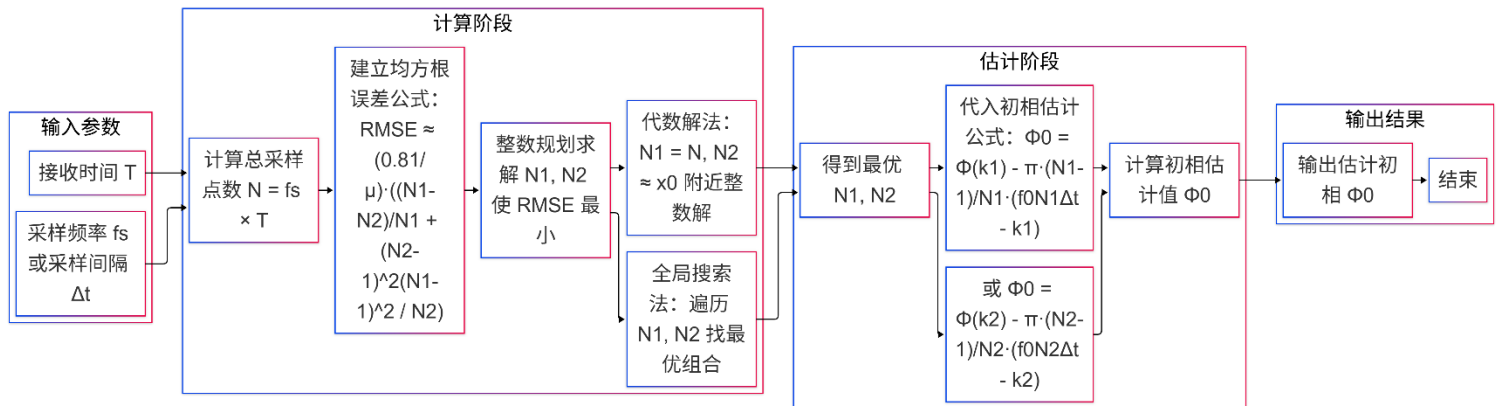
$$\bar{\sigma}_{\phi_0} = \frac{0.81}{(N_1 - N_2)} \sqrt{\frac{1}{\mu} \left[ \frac{(N_2 - 1)^2}{N_1} + \frac{(N_1 - 1)^2}{N_2} \right]}$$

## 改进的初相估计算法

对初相估计方法进行了优化改进.在 $\bar{\sigma}_{\phi_0}$ 中,测量中 $\mu$ 为定值,  $0 < N_2 < N_1 \leq N, N_1, N_2 \in \mathbf{Z}^+$  ( $\mathbf{Z}^+$ 为正整数集合).为使均方根误差最小,则问题转化为以 $N_1, N_2$ 为决策变量,  $\min(\bar{\sigma}_{\phi_0})$ 为目标函数的整数规划问题.在给定 $T, \Delta t$ 的条件下,  $N = T/\Delta t$ , 则可求出使 $\bar{\sigma}_{\phi_0}$ 最小的 $N_1, N_2$ 的值.由代数方法可求出使 $\bar{\sigma}_{\phi_0}$ 最小的 $N_1, N_2$ 的解为:  $N_1 = N$ .设方程 $2x^3 - 2x^2 + 3N(N-1)x - N^2(N-1) = 0$ 的实数解为 $x_0$ , 则 $N_2$ 在 $x_0, [x_0], [x_0] + 1$ 中使 $\bar{\sigma}_{\phi_0}$ 最小的整数处取得.由于上述的整数规划模型中仅有两个决策变量, 所以也可以使用全局搜索法来进行求解.在实际应用中, 为了使初相估计的误差最小, 可以先依据上述方法求出所选用的 $N_1$ 和 $N_2$ 的值, 然后将 $N_1, N_2$ 代入 $\phi_0$ , 进行初相估计.

改进的初相估计算法可以概括为:

- 1) 根据接收时间 $T$ , 采样频率 $f_s$  (或采样间隔 $\Delta t$ ), 计算总采样点数 $N$ 。
- 2) 根据算出的 $N$ 值及式 $\bar{\sigma}_{\phi_0}$ 的推导结果, 在 $\bar{\sigma}_{\phi_0}$ 最小化的原则下, 求出相应的 $N_1$ 和 $N_2$ 的值。
- 3) 将求出的 $N_1, N_2$ 代入 $\phi_0$ , 进行初相估计。



接下来就是根据论文中的改进算法在 Matlab 中实现。

首先检查输入并确定总采样点数为第一段长度，确保输入向量长度与给定  $N$  一致。接着通过解一元三次方程得到一个实数近似解，作为第二段长度的候选中心；对该实数解取上下取整得到候选整数集合，并过滤掉不合法的值（如小于 1 或等于第一段长度的情况）。对每个候选整数计算一个衡量平均均方根误差的指标，选择使该指标最小的整数作为最终的第二段长度。

```
% 求三次方程根并构建候选集合
coeffs = [2, -2, 3*N*(N-1), -N^2*(N-1)];
rts = roots(coeffs);
x0 = real(rts(abs(imag(rts))<1e-10)); x0 = x0(1);
cands = unique([max(1,floor(x0)), min(N-1,ceil(x0))]);
```

```
% 在候选中选择使目标最小的 N2
bestMetric = Inf; bestN2 = cands(1);
for n2 = cands
    num = sqrt(((n2-1)^2)/N + ((N-1)^2)/n2);
    den = N - n2;
    if den <= 0, continue; end
    metric = num / den;
    if metric < bestMetric
        bestMetric = metric; bestN2 = n2;
    end
end
```

确定两段长度后，从输入信号截取前  $N_1$  点与前  $N_2$  点作为两段序列，分别对它们做 FFT 并找到各自幅值最大的谱线索引及对应相位。用两段 DFT 在最大谱线处的相位作为原始相位观测量，为消去频率相关项按论文给出的相位组合关系合成为初相估计（代码里以弧度计算）。相关步骤可用下列代码实现要点：

```
s1 = signal_vec(1:N1);
s2 = signal_vec(1:N2);
S1 = fft(s1); S2 = fft(s2);
[~, idx1] = max(abs(S1)); [~, idx2] = max(abs(S2));
phi1 = angle(S1(idx1)); phi2 = angle(S2(idx2));
k1 = idx1 - 1; k2 = idx2 - 1; % 论文为 0-based 索引
```

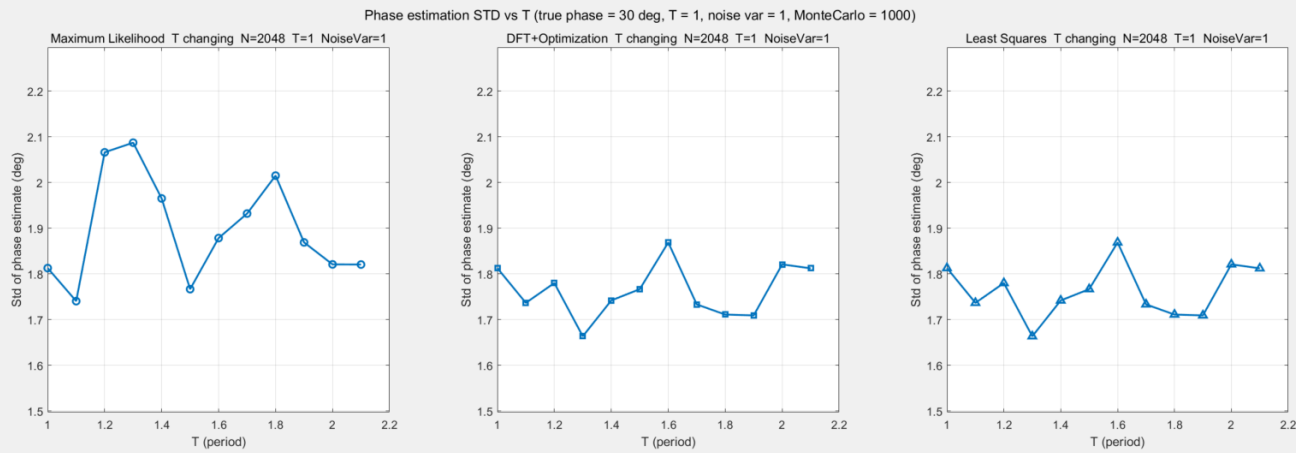
```
% 按论文相位组合关系消去频率项得到初相（简化表示）
```

```
term1 = (phi1 - phi2) / pi;
inside = (N1/(N1 - N2))*(term1 + k1*(1-1/N1) - k2*(1-1/N2)) -
```

```
k1;  
phi0_rad = phi1 - pi*(1-1/N1)*inside;
```

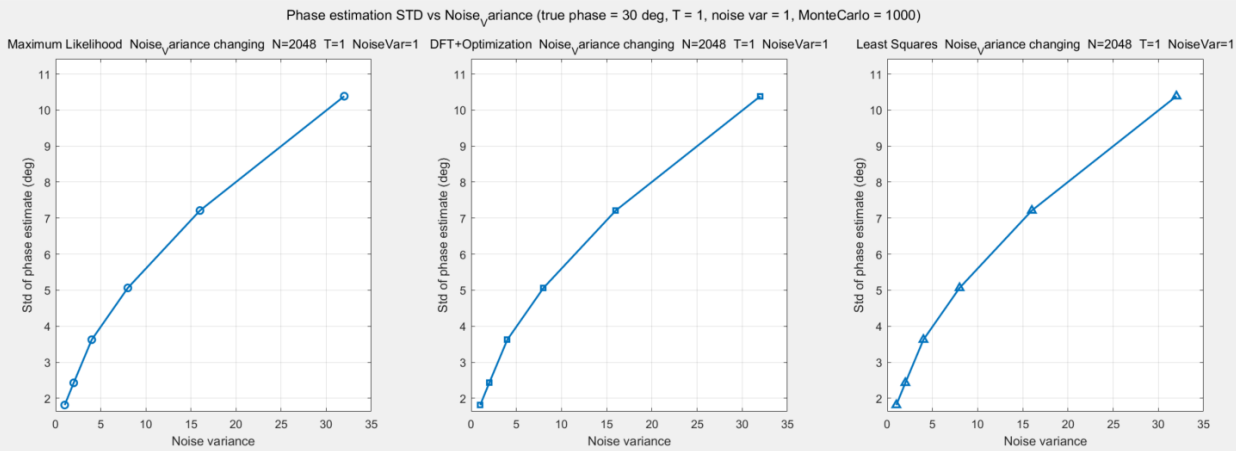
输出把弧度值或度值按需要返回，并保留可选的调试信息（例如选择的 N2、原始相位值、目标函数值等），以便在仿真和验证时对比理论推导结果。

接下来，我们再对不同 T 下做初相估计，这次 DFT 使用改进算法。



T (period)	1	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2	2.1
Mean (deg)	30.083	29.985	30.037	30.013	30.083	30.042	30.029	29.905	29.968	30.052	30.039	29.924
Std_Dev (deg)	1.8125	1.7365	1.7799	1.6634	1.7417	1.7665	1.8686	1.7332	1.7109	1.7091	1.8207	1.8120

不同噪声方差下做初相估计，DFT 改进算法



Noise variance	1	2	4	8	16	32
Mean (deg)	30.08324138	29.99186439	30.09761559	29.88007599	30.29250675	29.88583051
Std_Dev (deg)	1.812580183	2.427903674	3.625877556	5.062539862	7.210811852	10.38374514

之前原始的 DFT 算法的估算 Mean 值甚至出现了负数和 100 多的数值，但现在经过算法改进之后，不管是不同周期（整周期、非整周期）下还是不同噪声方差下，Mean 值几乎就在  $30^\circ$  附近，估值准确度远大于以前，而且标准差也小了很多，已经比最大似然估计要稳定了，与最小二乘法几乎相当。

## 实验收获和体会

刚开始的时候，我以为最大似然估计、DFT 和最小二乘法会有明显的差距，但在固定参数下，比如  $N=256$ 、噪声方差=1 时，三种方法的结果几乎完全一样，均值都在  $30^\circ$  左右，标准差也都在  $5^\circ$  左右，这让我意识到在理想条件下，方法的选择并不是关键，反而是参数设置决定了性能。后来我把  $N$  从 256 增加到 65536，标准差从  $5^\circ$  一路下降到  $0.33^\circ$ ，这个趋势让我很直观地感受到“大样本能抵消噪声”的理论在数据上的体现，比单纯看公式更有说服力。

另一个让我印象深刻的是周期  $T$  的变化。比如固定  $N=256$ 、噪声方差=1 时，整周期（ $T=1$ ）下三种方法的标准差都在 5 度左右，可到了非整周期（比如  $T=1.4$ ），DFT 的标准差直接飙升到 88 度，均值偏离到 44 度，而最小二乘的标准差还稳定在 4.9 度，最大似然也只有 5.5 度，这才直观感受到 DFT 对非整周期的敏感性有多强。还有噪声方差的影响，当方差从 1 增加到 32 时，DFT 的标准差从 5 度涨到 97 度，几乎失去估计能力，而最大似然和最小二乘虽然标准差也上升到 35 度左右，但均值始终围绕 30 度真实值，抗噪性明显更优。另外，当  $N$  从 256 二倍增到 65536 时，三种方法的标准差都从 5 度左右降到  $0.33$  度，这让我切实体会了样本数增加对估计精度的提升。

在写代码的过程中，我也学到不少。以前我习惯把每个情况单独写脚本，结果代码冗余严重，这次我尝试模块化，把三种核心估计算法封装成独立函数，加了 DFT 优

化的开关参数，用户只需调整一个变量就能切换是否启用优化，还能自动判断哪个参数是自变量，不用手动修改代码结构。这种设计不仅让代码更整洁，还大大提高了实验效率，后面测试不同  $N$ 、 $T$ 、噪声方差时，省了很多重复劳动。

解决 DFT 优化问题的过程，让我学会了把理论文献转化为实际代码。一开始看到 DFT 在非整周期和高噪声下的糟糕表现，我查了相关文献，找到了基于均方根误差最小化的优化方法——通过解三次方程确定两段采样长度  $N_1$  和  $N_2$ ，再结合两段 DFT 的相位进行估计。实现后测试发现，优化后的 DFT 在  $T=1.4$  时，标准差降到 1.74 度，均值稳定在 30.08 度，和最小二乘几乎持平；噪声方差=32 时，标准差也只有 10.38 度，远好于原始 DFT 的 97 度。

实验前推导最大似然估计的对数似然函数、DFT 的相位估计公式时，觉得只是抽象的数学运算，但通过代码实现后，看着公式里的  $N$ 、噪声方差  $\sigma$  这些参数变化时，估计标准差真的按理论趋势波动——比如  $N$  增大标准差减小、噪声方差增大标准差上升，这种直观的对应该让我对这些方法的原理理解更透彻了。这次实验也让我明白，做工程实验不能只追求出结果，更要注重过程中的思考。