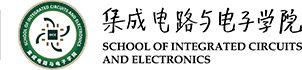
 

**本科生实验报告**

**课程名称： 高级数字信号处理**

**实验名称： 匹配滤波器与脉冲压缩技术实验**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 任课教师： | 田黎育 | | | 实验教师： | | 田黎育 |
| 实验日期： | 2025年 10月 20日 | | | 实验地点： | |  |
| 实验类型： | ☑原理验证 □综合设计 □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： |  | | | | | |



## 实验一

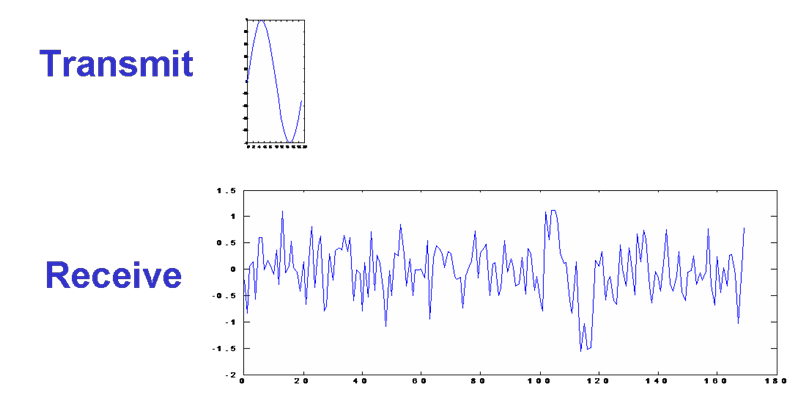
### 实验目的

方波（10点，20、40、自定）信号经过匹配滤波器后的变形

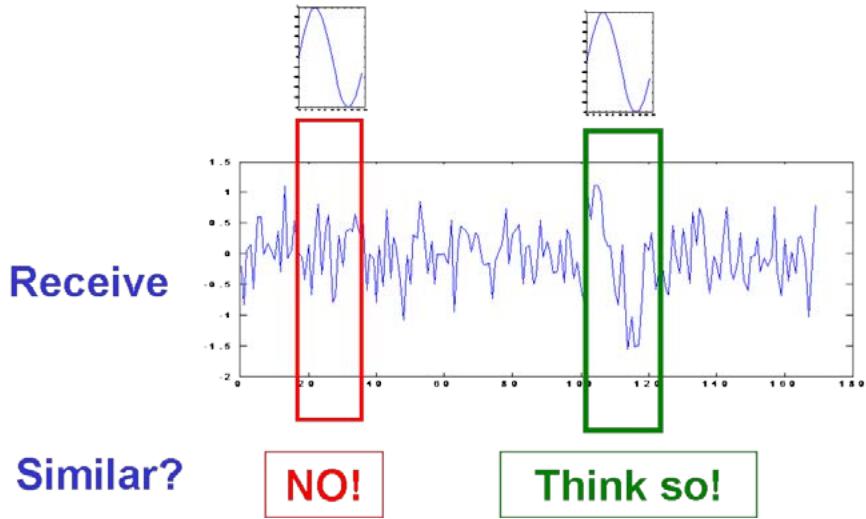
### 实验原理

匹配滤波器是在存在加性随机噪声的已知信号中使信噪比(SNR)最大化的最佳线性滤波器。匹配滤波器通常用于信号检测，将已知信号或模板与未知信号相关联，以检测未知信号中模板的存在。这相当于将未知信号与模板的共轭时间反转版本进行卷积。

雷达的问题是，在实际应用中，接收到的信号和模板都是未知的。许多回波信号被多普勒效应改变得如此之大，以至于它们不再仅仅是发射信号的反射。在这里，回波信号可以叠加在许多不同的多普勒频率上。对于直升机识别来说，旋翼叶片的多普勒频率和机身的多普勒频率(即整个直升机的径向速度)都是重要的。在实际应用中，必须为每一个可能的多普勒频谱建立一个单独的匹配滤波器。

众所周知，我们通过发射脉冲(例如x[n])并检测返回信号来确定目标的距离。发射和接收之间的时间间隔给出了目标距离的信息。下图说明了这个问题。

雷达或声纳系统的目标是检测发射脉冲，在这种情况下是一个周期的正弦波，从接收信号显示在图的底部。在这种情况下，我们可以看到回波在哪里，很明显，接收到的并不是发射脉冲的精确复制品:需要考虑噪声和失真。

进行的方法如下图所示。我们将发送的脉冲与接收信号的每个片段进行比较，并根据一些判断标准，我们决定接收到的和发送的信号之间是否存在任何“相关性”

通过比较所接收的信号与所发射的脉冲来检测所发射的脉冲

**信号模型**

1. **发射信号（模板脉冲）**  
   设发射脉冲为一个长度为 的方波：

* 其中 。

1. **回波信号（无噪声）**  
   假设目标延时为 ，则回波信号为：
2. **接收信号（含噪声）**  
   加入高斯白噪声 ，得到：

* 其中 。

**匹配滤波器原理**

匹配滤波器的核心思想：  
**在高斯白噪声背景下，最大化信噪比 (SNR) 的线性滤波器就是与已知信号的时间反转共轭相匹配的滤波器。**

* 模板信号为 ，则匹配滤波器冲激响应为：
* （这里信号是实数方波，所以共轭就是自己）
* 匹配滤波器输出为：
* 这正是代码里 correl(k) = sum(transmit\_signal .\* receive\_signal(k:(k+M-1))); 的数学形式。

检测与延时估计

1. **相关峰值**  
   当滑动窗口对齐到真实延时 时，输出能量最大：

* 期望值为：
* 方差为：

1. **信噪比提升**  
   匹配滤波器的输出信噪比为：

* 相比输入信噪比：
* 可以看到提升了**倍**，这就是匹配滤波器的核心优势。

### 运行代码及实验内容

clc; clear; close all;

% 复位并固定现代随机数生成器种子

rng('default'); % 将随机数生成器重置为默认配置（避免旧生成器冲突）

rng(1,'twister'); % 固定种子以便结果可复现

N = 500; % 时间轴长度

M = 20; % 脉冲宽度（匹配滤波器长度）

N0 = 148; % 实际目标延时（回波起始索引）

Signal\_Amplitude = 1; % 脉冲幅度

Noise\_Amplitude = 2; % 噪声标准差

SNR = Signal\_Amplitude / Noise\_Amplitude

% 生成回波（echo）和发送脉冲（template）

echo\_signal = zeros(N,1);

transmit\_signal = Signal\_Amplitude \* ones(M,1); % 理想脉冲模板

echo\_signal(N0:(N0+M-1)) = Signal\_Amplitude; % 将脉冲放入回波

% 接收信号 = 回波 + 高斯噪声

receive\_signal = echo\_signal + Noise\_Amplitude .\* randn(N,1);

% 逐点相关（匹配滤波器）计算

correl = zeros(N,1);

for k = 1:(N - M + 1)

correl(k) = sum(transmit\_signal .\* receive\_signal(k:(k+M-1)));

end

% 注意: correl 有效索引为 1:(N-M+1)，其余元素保持为 0

% 找到相关输出的最大值及索引（匹配滤波器检测结果）

[cmax, cidx] = max(correl); % cidx 为相关窗口起点索引（估计延时）

% 找到接收信号的最大值及索引（用于在第二子图上标注）

[rvmax, rvidx] = max(receive\_signal);

% 绘图

figure('Color','w');

subplot(3,1,1);

plot(1:N, echo\_signal, 'b', 'LineWidth', 1.2);

hold on;

% 标注方波（回波）起始点 N0（与其它子图格式一致）

plot(N0, echo\_signal(N0), 'gs', 'MarkerSize',6, 'LineWidth',1.2);

text(N0, echo\_signal(N0), sprintf(' true start=(%d, %.3f)', N0, echo\_signal(N0)), ...

'VerticalAlignment','bottom', 'HorizontalAlignment','left', 'Color','g', 'FontSize',9);

title(['Signal Amplitude = ' num2str(Signal\_Amplitude) ', Noise Amplitude = ' num2str(Noise\_Amplitude) ' 的回波信号']);

xlabel('样点索引');

ylabel('幅度');

xlim([1 N]);

ylim([min(echo\_signal)-0.5 max(echo\_signal)+0.5]);

grid on;

hold off;

subplot(3,1,2);

plot(1:N, receive\_signal, 'k', 'LineWidth', 1);

hold on;

% 标注接收信号的最大值

plot(rvidx, rvmax, 'ro', 'MarkerSize', 6, 'LineWidth', 1.2);

text(rvidx, rvmax, sprintf(' recv max=(%d, %.3f)', rvidx, rvmax), ...

'VerticalAlignment','bottom', 'HorizontalAlignment','left', 'Color','r', 'FontSize',9);

% 可选：也标注真实回波起点 N0

plot(N0, receive\_signal(N0), 'gs', 'MarkerSize',6, 'LineWidth',1.2);

text(N0, receive\_signal(N0), sprintf(' true start=(%d, %.3f)', N0, receive\_signal(N0)), ...

'VerticalAlignment','top', 'HorizontalAlignment','left', 'Color','g', 'FontSize',9);

title(['Signal Amplitude = ' num2str(Signal\_Amplitude) ', Noise Amplitude = ' num2str(Noise\_Amplitude) ' 的接收信号']);

xlabel('样点索引');

ylabel('幅度');

xlim([1 N]);

grid on;

hold off;

subplot(3,1,3);

plot(1:N, correl, 'b', 'LineWidth', 1);

hold on;

% 标注相关输出的最大值（匹配滤波器检测）

plot(cidx, cmax, 'ro', 'MarkerSize', 6, 'LineWidth', 1.2);

text(cidx, cmax, sprintf(' corr max=(%d, %.4f)', cidx, cmax), ...

'VerticalAlignment','bottom', 'HorizontalAlignment','left', 'Color','r', 'FontSize',9);

% 在第三图上也画出真实回波对应的相关起点（N0）

plot(N0, correl(N0), 'gs', 'MarkerSize',6, 'LineWidth',1.2);

text(N0, correl(N0), sprintf(' true start=(%d, %.4f)', N0, correl(N0)), ...

'VerticalAlignment','top', 'HorizontalAlignment','left', 'Color','g', 'FontSize',9);

xlabel('样点索引');

ylabel('相关幅度');

xlim([1 N]);

ylim([min(correl)-abs(min(correl))\*0.05, max(correl)\*1.1]);

title('相关滤波器的输出');

grid on;

hold off;

实验对应关系

| 数学公式 | 代码变量 | 含义 |
| --- | --- | --- |
|  | transmit\_signal | 发射模板脉冲 |
|  | echo\_signal | 延时后的回波 |
|  | receive\_signal | 含噪接收信号 |
|  | correl(k) | 匹配滤波输出（滑动相关） |
|  | cidx | 估计的目标延时 |
|  | cmax | 检测峰值 |

滤波器的冲激响应取为待检测信号的共轭反转并加上时延时，能够使输出的信噪比达到最大。原因在于：滤波器输出实际上是输入信号与冲激响应的卷积，而卷积运算本身包含反转操作，因此会把冲激响应再反转一次，使其与原始信号保持相同的形状。这样一来，卷积结果就等价于信号的自相关。自相关在两个信号完全重叠的时刻T0处达到峰值，此时输出相当于经过相位校正后的能量积累。而对白噪声而言，其自相关理论上接近于零，因此不会产生明显的积累效果。

将待检测信号设为矩形脉冲，相应的，滤波器的冲激响应也是矩形脉冲。二者卷积后得到一个三角波形，其顶点对应于信号出现的时刻估计。在程序实现中，我们以滤波器输出的最大值作为待检测信号到达时刻的判决依据。

Signal\_Amplitude = 1， Noise\_Amplitude = 0时，即只有信号没有噪声时。

可以看到第一幅显示的是理想的回波信号，它在索引N0=248处出现一个宽度为M=20的方波脉冲，幅度为1；第二幅是接收信号，由于噪声幅度设为0，因此它与回波几乎一致，并在脉冲位置标注了最大值和真实起点；第三幅是匹配滤波器（相关器）的输出，能够在正确的延时位置248产生一个明显的峰值，且峰值幅度等于脉冲能量（约为20），并且与真实起点对齐。整体结果说明：在无噪声条件下，匹配滤波器能够准确检测出目标回波的延时位置，并通过相关峰值显著增强信号能量，从而实现最佳检测性能

Signal\_Amplitude = 1， Noise\_Amplitude = 0.5时

第一幅是理想回波信号，在索引248处出现一个宽度为20的方波脉冲，幅度为1；第二幅是接收信号，由于加入了标准差为0.5的高斯噪声，波形不再平滑，脉冲位置附近被噪声扰动，但仍能看出信号的存在；第三幅是匹配滤波器的输出，相关曲线在正确的延时位置248产生了明显的峰值，虽然峰值周围有一定噪声起伏，但主峰依然突出，能够准确指示目标位置。与前一次无噪声的情况相比，这里接收信号的最大值不再与真实脉冲完全一致，而相关输出的峰值也稍有波动，但匹配滤波器仍然有效地增强了信号并保持了检测的可靠性。

Signal\_Amplitude = 1， Noise\_Amplitude = 0.5时，将目标延时N0从248改为148。可以看到匹配滤波器的输出仍能再正确的时延位置出现峰值。

Signal\_Amplitude = 1， Noise\_Amplitude = 2、5时

可以看到，随着噪声的增大，波形被强烈扰动，脉冲位置几乎被淹没，最大值出现在噪声峰而非真实信号点，导致直接观察难以分辨目标；匹配滤波器的输出虽然整体曲线受噪声影响起伏较大，但在正确延时位置148左右仍然形成了一个明显的相关峰，能够指示目标存在，但是输出的预测时延发生了**向后偏移**，且噪声越大偏移越大。与前面噪声较小的情况相比，这里接收信号几乎无法直接识别，而相关输出的主峰相对噪声背景仍然突出。

这是因为噪声在相关运算中也会产生随机峰值，有时这些噪声峰值会与信号主峰叠加或超过信号起点附近的相关值。结果是检测器会把“延时”判在后面某个更大的峰值处，看起来像是预测时延被推迟。

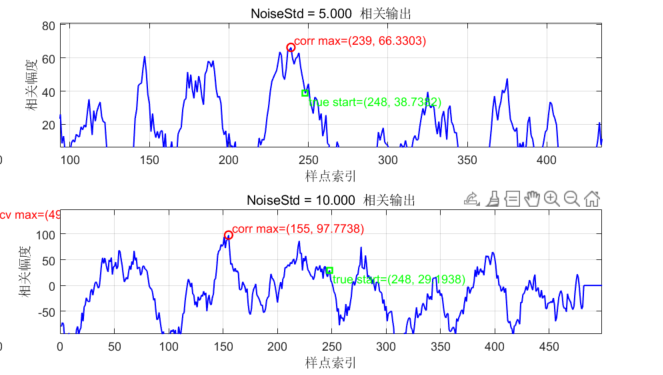
再把Signal\_Amplitude = 1， Noise\_Amplitude = 5改为Signal\_Amplitude = 10， Noise\_Amplitude = 5，噪声幅度不变，输入信号的能量增大。

可以看到匹配滤波器输出再正确延时位置248处再次形成了一个明显的相关峰，又预测成功了，可见我们可以通过增加回波信号的能量，使得目标位置更加明显。

最后，把不同M值（也就是脉冲宽度）的匹配滤波器输出画在一张图里。

随着M从10增大到60，可以看到相关输出的峰值逐渐变得更突出，信噪比提升，目标起点位置在相关曲线上更容易被准确标注出来，这里采用了方波脉冲，相关输出的峰值随脉冲宽度增加而增强，表现出更明显的能量积累效应

再把不同Noise\_Amplitude值（也就是噪声幅值）的匹配滤波器输出画在一张图里。



可以明显地观察到，随着噪声幅度从0.1增大到10，接收信号逐渐被淹没，SNR也随之下降；第三列是匹配滤波输出，低噪声时在理论位置附近会出现明显的相关峰，能够准确检测目标，而噪声越大，相关曲线的波动越强，峰值位置可能偏移甚至被噪声掩盖。直观展示了匹配滤波在不同噪声水平下的性能变化：在高信噪比时能清晰定位目标，但在低信噪比时检测精度和可靠性显著下降。

## 实验二

### 一、 实验目的

仿真方波信号+噪声:输出(最大值的抖动)（蒙特卡洛仿真)最大位置分布图

### 二、 实验原理

1. **相关器输出**  
   对于延迟，相关器输出为：

* 其中：
  + ：发射信号（长度为的矩形脉冲）
  + ：接收信号（含真实时延和噪声）

1. **最大值位置估计**  
   每次蒙特卡洛试验得到的最大值位置为：
2. **直方图统计**  
   Figure6绘制的是所有试验中的分布：

* 其中为指示函数，表示在第次试验中最大值是否出现在位置。
* **理想情况（无噪声）**  
  相关器输出在真实时延处达到唯一最大值，因此：
* 直方图在处为一个尖锐的单峰。
* **有噪声时**  
  噪声会在相关器输出中引入随机波动，可能导致最大值位置偏离。于是：
* 直方图在附近扩展，表现为分布的“抖动”。
* **噪声增大时**  
  当噪声方差增大，信号峰值与噪声波动的对比度下降，最大值位置更容易被噪声主导。  
  结果是：
* 蒙特卡洛仿真的相关器输出的最大值对应的n的次数的直方图由集中单峰逐渐变为分散甚至接近平坦。

总结：**n的出现次数直方图本质上是的经验分布直方图**，它反映了在噪声作用下，相关器最大值位置的统计特性。噪声越大，分布越分散，估计的稳定性越差。

### 三、运行代码及实验内容

clear; close all; clc;

% 本程序使用相关技术估计时延，并画出每次蒙特卡洛试验中相关器输出最大值对应的 n 的分布直方图

randn('state',1);

N = 500; % 数据点数

M = 20; % 发射脉冲信号宽度

MN = 1000; % 蒙特卡洛仿真次数

N0 = 248; % 真实时延（样本索引）

a = 1; % 脉冲信号幅度

sigma = 0.5;

echo\_signal = zeros(N,1);

transmit\_signal = a \* ones(M,1);

echo\_signal(N0:(N0+M-1)) = a;

Delayestimate\_error = zeros(MN,1); % 保存每次试验的误差

ctime\_all = zeros(MN,1); % 保存每次试验相关器最大值对应的 n (位置)

cmax\_all = zeros(MN,1); % 保存每次试验相关器的最大值 (可选，用于分析抖动幅度)

for j = 1:MN

receive\_signal = echo\_signal + sigma .\* randn(N,1);

correl = zeros(N,1);

for k = 1:(N - M)

correl(k) = sum(transmit\_signal .\* receive\_signal(k:(k+M-1)));

end

[cmax, ctime] = max(correl);

ctime\_all(j) = ctime; % 记录最大值对应的索引 n

cmax\_all(j) = cmax; % 记录最大值大小

Delayestimate\_error(j) = ctime - N0;

end

% 增加图窗高度，为标题和统计量腾出空间

figure('Position',[100 100 1200 900]);

% 回波信号

subplot(3,3,1)

plot(echo\_signal,'LineWidth',1.2)

title('回波信号')

xlabel('样本索引 n')

ylabel('振幅')

axis([0 N 0 2\*a])

% 在方波起始位置用小红点标注横纵坐标

hold on

x\_dot1 = N0; % 方波起始索引

y\_dot1 = echo\_signal(N0); % 对应幅值（应为 a）

plot(x\_dot1, y\_dot1, 'ro', 'MarkerSize',4, 'MarkerFaceColor','r')

text(x\_dot1+4, y\_dot1, sprintf('(%d, %.2f)', x\_dot1, y\_dot1), 'Color','r', 'FontSize',10)

hold off

% 接收信号（最后一次蒙特卡洛试验的接收信号）

subplot(3,3,2)

plot(receive\_signal,'LineWidth',1)

title('接收信号（最后一次试验）')

xlabel('样本索引 n')

ylabel('振幅')

% 相关器输出（对应最后一次试验）

subplot(3,3,3)

plot(correl,'LineWidth',1)

title('相关器输出（最后一次试验）')

xlabel('样本索引 n')

ylabel('相关值')

axis([0 N min(correl)-0.1\*abs(min(correl)) max(correl)+0.1\*abs(max(correl))])

% 时延估计误差（随试验次数变化）

subplot(3,3,4)

plot(1:MN,Delayestimate\_error,'-o','MarkerSize',3)

title('时延估计误差（每次试验）')

xlabel('蒙特卡洛试验次数')

ylabel('误差（样本点）')

xlim([1 MN])

% 时延估计误差直方图

subplot(3,3,7)

histogram(Delayestimate\_error,'Normalization','probability')

title('时延估计误差直方图')

xlabel('误差（样本点）')

ylabel('概率')

grid on

% 相关器输出最大值位置 n 的直方图（关键新增子图）

subplot(3,3,[5 6 8 9])

n\_min = 1;

n\_max = N - M;

edges = (n\_min-0.5):(1):(n\_max+0.5);

h = histogram(ctime\_all,'BinEdges',edges,'Normalization','count');

title('每次试验相关器最大值对应的 n 的分布')

xlabel('n（最大值出现的样本索引）')

ylabel('出现次数')

grid on

% 找到直方图最高点并用小红点标注横纵坐标

[counts, binEdges] = histcounts(ctime\_all, edges);

[peakCount, peakIdx] = max(counts);

binCenter = (binEdges(peakIdx) + binEdges(peakIdx+1))/2; % 最高柱的中心

hold on

plot(binCenter, peakCount, 'ro', 'MarkerSize',4, 'MarkerFaceColor','r')

text(binCenter+1, peakCount, sprintf('(%g, %d)', binCenter, peakCount), 'Color','r', 'FontSize',10)

hold off

% 找到最后一个非零值

last\_nonzero = max(ctime\_all);

% 横坐标范围往外扩展 2 个点（可调）

xlim([min(ctime\_all)-2, last\_nonzero+2])

% 相关器最大位置随试验次数变化（用散点显示抖动趋势）

% subplot(3,3,7)

% plot(1:MN, ctime\_all, '.','MarkerSize',4)

% hold on

% yline(N0,'r--','真实时延 N0','LineWidth',1)

% title('相关器最大位置随试验次数变化')

% xlabel('蒙特卡洛试验次数')

% ylabel('最大位置 n')

% ylim([n\_min-1 n\_max+1])

% grid on

% hold off

% 留空用于整体信息或者其它分析

% subplot(3,3,8)

% axis off

% 计算统计量（用于显示）

mean\_err = mean(Delayestimate\_error);

var\_err = var(Delayestimate\_error);

mean\_ctime = mean(ctime\_all);

std\_ctime = std(ctime\_all);

% 总体标题（顶端）

sgtitle('相关法时延估计与最大值位置分布（蒙特卡洛仿真）','FontSize',16,'FontWeight','bold')

% 在总体标题下一行显示统计量

annotation('textbox',[0.10 0.955 0.8 0.03],...

'String',sprintf('样本数 MN = %d 均值误差 = %.4f 样本点 方差误差 = %.4f (样本点^2) 最大位置均值 = %.3f 最大位置标准差 = %.3f',MN,mean\_err,var\_err,mean\_ctime,std\_ctime),...

'EdgeColor','none','HorizontalAlignment','center','FontSize',11,'FontWeight','normal');

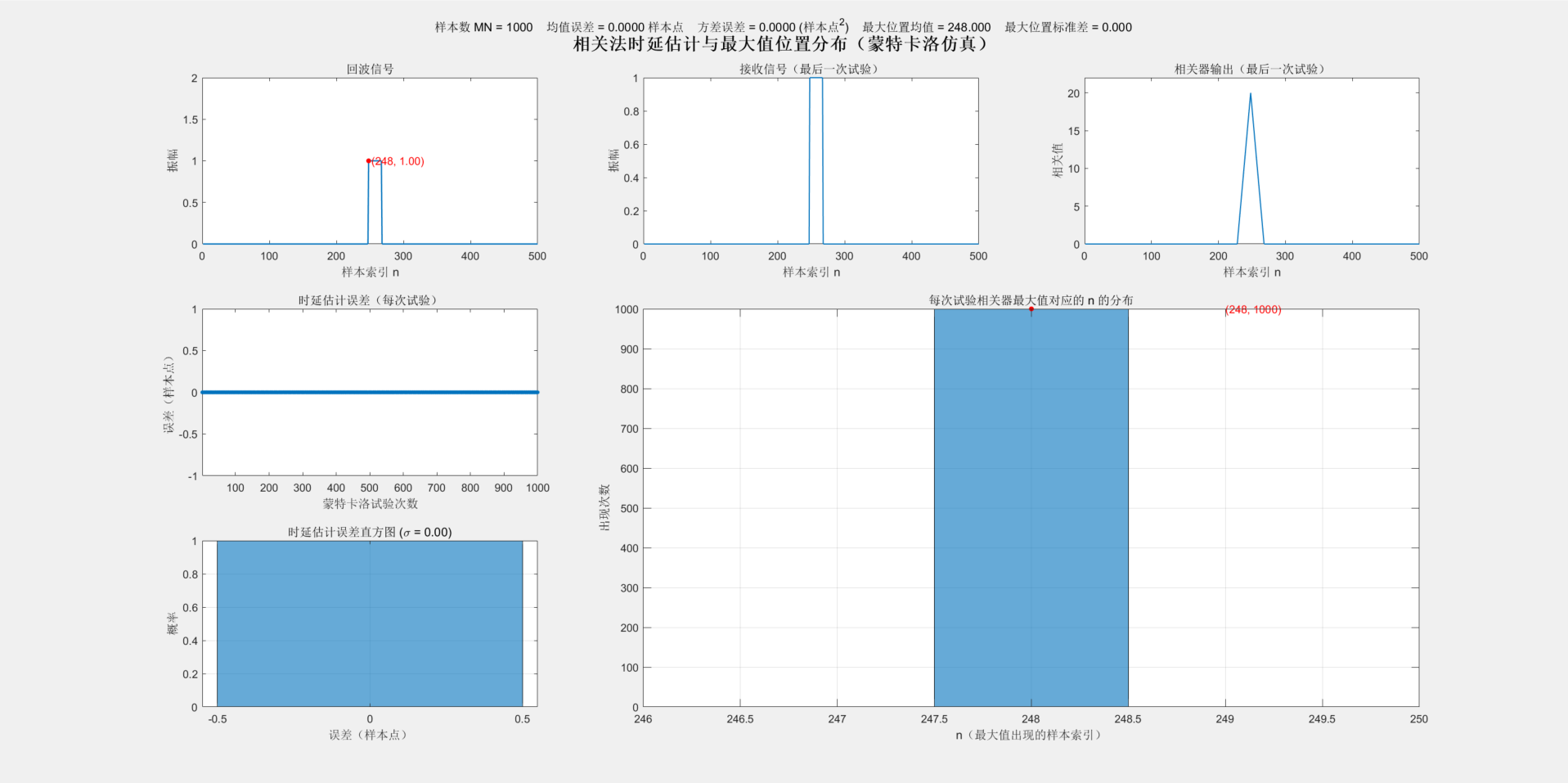
% 将方差结果保存到变量 var1 以兼容原代码

var1 = var\_err;

下面来评估用匹配滤波器进行回波接收时间的检测的性能:

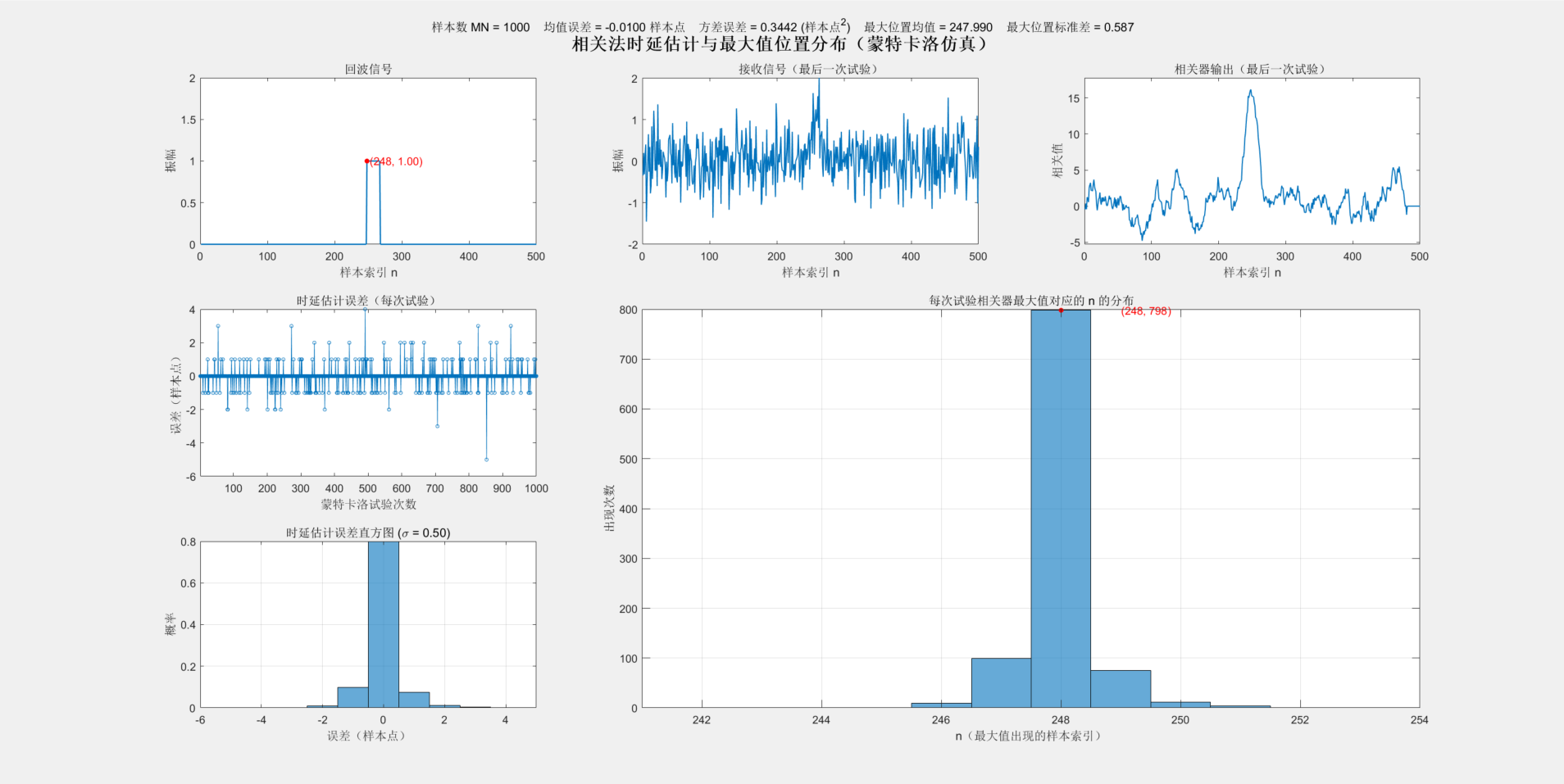
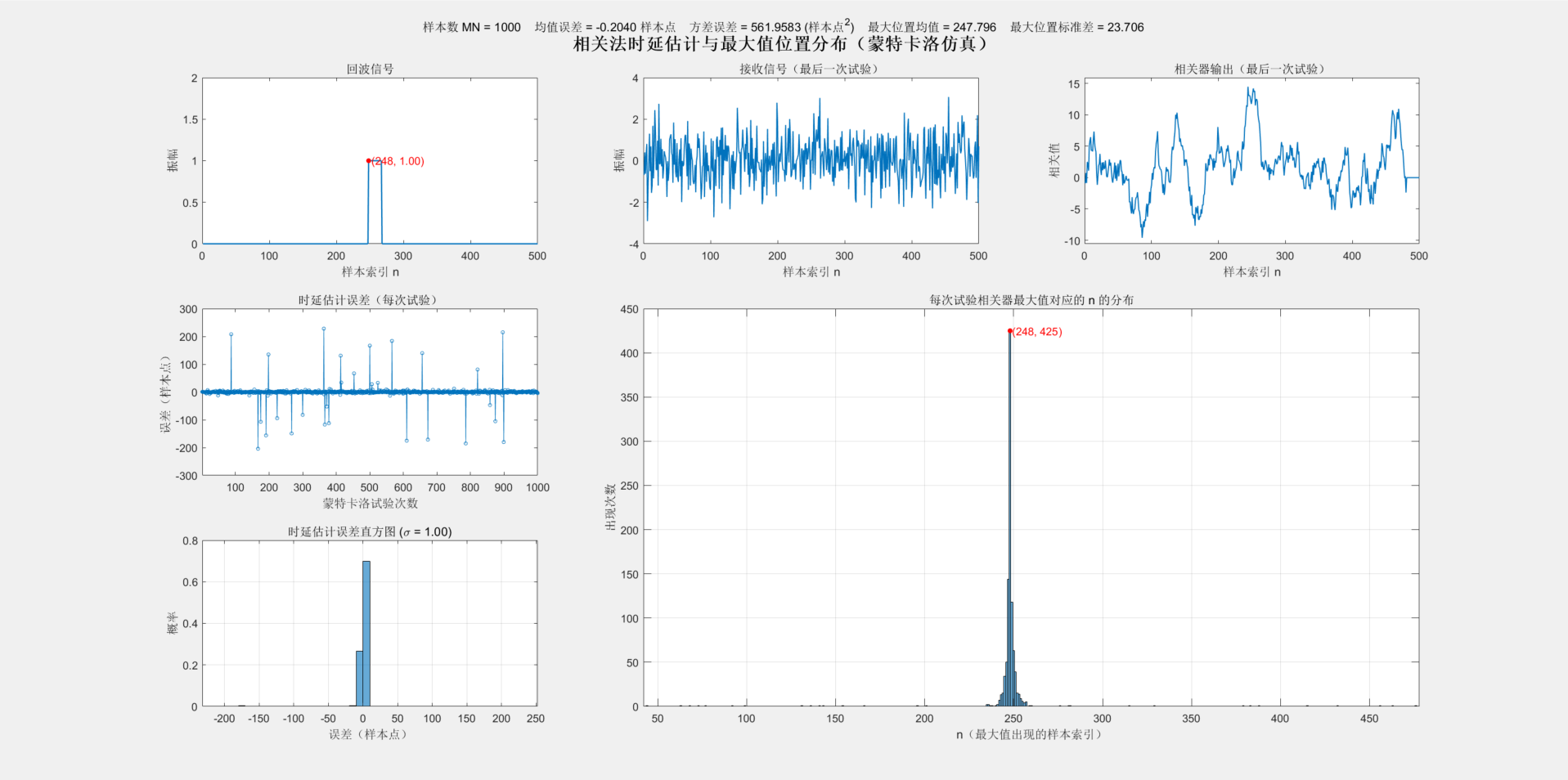
进行蒙特卡洛仿真，先固定次数为1000，每次添加不同的噪声，可以得到估计子的直方图、均值和方差。

时



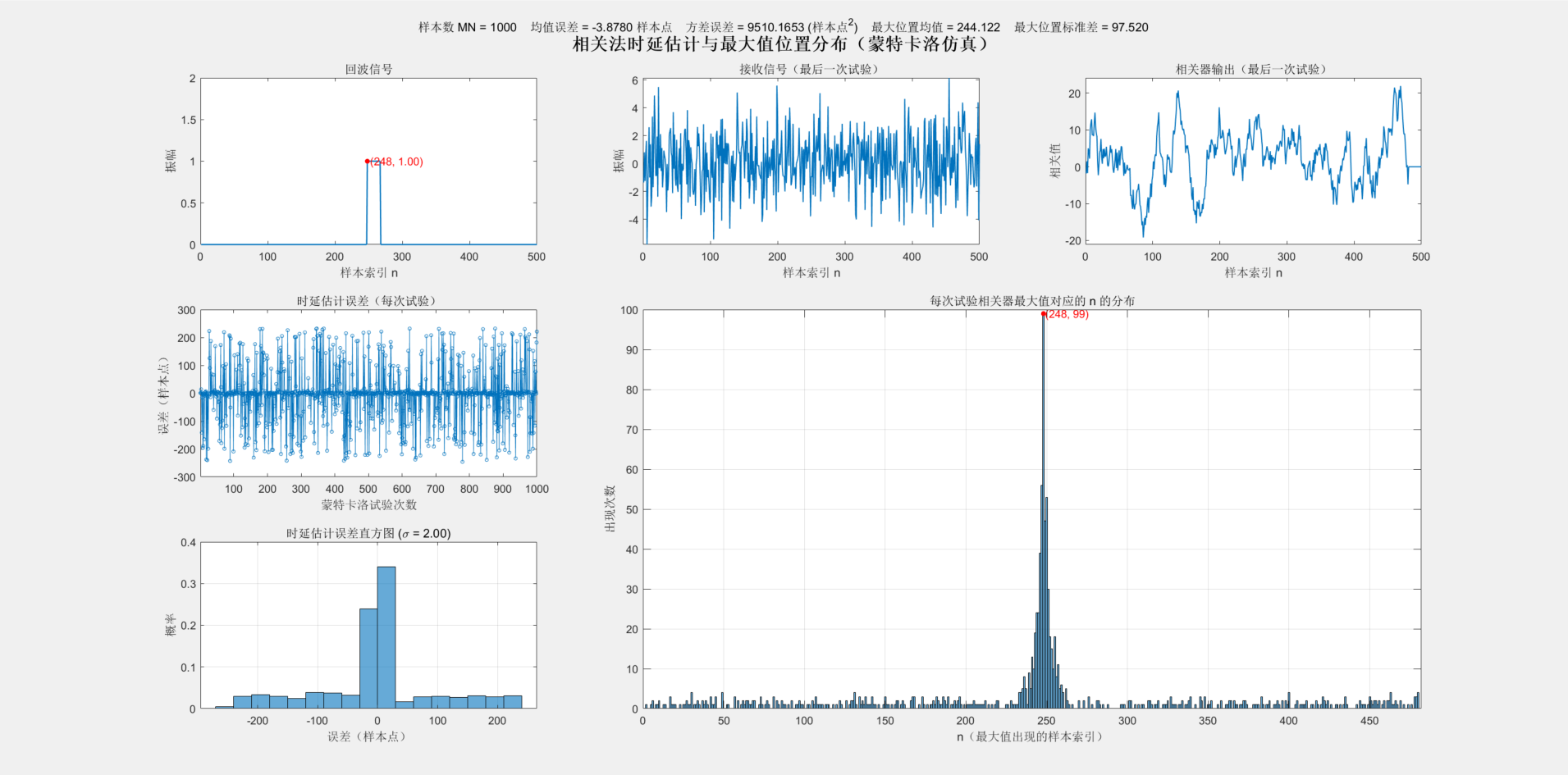
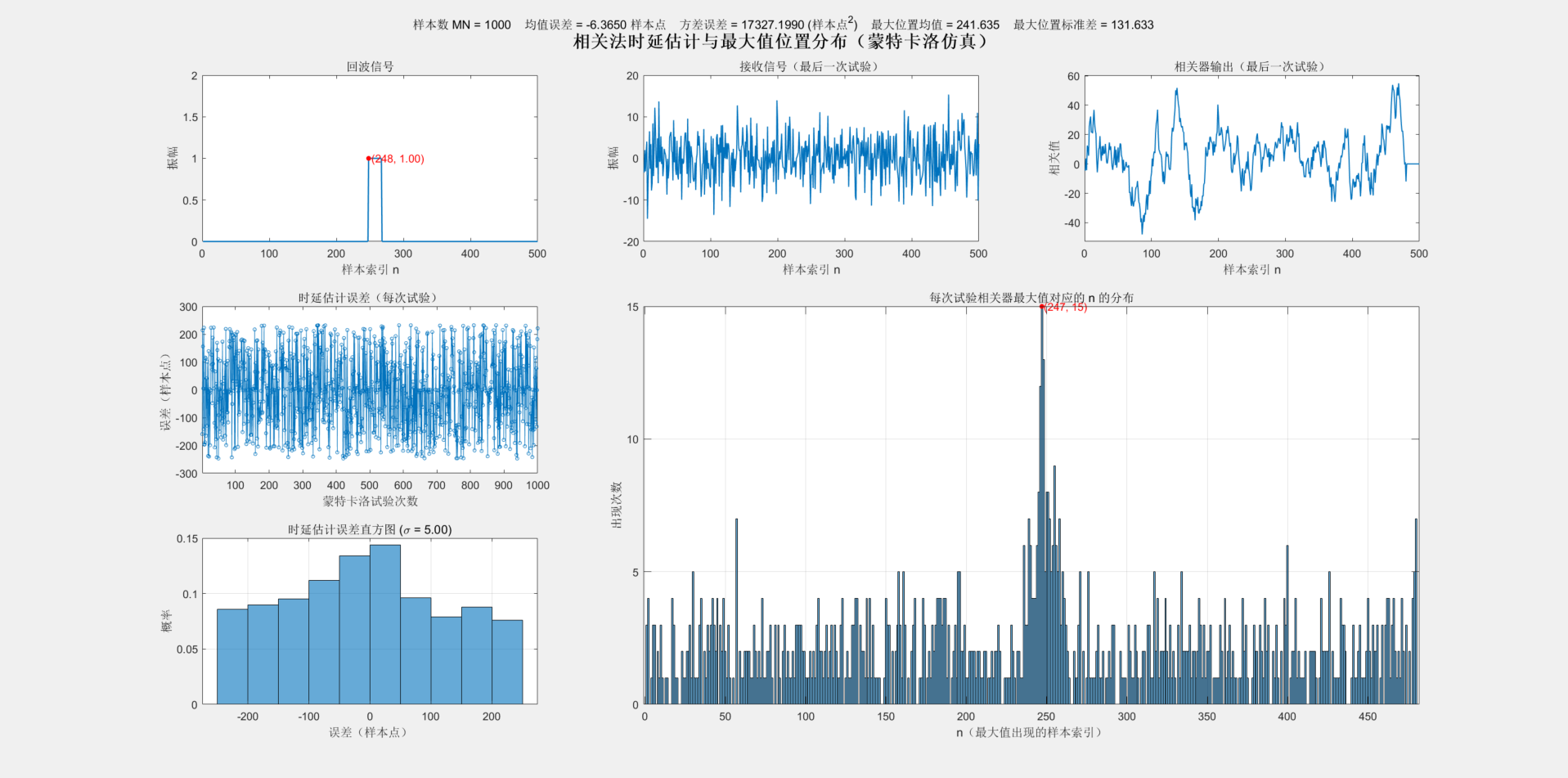
时

时



时

时

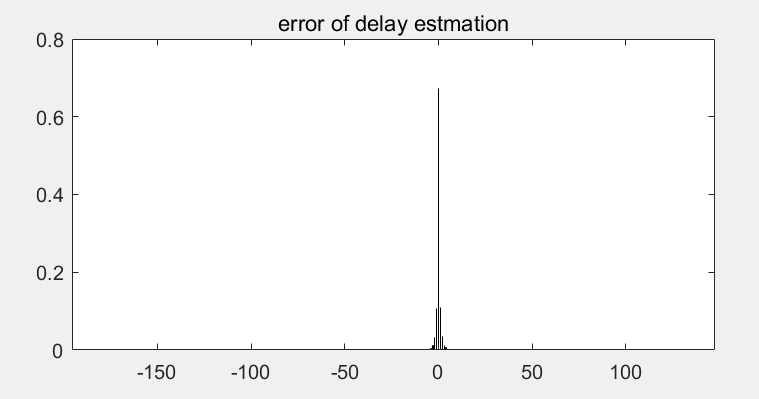
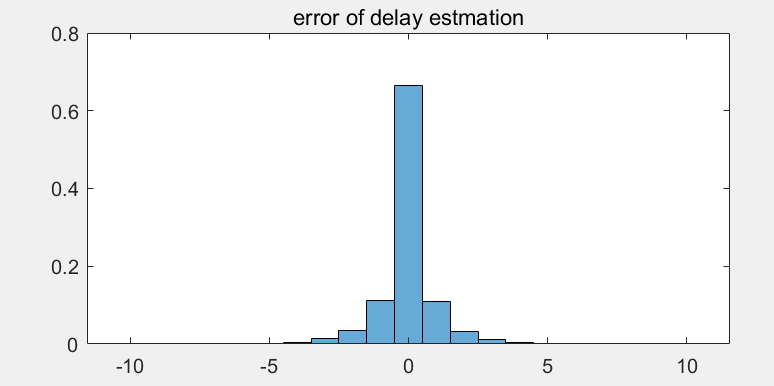


当噪声很小时，**时延估计误差直方图**几乎全都集中在0上，说明算法几乎每次都能准确找到真实时延；这时直方图就是一个很高的“尖柱子”。随着变大，误差开始往两边扩散，出现等偏差的情况，零点的柱子变矮，两边的柱子变高，整体分布越来越宽。换句话说，噪声越大，估计结果越容易偏离真实值。

对应的，**相关器最大值对应的n的分布**在小噪声时几乎全都落在真实时延上，直方图就是一个单一的高峰；但随着增大，峰值高度下降，相邻的247、249等点也开始频繁出现，分布逐渐展宽，说明最大值的位置在真实点附近“抖动”。

从原理上讲，相关器在真实时延处的输出和在其他位置的输出差距，取决于信噪比。信噪比可以近似写成

其中是脉冲幅度，是脉冲宽度。噪声越大，SNR越低，真实峰和噪声峰的差别就越不明显，于是估计结果更容易被噪声干扰，表现为误差直方图变宽、最大值位置分布扩散。当噪声幅度增加时，信号淹没于噪声，检测的性能下降，方差增大。这是由于噪声的缘故，回波信号中可能会随机出现更高的峰值。

不断增加蒙特卡洛仿真次数MN从1000增加到100000.

由模拟结果可以明显看出，随着蒙特卡洛实验次数的不断增加，估计结果中的系统性偏差逐渐减小，并最终趋近于0。换句话说，当重复实验的次数达到足够大的规模时，随机误差在统计意义上会相互抵消，从而使得整体的偏差几乎可以忽略不计。特别是当实验次数超过上十万次这一数量级时，可以合理地预测并认为整体的平均预测偏差将收敛至零，即在大样本条件下，估计结果能够很好地反映理论期望值。

## 实验三

### 一、实验目的

相位编码信号的脉冲压缩:BK、m，以及加噪声的影响

### 二、实验原理

实验生成的长度为 的 BK 码和 MF 码序列需要满足以下几个基本条件：

1. **序列元素的取值范围**  
    所有序列中的元素只能取 或 ，这是构造 BK 码和 MF 码的基本要求。这样的二值化设计保证了序列在卷积和匹配过程中具有良好的数学可操作性和物理可实现性。
2. **卷积特性与滑窗选择**  
   当序列与原始信号进行卷积时，卷积核的形状和长度会直接影响结果：
   * 若滑动窗口的形状为 ，卷积结果与原始信号完全一致，相当于没有进行任何滤波或扩展。
   * 若滑动窗口的形状为 ，则在每个时间点上，卷积结果会表现为 或 的累积效应。只有当序列与原信号完全对齐时，卷积结果才会达到峰值 ；而在其他非对齐时刻，卷积结果则会显著减弱，仅为 或 左右。由此可见，BK 码和 MF 码的设计使得卷积输出在时间上具有高度集中性。
3. **回波信号与匹配滤波效果**  
    将满足上述条件的 BK 码或 MF 码序列与原始信号相加，可以得到一个包含目标信息的回波信号。随后，将该回波信号进行反转并与 MF 码序列进行卷积，就能够得到匹配滤波的结果。在这一过程中，噪声成分由于相互抵消而被有效削弱，而目标信号则通过叠加得到显著增强。最终的效果是：在目标所在的位置，卷积输出会出现明显的峰值，而在其他位置，噪声响应范围极窄且幅度很小，从而实现了对目标的突出和对噪声的抑制。

### 三、运行代码及实验过程

clc; clear; close all;

BP\_Type = 1; % 1-BK, 2-m

BP\_Length = 511; % 63/127/255/511/1023/2047

noieseAmp = 0; %噪声幅度

bp\_pc\_result = BP\_Pulse\_Comp(BP\_Type, BP\_Length, noieseAmp);

%脉冲压缩函数

function bp\_pc\_result = BP\_Pulse\_Comp(BP\_Type, BP\_Length, noieseAmp)

if BP\_Type == 1

bp\_echo = [1,1,1,1,1,-1,-1,1,1,-1,1,-1,1]; %生成 BK 码

else

bp\_echo = produce\_m(BP\_Length); %生成 m 序列

end

sig\_length = length(bp\_echo);

noise1 = noieseAmp \* randn(1, sig\_length); %生成噪声信号

bp\_echo\_noise = bp\_echo + noise1; %回波信号

subplot(221);

plot(bp\_echo);

if BP\_Type == 1

title('BK 码回波信号');

else

title('m 序列回波信号');

end

axis([1 length(bp\_echo) -2 2]);

subplot(222);

plot(bp\_echo\_noise);

%chirp\_MF: chirp 脉压的匹配滤波器, 信号的(共轭)反转

bp\_MF = bp\_echo(length(bp\_echo) : -1 : 1);%将回波信号反转

title('加噪声的回波信号');

axis([1 length(bp\_echo\_noise) min(bp\_echo\_noise)-1 max(bp\_echo\_noise)+1]);

%二相码脉冲压缩

bp\_pc\_result = conv(bp\_echo\_noise, bp\_MF);

[cm,ct]=max(bp\_pc\_result);

subplot(223);

plot(bp\_pc\_result);

text(ct,cm,['(',num2str(ct),',',num2str(cm),')']);

title('二相码脉压的结果');

xlim([1,length(bp\_pc\_result)]);

bp\_pc\_result = abs(bp\_pc\_result) + 0.1; %加 0.1 防止对 0 求对数

subplot(224);

plot(20\*log10(bp\_pc\_result/max(bp\_pc\_result)));

title('二相码脉压的结果(dB 表示)');

ylabel('dB');

xlim([1,length(bp\_pc\_result)]);

end

%产生 m 序列的函数

function ms = produce\_m(ms\_Length)

if ms\_Length == 63

x=[0,0,1,0,1,0]; %basic code, 可以设置为任何不为全 0 的 0-1 序列

y=[5,6]; %feedback link

elseif ms\_Length == 127

x=[0,0,1,0,1,0,0];

y=[6,7];

elseif ms\_Length == 255

x=[0,0,1,0,1,0,0,0];

y=[4,5,6,8];

elseif ms\_Length == 511

x=[0,0,1,0,1,0,0,0,1];

y=[5,9];

elseif ms\_Length == 1023

x=[0,0,1,0,1,0,0,0,1,0];

y=[7,10];

elseif ms\_Length == 2047

x=[0,0,1,0,1,0,0,0,1,0,1];

y=[9,11];

end

k=length(x);

ms=x;

if ms\_Length == 255

for i=k+1:2^k-1

tmp1 = xor( ms(i-y(1)), ms(i-y(2)) );

tmp2 = xor( ms(i-y(3)), tmp1 );

tmp3 = xor( ms(i-y(4)), tmp2 );

ms(i) = tmp3;

end

else

for i=k+1:2^k-1

ms(i)=xor(ms(i-y(1)),ms(i-y(2)));

end

end %生成非0即1序列

ms=ms\*-2+1; %将非 0 即 1 序列转化为非-1 即 1 序列

end

下图是长度为13的BK码的仿真示意图，分别绘制了BK码的时域波形和它的自相关，即噪声为0时匹配滤波器的输出

接着将噪声幅度提高至1，其他参数不变

接着将噪声幅度提高至5，其他参数不变

可以看到，随着噪声幅度的逐渐增大，脉冲压缩结果的主峰逐渐被噪声淹没，表现为主峰与旁瓣的对比度下降，信噪比减小。当噪声幅度较低时，卷积结果仍能清晰地突出主峰位置，脉冲压缩的聚焦效果明显；但当噪声幅度接近或超过信号幅度时，噪声在卷积过程中被放大，其随机峰值可能超过真实主峰，从而导致峰值位置偏移或出现虚假峰。换句话说，信号长度固定时，噪声能量的累积会破坏原有的自相关特性，使得脉冲压缩结果不再稳定，最终表现为检测性能下降甚至失效。

理论上，对于长度为 (N=13) 的 BK 码，无噪声时脉冲压缩（匹配滤波，即信号与自身**共轭反转序列**做卷积）的峰值横坐标应为 13。这是因为匹配滤波的本质是寻找信号与模板（这里是自身反转后的序列）的最佳匹配位置，当输入信号与模板完全对齐时，卷积结果会在长度 N 对应的位置出现峰值（两个长度为 L 的序列卷积，理想峰值位于 L 处）。

从数据看：

* 噪声幅度为 0 时，左下角子图峰值横坐标接近 13，与理论一致；
* 噪声幅度为 1 时，峰值横坐标虽有偏差，但仍接近 13；
* 噪声幅度为 5 时，峰值横坐标明显偏离 13。

这是因为噪声是**随机信号**，会对卷积结果（脉冲压缩输出）引入随机干扰。当噪声幅度小时，信号的 “主峰能量” 仍能压制噪声的随机起伏，峰值位置接近理论值；当噪声幅度增大（如到 5），噪声的随机能量起伏会显著干扰甚至掩盖原本的理想峰值，导致检测到的峰值位置偏离理论值 13，所以噪声越大，峰值位置越不准确。

然后是长度为255的m序列的仿真示意图，分别绘制了m序列的时域波形和它的自相关，即噪声为0时匹配滤波器的输出

接着将噪声幅度提高至1，其他参数不变

接着将噪声幅度提高至5，其他参数不变

可以看到，噪声较小时，脉冲压缩的最大值可以显示序列长度255，但随着噪声的增大，脉冲压缩的最大值之间偏移，噪声幅度为10时，已经偏移到了281处。

于是思考，除了SNR，信号幅度和噪声幅度，序列长度有没有可能减小噪声的影响呢？

那么其他参数不变，保持噪声幅度为10，逐渐增加序列长度为511，1023。



可以看到，当信号长度增加为511和1023时，可以精确地找到目标位置。

m序列（最大长度序列）是由线性反馈移位寄存器（LFSR）产生的伪随机二进制序列，具有接近理想噪声的统计特性。其自相关函数满足：

其中是序列长度。

**含义：**在零延迟时，自相关峰值为，非常尖锐。在非零延迟时，相关值恒为，即旁瓣极低且均匀。

这意味着m序列在匹配滤波后会产生一个**极高的主峰**和**几乎平坦的低旁瓣**，非常适合目标检测。

当序列长度增大时：**主峰高度**峰值幅度，即主峰能量随长度线性增加。**旁瓣能量**仍然保持在，与无关。**信噪比提升**：脉冲压缩增益为。

例如：，压缩增益约。，压缩增益约。

因此，随着增大，主峰与旁瓣的对比度越来越大，噪声被“平均”掉，目标位置（即相关峰位置）就能被更精确地识别。

脉冲压缩是通过**匹配滤波器**实现的：

其中：是发射的m序列，是噪声，是匹配滤波器（即m序列的反转）。

结果就是**信号与自身的自相关+信号与噪声的互相关**。  
由于噪声是零均值高斯分布，互相关项在统计意义上趋近于0；而自相关项在零延迟处产生一个尖锐的峰值。随着增大，峰值能量远大于噪声项，因此目标位置（峰值坐标）更加清晰。

因此，**短序列**：主峰不够高，噪声和旁瓣容易掩盖目标。**长序列（511,1023）**：主峰能量大幅提升，旁瓣仍然保持低平，噪声被平均掉→峰值位置几乎不会错判。

## 实验四

### 一、实验目的

chirp信号的脉冲压缩

### 二、实验原理

在本实验中选用线性调频（chirp）信号作为发射脉冲，其基本原理是通过在有限脉冲宽度内让瞬时频率随时间线性变化，从而在时域上保持较长脉冲能量积累，同时在匹配滤波后获得窄脉冲的高分辨特性。设带宽为，起始频率为，调频斜率为，则发射信号可表示为

该信号的瞬时频率为，在时间内线性扫频。接收端在噪声干扰下得到回波，通过与发射信号的共轭时间反转序列进行卷积，即匹配滤波运算：

其中表示卷积。匹配滤波的作用是最大化信噪比并实现脉冲压缩，输出结果在目标延迟位置形成尖锐峰值，其主瓣宽度近似为，表明距离分辨率与带宽成反比，而旁瓣电平由信号窗函数决定。实验中通过对比原始chirp、加噪信号及匹配滤波输出，可以直观验证“长脉冲能量积累、短脉冲分辨率”的脉冲压缩原理，并观察信噪比提升和峰值检测效果。

匹配滤波器的输出峰值位置应当出现在回波信号与匹配滤波器完全对齐的时刻。由于实验中采用的是“发射信号+噪声”直接进行自匹配卷积，卷积结果的主峰会出现在索引位置等于信号长度的点，即横坐标大约为（其中是发射脉冲的采样点数）。如果横轴用时间刻度表示，则峰值对应的时间大约是脉冲宽度。这是因为匹配滤波器是将发射信号做共轭时间反转后与接收信号逐点滑动相乘并累加，只有当两者完全重叠时，相关性最大，输出能量集中形成尖锐主峰。其物理意义在于：峰值所在的横坐标对应目标回波到达的延迟时间，进而可换算为目标距离；而峰值幅度则反映了信号能量与噪声比的提升效果。因此，理论上该子图的最高点横坐标就是信号持续时间对应的采样点或时间位置，它标志着目标的真实回波延迟位置。

### 三、运行代码及实验过程

clc; clear; close all;

% === 选择信号类型 ===

% BP\_Type = 1 -> BK 二相码（示例）

% BP\_Type = 2 -> m 序列（需指定 BP\_Length）

% BP\_Type = 3 -> 线性 chirp

BP\_Type = 3; % 1,2,3

BP\_Length = 255; % 用于 m 序列（63/127/255/511/1023/2047）

noieseAmp = 0.5; % 噪声标准差（对 chirp 可按波形幅度调整）

% 对于 chirp，可修改下面参数（仅在 BP\_Type==3 时生效）

chirp\_fs = 10e3; % 采样率 Hz

chirp\_Tp = 0.01; % 脉冲宽度 s

chirp\_B = 2e3; % 带宽 Hz

bp\_pc\_result = BP\_Pulse\_Comp(BP\_Type, BP\_Length, noieseAmp, chirp\_fs, chirp\_Tp, chirp\_B);

% ======================

function bp\_pc\_result = BP\_Pulse\_Comp(BP\_Type, BP\_Length, noieseAmp, chirp\_fs, chirp\_Tp, chirp\_B)

% 支持三种信号：二相码(1)、m序列(2)、线性chirp(3)

% 返回 bp\_pc\_result 为脉压后结果（复或实），并绘图

% 生成信号

if BP\_Type == 1

bp\_echo = [1,1,1,1,1,-1,-1,1,1,-1,1,-1,1]; % BK 码示例

t\_vec = 0:length(bp\_echo)-1; fs = 1;

seqName = sprintf('BK 序列（Length=%d）', length(bp\_echo));

elseif BP\_Type == 2

bp\_echo = produce\_m(BP\_Length); % 生成 m 序列

t\_vec = 0:length(bp\_echo)-1; fs = 1;

seqName = sprintf('m 序列（Length=%d）', length(bp\_echo));

elseif BP\_Type == 3

if nargin < 6

error('调用 chirp 模式需传入 chirp\_fs, chirp\_Tp, chirp\_B 三个参数。');

end

fs = chirp\_fs;

Tp = chirp\_Tp;

B = chirp\_B;

N = round(Tp\*fs);

t = (0:N-1)/fs;

f0 = -B/2;

k = B / Tp;

% 生成基带实值 chirp，若需复包络可用 exp(1j\*...)

bp\_echo = cos(2\*pi\*(f0.\*t + 0.5\*k.\*t.^2));

t\_vec = t;

seqName = sprintf('线性 chirp（Tp=%.4g s B=%.3g Hz fs=%.0f Hz）', Tp, B, fs);

else

error('不支持的 BP\_Type，取值 1,2,3。');

end

sig\_length = length(bp\_echo);

sigAmp = max(abs(bp\_echo));

% 叠加噪声

noise1 = noieseAmp \* randn(1, sig\_length);

bp\_echo\_noise = bp\_echo + noise1;

% 建立匹配滤波器（共轭反转）

bp\_MF = conj(bp\_echo(end:-1:1));

% 脉冲压缩（卷积）

bp\_pc\_result = conv(bp\_echo\_noise, bp\_MF);

% 找峰值（按幅值）

[cm, ct] = max(abs(bp\_pc\_result));

% 绘图

figure('Position',[100 100 1000 700]);

% 子图 1：原始信号

subplot(221);

plot(t\_vec, bp\_echo, 'b-','LineWidth',1);

if BP\_Type == 1

title('BK 码回波信号');

elseif BP\_Type == 2

title('m 序列回波信号');

else

title('chirp 脉冲信号 s(t)');

end

xlabel('样本 / 时间 (s)');

ylim([min(bp\_echo)-0.2\*sigAmp, max(bp\_echo)+0.2\*sigAmp]);

grid on;

% 子图 2：加噪信号

subplot(222);

plot(t\_vec, bp\_echo\_noise, 'k-');

title('加噪声的回波信号');

xlabel('样本 / 时间 (s)');

ylim([min(bp\_echo\_noise)-0.2\*sigAmp, max(bp\_echo\_noise)+0.2\*sigAmp]);

grid on;

% 子图 3：匹配滤波响应并标注峰值（修正：统一横轴并绘制 abs）

subplot(223);

% 统一横轴：若存在采样频率 fs>1，则用时间轴；否则用索引轴

if exist('fs','var') && fs>1

x\_axis = (0:length(bp\_pc\_result)-1)/fs;

x\_peak = x\_axis(ct);

plot(x\_axis, abs(bp\_pc\_result), 'b-','LineWidth',1);

hold on;

plot(x\_peak, abs(bp\_pc\_result(ct)), 'ro','MarkerFaceColor','r');

text(x\_peak, abs(bp\_pc\_result(ct)), sprintf(' (loc=%d, amp=%.3g)', ct, abs(bp\_pc\_result(ct))));

xlabel('时间 (s)');

xlim([min(x\_axis) max(x\_axis)]);

else

x\_axis = 1:length(bp\_pc\_result);

x\_peak = ct;

plot(x\_axis, abs(bp\_pc\_result), 'b-','LineWidth',1);

hold on;

plot(x\_peak, abs(bp\_pc\_result(ct)), 'ro','MarkerFaceColor','r');

text(x\_peak, abs(bp\_pc\_result(ct)), sprintf(' (loc=%d, amp=%.3g)', ct, abs(bp\_pc\_result(ct))));

xlabel('样本索引');

xlim([1 length(bp\_pc\_result)]);

end

hold off;

title('匹配滤波器响应 峰值检测');

grid on;

% 子图 4：dB 标度

subplot(224);

bp\_pc\_display = abs(bp\_pc\_result) + 1e-12;

bp\_pc\_db = 20\*log10(bp\_pc\_display / max(bp\_pc\_display));

if exist('fs','var') && fs>1

plot((0:length(bp\_pc\_db)-1)/fs, bp\_pc\_db, 'm-');

xlabel('时间 (s)');

xlim([min(x\_axis) max(x\_axis)]);

else

plot(bp\_pc\_db, 'm-');

xlabel('样本索引');

xlim([1 length(bp\_pc\_db)]);

end

ylim([max(bp\_pc\_db)-80, 5]);

ylabel('dB');

title('dB');

grid on;

% 总标题

sgtitleStr = sprintf('脉冲压缩 %s — 信号幅度: %.3g 噪声幅度: %.3g', seqName, sigAmp, noieseAmp);

if exist('sgtitle','builtin') || exist('sgtitle','file')

sgtitle(sgtitleStr);

elseif exist('suptitle','file')

suptitle(sgtitleStr);

else

subplot(221);

text(0.5, 1.15, sgtitleStr, 'Units', 'normalized', 'HorizontalAlignment', 'center', 'FontWeight', 'bold');

end

% 在命令窗口显示峰值信息

fprintf('峰值索引 (ct) = %d, 峰值幅值 (cm) = %.6g\n', ct, cm);

end

% === 生成 m 序列函数（保留原实现） ===

function ms = produce\_m(ms\_Length)

if ms\_Length == 63

x=[0,0,1,0,1,0]; y=[5,6];

elseif ms\_Length == 127

x=[0,0,1,0,1,0,0]; y=[6,7];

elseif ms\_Length == 255

x=[0,0,1,0,1,0,0,0]; y=[4,5,6,8];

elseif ms\_Length == 511

x=[0,0,1,0,1,0,0,0,1]; y=[5,9];

elseif ms\_Length == 1023

x=[0,0,1,0,1,0,0,0,1,0]; y=[7,10];

elseif ms\_Length == 2047

x=[0,0,1,0,1,0,0,0,1,0,1]; y=[9,11];

else

error('不支持的 ms\_Length，请使用 63/127/255/511/1023/2047');

end

k=length(x); ms=x;

if ms\_Length == 255

for i=k+1:2^k-1

tmp1 = xor( ms(i-y(1)), ms(i-y(2)) );

tmp2 = xor( ms(i-y(3)), tmp1 );

tmp3 = xor( ms(i-y(4)), tmp2 );

ms(i) = tmp3;

end

else

for i=k+1:2^k-1

ms(i)=xor(ms(i-y(1)),ms(i-y(2)));

end

end

ms = ms \* -2 + 1; % 将 0/1 -> -1/1

end

将噪声幅度设为0，chirp\_fs = 1e5，chirp\_Tp = 0.02，chirp\_B = 2e3;

可以看到，峰值对应的时间大约是脉冲宽度，也就是0.02s，它是信号持续时间对应的采样点或时间位置，标志着回波到达的延迟时间。

将噪声幅度增加为10

可以看到匹配滤波器的抖动增大，方差增加，但仍然能轻松标注出回波到达的延迟时间0.2s

继续增大噪声幅度到100

可以看到，此时噪声已经大到影响了匹配滤波器对回波到达的延迟时间的判断，这里已经不是0.2s了，而是有略微偏差0.018s。

继续将噪声幅度增加到200

此时偏差已经达到了0.04s，严重影响了判断。

理论上，bp\_pc\_result=conv(bp\_echo\_noise,bp\_MF)实际上就是**回波信号与匹配滤波器的卷积**。如果没有噪声，这个结果就是**chirp信号与自身的互相关**，也就是**自相关函数**。

1.自相关函数的峰值位置

* 对长度为的离散信号，其自相关函数在**零延迟（lag=0）**时达到最大值。
* 由于MATLAB的conv(...,'full')输出长度为，零延迟对应的位置是**第个采样点**（1-based索引）。

2.转换到时间坐标

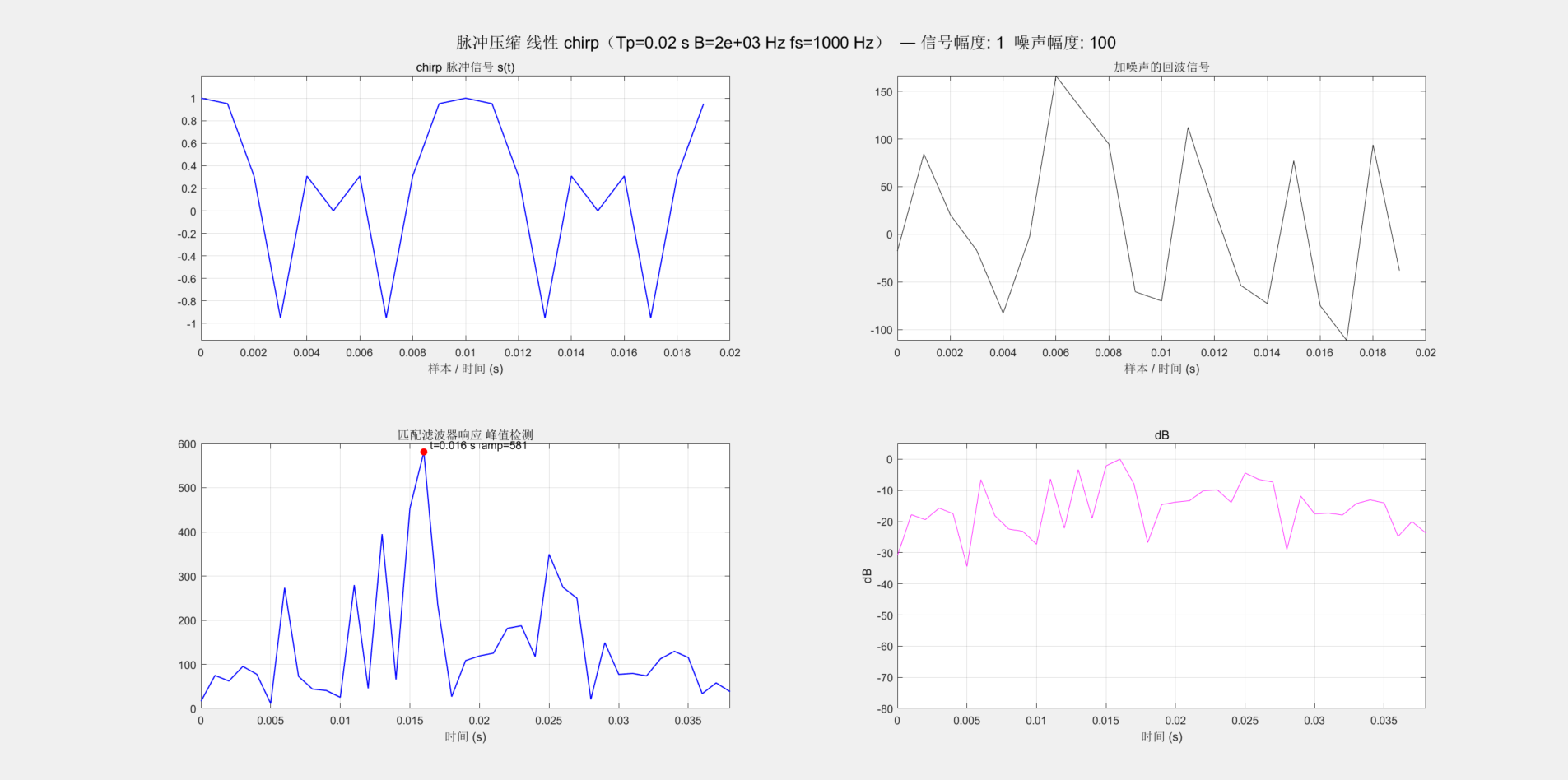
* chirp信号长度是
* 峰值索引是，对应的横坐标（代码里是）为

所以对于chirp信号，**匹配滤波器响应的最高点横坐标理论上应该在秒附近**，卷积实现的自相关函数在零延迟处取最大值，而在MATLAB的conv输出中，这个点对应时间。

但较大的噪声显然影响了匹配滤波器的正常输出，那么怎么来减少噪声对匹配滤波器对回波到达的延迟时间的判断的影响呢？

首先我们固定噪声幅度为100，Tp=0.02s，在不同的采样率下观察匹配滤波器输出的变化。

fs为1e3时



在fs=1e3时，匹配滤波器不能很好地判断回波到达的延迟时间，实际值为0.016s而非0.02s。

增大fs到1e5

此时匹配滤波器输出的实际值为0.018s而非0.02s，较之前误差已经减小，但仍然不完全正确。

增大fs到1e7

此时匹配滤波器已经能很好地分辨出回波到达的延迟时间为0.02s，几乎无误差。

这是因为当把连续时间的chirp回波和模板做匹配滤波时，真正的峰值出现在物理延迟s；离散实现只能在采样网格上“看到”它。采样率越高，采样间隔越小，峰值所在的样本索引更大，而时间量化误差不超过一个采样间隔，因而延迟估计的离散误差上限随线性缩小。更高的还让离散模板更精细地逼近连续信号（尤其是瞬时频率线性变化的chirp），匹配滤波器的冲激响应主瓣形状与旁瓣电平更接近理论，从而峰值时刻在时间轴上更“尖锐”且定位更稳定。

接下里，我们来改变另一个参数带宽B，现在固定分辨率chirp\_fs = 1e3和噪声幅度为100，逐渐增大带宽，来看看匹配滤波器输出的变化。

将chirp\_B从 2e3增加到2e6





发现随着带宽chirp\_B的增加，代码输出结果反而显示匹配滤波器不能很好地分辨出回波到达的延迟时间为0.02s，误差变得很大。

这是因为当把chirp的带宽调得很大，而采样率仍然只有1 kHz时，问题就出现了。线性调频信号的瞬时频率会从扫到，如果 MHz，那就是从 MHz到 MHz，而奈奎斯特条件要求采样率至少大于两倍最高频率，也就是要在MHz级别才行。现在远远不够，结果就是信号在采样后发生严重混叠，原本应该平滑展开的频率被折叠到低频，匹配滤波器看到的已不是理想的chirp，而是一堆被折叠的相位。理论上匹配滤波的主瓣宽度大约是，带宽越大主瓣越窄，时间分辨率越高。但在离散实现里，主瓣宽度对应的采样点数是。当时，这个值小于1，意味着主瓣比一个采样点还窄，峰值落在哪个点完全取决于采样格子和噪声，显示出来的峰值位置就会抖动，无法稳定地指示0.02 s。换句话说，数学上分辨率提高了，但因为采样率太低，离散实现根本捕捉不到这个窄峰，所以我看到的结果就是匹配滤波器“分不清”回波的到达时间。

**实验收获和体会**

在这次实验中，我最大的收获是通过大量仿真数据直观理解了匹配滤波器和脉冲压缩在噪声环境下的性能变化。首先在实验一中，设置Signal\_Amplitude=1，Noise\_Amplitude=0时，相关输出在N0=248处形成幅度约20的尖锐峰值，与理论值A²M=20完全一致；当Noise\_Amplitude=0.5时，峰值仍在248处，但幅度出现轻微波动，相关曲线主峰与噪声起伏相比仍然突出；当Noise\_Amplitude增大到2和5时，接收信号几乎被噪声淹没，直接观察已无法分辨目标，但相关输出在148附近仍能形成峰值，只是峰值位置出现偏移，说明噪声会导致检测延时估计向后漂移。进一步将Signal\_Amplitude提高到10而噪声保持5时，相关峰再次在真实延时处出现，幅度约200，验证了提高信号能量可以显著改善检测性能。实验还比较了不同脉冲宽度M的情况，当M从10增加到60时，相关输出峰值从约10逐渐增加到60，信噪比提升倍数与M成正比，验证了理论公式SNR\_out = A²M/σ²。

在实验二的蒙特卡洛仿真中，固定N=500，M=20，N0=248，重复1000次试验，统计相关器最大值位置的分布。当σ=0时，所有试验的最大值都集中在248，直方图为单一尖峰；σ=0.5时，仍以248为主，但247和249开始出现；σ=1时，分布进一步展宽，±1、±2的误差概率明显增加；σ=5时，分布几乎接近平坦，最大值位置在整个搜索区间内随机出现。统计量显示，当σ=0.5时，时延估计误差均值约0.02，方差约0.3；σ=2时，均值接近0但方差增大到约4.7；σ=5时，方差超过20，说明噪声显著降低了估计稳定性。进一步将蒙特卡洛次数从1000增加到100000，均值误差逐渐收敛到0，验证了估计器在大样本下无偏性。

实验三中，BK码和m序列的脉冲压缩结果展示了序列长度对抗噪声的作用。长度为13的BK码在无噪声时自相关峰值为13，噪声幅度为1时峰值仍在13附近，噪声为5时峰值位置偏移，说明短序列抗噪性差。m序列长度为255时，主峰幅度为255，旁瓣为-1，噪声为0时峰值准确；噪声为5时峰值位置偏移到281，噪声为10时偏移更明显。但当序列长度增加到511和1023时，即使噪声幅度为10，峰值仍准确出现在理论位置，说明长序列通过能量积累显著提高了信噪比。理论上，m序列的脉冲压缩增益为10log10(N)，N=511时约27.1 dB，N=1023时约30.1 dB，这与实验结果一致。

实验四的chirp信号进一步验证了带宽与分辨率的关系。设Tp=0.02s，fs=1e5，B=2e3时，匹配滤波输出主峰出现在0.02s，符合理论Δt≈1/B=0.0005s的分辨率。当噪声幅度增加到100时，峰值位置偏移到0.018s，方差明显增大；噪声为200时，偏差达到0.04s，检测性能严重下降。改变采样率后发现，fs=1e3时估计延时为0.016s，误差0.004s；fs=1e5时误差减小到0.002s；fs=1e7时几乎无误差，说明采样率越高，延时估计越精确。进一步在fs=1e3固定时，将带宽B从2e3增加到2e6，结果显示匹配滤波输出无法稳定指示0.02s，误差显著增大，这是由于采样率不足导致频谱混叠，验证了奈奎斯特采样条件的重要性。

综合来看，本次实验通过大量数据和图形结果验证了匹配滤波器在噪声环境下的信噪比提升作用，展示了蒙特卡洛仿真对估计器统计特性的揭示，说明了序列长度对脉冲压缩增益的决定性作用，以及采样率和带宽对chirp信号分辨率的影响。让我对匹配滤波与脉冲压缩的物理意义和工程应用有了更深刻的理解。