



# 中国研究生创新实践系列大赛

## “华为杯”第十七届中国研究生

### 数学建模竞赛

题目 基于通信仿真的载波恢复算法设计与 ASIC 实现

#### 摘 要：

ASIC 芯片已广泛应用于通信领域，其设计时合理平衡性能和资源，实现具体场景下的最优设计是一项重要工作。本文从题目要求出发，使用 MATLAB 软件构建了信号传输系统仿真框架，包括数字信号生成器、16-QAM 调制和解调、色散效应模拟与补偿、相噪和加性高斯白噪声的模拟。在此框架上对各任务要求分别进行了算法设计与硬件实现。

针对任务一，首先通过傅里叶变换分析了信道中色散效应与相位噪声的耦合机制。然后提出了将色散效应与相位噪声去耦合的方法，即根据色散效应与相位噪声产生的时移和频移进行补偿处理。对经过这种补偿处理后的纯净相位噪声进行了分析，提出了使用分段线性插值算法进行相噪补偿的方法。并经推论和仿真实验验证得到在本算法的假设下采用合适的傅里叶窗口大小，能在满足 RSNR 代价 $<0.3$  dB 条件下使 Pilot 开销最小为  $2/256$ 。最后进行了 ASIC 算法设计（硬件设计），根据题意给出了算法的电路结构。

针对任务二，首先确定一种导频最佳的插入方式，即首尾插入方式。然后通过 MATLAB 响应面实验工具建立色散、线宽与误码率的关系，在合理忽略色散对误码率的影响的前提下，分析不同线宽下误码率与窗口大小的关系和固定线宽条件下窗口大小与 RSNR 代价的关系，从而得到在设定误码率和满足 RSNR 代价 $<0.3$  dB 条件下色散、线宽和 Pilot 开销的关系。当线宽较大时，相噪对误码率的影响较显著，窗口大小减小有利于误码率的降低，同时 RSNR 代价也会减小。

针对任务三，综合考虑定点量化对 CR 算法的性能和资源的影响。在任务二的基础上首先对量化过程中产生的量化噪声进行了建模分析，得到了量化比特数和量化信噪比之间的关系。对算法中的主要运算数据进行了数值统计用以分析量化比特数的分配。并经推论和仿真实验验证得到在不同场景下 Pilot 开销最小与芯片资源消耗相对最少的量化模型方案。最后也进行了 ASIC 算法设计（硬件设计），根据题意给出了算法的改进电路结构，并对整体芯片资源消耗进行了统计计算。

针对任务四，为了实现位宽自动优化并综合考虑性能和资源两种因素，考虑到资源与芯片实现面积和功耗有关，首先建立资源使用量化评价函数。然后以 RSNR 代价作为性能评定指标，以不同位宽向量下的 RSNR 代价与资源使用评价函数分别赋予权重因子，组成综合评价目标函数。最后提出基于禁忌搜索算法的位宽自动优化方法，以得到性能和代价综合评价目标函数的最优 Pareto 解集，即最优位宽设计方案。

最后本文对 CR 算法的相位噪声提取效果以及整个信号传输系统的抗噪声性能进行了检验，检验结果为：该算法能较好地提取出信道中的相位噪声；该算法提高了整个系统的抗噪声性能。

**关键词：**仿真框架；CR 算法；噪声耦合；去耦合；ASIC 算法设计；量化噪声

# 目 录

1. 问题重述	4
1.1 问题背景	4
1.2 问题要求	4
2. 模型假设	5
3. 符号说明	5
4. 模型准备	6
4.1 16QAM 调制	6
4.2 导频插入	6
4.3 色散、相位噪声、加性高斯白噪声	7
4.4 色散补偿	8
4.5 相噪补偿	8
4.6 解调	8
5. 任务一：Pilot 开销最小的 CR 算法设计	10
5.1 问题分析	10
5.2 模型建立	10
5.3 模型求解	11
5.4 ASIC 算法设计（硬件设计）	13
5.4.1 导频相噪提取	14
5.4.2 频移计算	15
5.4.3 色散去耦合计算	15
5.4.4 荷载相噪求解	15
5.4.5 荷载相噪补偿	16
5.4.6 FFT 与 IFFT	16
6. 任务二：色散、线宽与 Pilot 开销间的量化	17
6.1 问题分析	17
6.2 模型建立	17
6.3 模型求解	17
7. 任务三：资源最低的 CR 算法设计	20
7.1 问题分析	20
7.2 模型建立	20
7.3 模型求解	21
7.4 ASIC 算法设计（硬件设计）	23
8. 任务四：自动优化位宽和实现性设计方案的制定	25
8.1 问题分析	25
8.2 模型建立与求解	25
9. 模型检验与评价	27
9.1 模型检验	27
9.2 模型评价	27
9.2.1 模型优点	27
9.2.2 模型缺点	27
参考文献	28
附 录	29

附录 1 主程序代码.....	29
附录 2 RSNR 计算程序代码.....	32
附录 3 插入导频程序代码.....	34
附录 4 插入色散程序代码.....	35
附录 5 插入相噪程序代码.....	36
附录 6 相噪补偿程序代码.....	37

# 1. 问题重述

## 1.1 问题背景

DSP (Digital Signal Process) 即数字信号处理器, 也可称为 DSP 基带芯片, 它能够实现数字信号处理技术的芯片, 其特殊的 DSP 指令, 可以用来快速的实现各种数字信号处理算法<sup>[1]</sup>。这种芯片往往是基于专用集成电路 (ASIC) 实现的, 细小的芯片具有比单光纤更优的容量, 使网络流量增长更加迅猛。ASIC 芯片算法的设计步骤如图 1.1 所示。其 DSP 算法主要分两步, 首先在只考虑浮点计算的前提下根据信道损伤的物理模型设计补偿算法; 然后根据芯片资源和功耗约束, 将算法修整为 ASIC 芯片可实现的定点形式, 该步骤需要将算法细化为芯片上最基本的乘、加等运算, 此外还要考虑定点量化噪声的影响。合理平衡性能和资源, 实现具体场景下的最优设计, 对 DSP 芯片算法工程的应用具有重要意义。

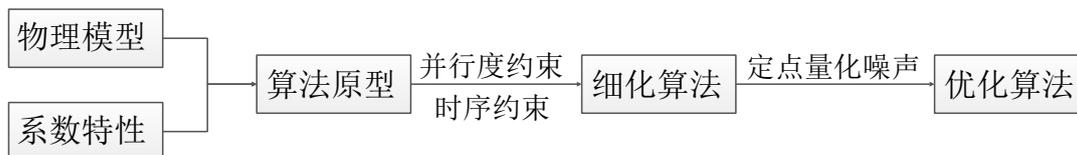


图 1.1 ASIC 芯片算法设计步骤

## 1.2 问题要求

本题在不考虑色散补偿和误码率计算的复杂度和资源, 只需考虑载波恢复 (Carrier Recovery, CR) 优化算法 (计算相噪+补偿相噪) 相关的资源的情况下, 解决以下任务:

### ➤ 任务一: Pilot 开销最小的 CR 算法设计

设置波特率为 150 G baud 的标准 16QAM 信号, 令线宽为 100 kHz, 色散值为 20000 ps/nm, 算法的并行度固定为 128, 不考虑定点量化。以加法、乘法、查表和缓存为基础, 并以 RSNR 代价 < 0.3 dB 为目标, 设计一套 Pilot 开销最小的 CR 算法。

### ➤ 任务二: 色散、线宽与 Pilot 开销间的量化

设置线宽从 10 kHz~10 MHz, 色散值从 0~10000 ps/nm 变化场景, 以 RSNR 代价 < 0.3 dB 为目标, 定量得到色散、线宽与 Pilot 开销的关系。

### ➤ 任务三: 资源最低的 CR 算法设计

设置线宽从 10 kHz~10 MHz, 色散值从 0~10000 ps/nm 变化场景。考虑芯片实现的资源以及定点量化对性能和资源的影响, 且导频开销可任意变化 (前提是须确保净荷的流量为 > 145 G baud), 设计资源最低的 CR 算法。

### ➤ 任务四: 自动优化位宽和实现性设计方案的制定

实际处理过程种性能和资源的合理平衡与具体场景有关。选出任务三中有代表性的一种场景, 给出统筹性的“性能-资源”综合考虑下的算法设计思路, 构造性能和资源的综合代价函数, 给出一套自动优化位宽和实现性设计的方案并量化。

## 2. 模型假设

1. 假设本文 CR 算法中接收端的载波已同步；
2. 假设色散补偿时不考虑定点量化的影响；
3. 假设把加入的色散当作已知进行直接补偿。

## 3. 符号说明

符号	含义	单位
$\hat{S}$	接收信号	V
$S$	发送信号	V
$S_{16QAM}(t)$	经 16QAM 调制后的信号	V
$N$	加性高斯白噪声信号	V
$A$	信号幅度	V
$Pilot_{overhead}$	Pilot 开销	---
$Pilot_{usage}$	Pilot 利用率	---
$d_{16QAM}$	16QAM 信号星座图上信号点间的最小距离	---
$L$	QAM 信号比特位长	---
$M$	QAM 电平数	---
$l$	傅里叶变换窗口长度	---
$H_s(f)$	色散效应的频率响应	---
$\lambda$	波长	nm
$Dz$	色散值	ps/nm
$c$	光速	m/s
$f$	频点	Hz
$\theta_t$	随机相位噪声	rad
$\hat{\theta}_t$	荷载相噪估计值	rad
$LW$	一个激光器线宽指标	Hz
$f_b$	波特率	baud
$X_i$	均值为 0, 方差为 1 的随机变量	---
$P_e$	符号错误概率	---
$E_{av}$	16QAM 的平均能量	J
$\omega_0$	频移	Hz
$Q$	信号功率	W
$C$	资源消耗	U

## 4. 模型准备

本文通过 MATLAB 软件的构建仿真平台完成，主要分两大块——信号传输系统框架的整体构建以及 CR 算法的设计。CR 算法在整个系统中的位置如图 4.1 所示，其中导频插入和相噪补偿步骤属于本文中 CR 算法的两大重要节点。

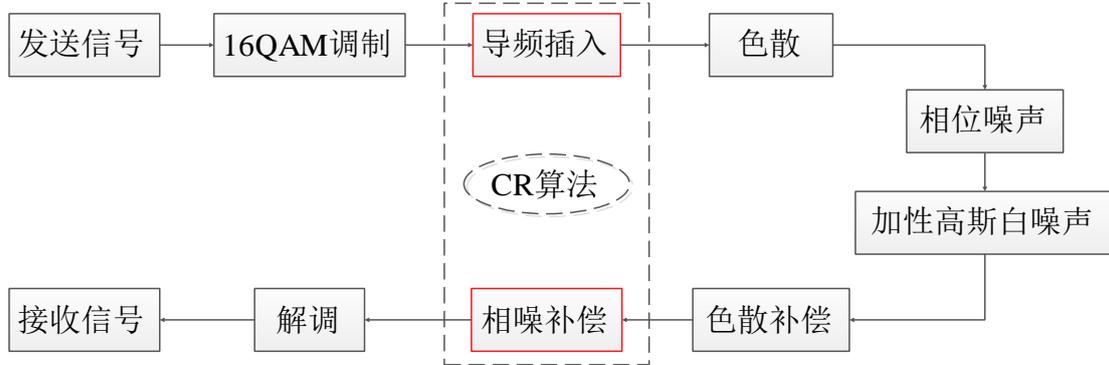


图 4.1 信号传输系统框图

### 4.1 16QAM 调制

对于 16QAM，多元正交振幅调制是两个相互正交的同频载波调制后调制信号的双边带调制，利用这种已调信号在同一带宽内频谱正交的性质来实现两路并行的数字信息传输。16QAM 信号调制框图如图 4.2 所示。

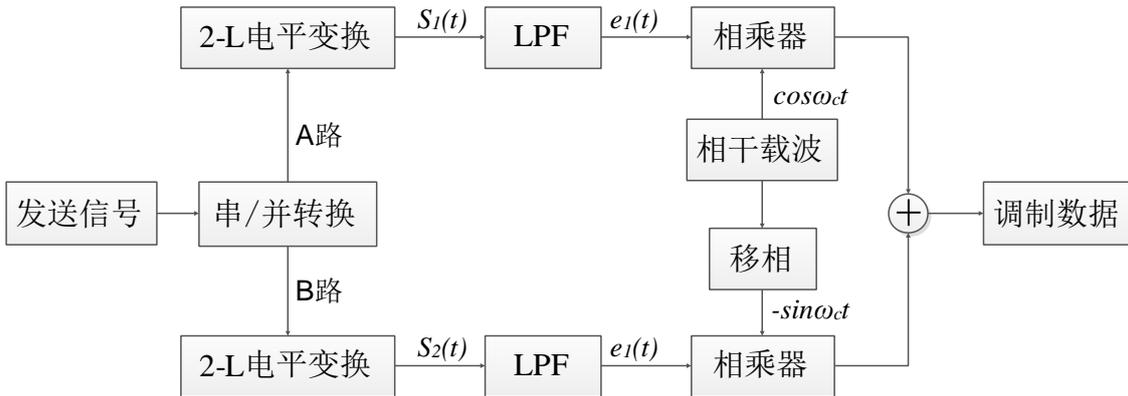


图 4.2 16QAM 信号调制框图

经过 16QAM 调制系统后的信号  $S_{16QAM}(t)$  如式 4-1 所示，式中最小幅度的信号能量  $E_0$  为  $A_0^2 T/2$ 。若已调信号的最大幅度为 1，则 16QAM 信号星座图上信号点间的最小距离  $d_{16QAM}$  如式 4-2 所示。

$$S_{16QAM}(t) = A_0 a_i \cos \omega_0 t + A_0 b_i \sin \omega_0 t = \sqrt{E_0} a_i \phi_1(t) + \sqrt{E_0} b_i \phi_2(t) \quad (4-1)$$

$$d_{16QAM} = \sqrt{2}/(L-1) = \sqrt{2}/(\sqrt{M}-1) = 3\sqrt{2} \quad (4-2)$$

式中： $L$ —QAM 信号比特位长；

$M$ —QAM 电平数。

### 4.2 导频插入

导频的插入方法分为频域导频插入法和时域导频插入法，在信道变化较为缓慢的情况下，时域导频插入法更优<sup>[2]</sup>。鉴于本题主要研究对象为光纤，其信道变化较慢，故本文采用的是时域导频插入法。时域导频数据取星座图上一固定点，在 16QAM 调制后直接插入帧中。然后可形成插入导频后的帧结构，信号以图 4.3 所示的若干长度为  $l$  的帧数据（含

若干单位时域数据) 依次传输。

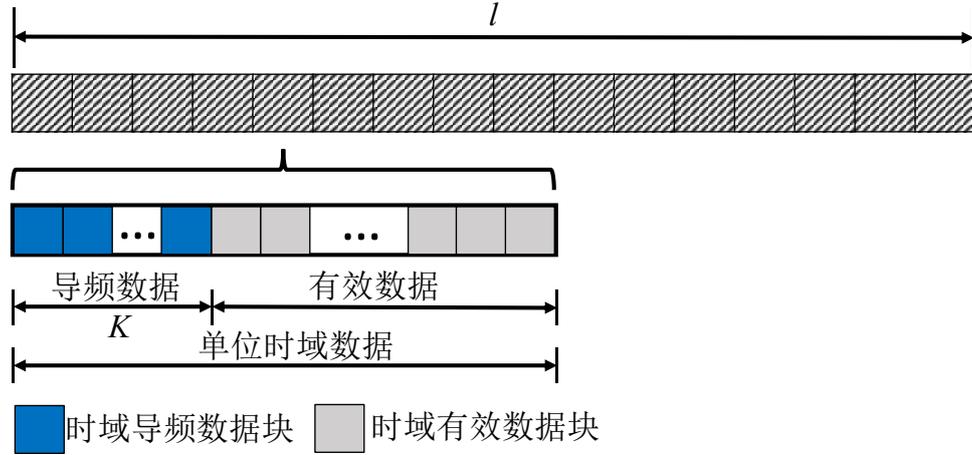


图 4.3 帧结构示意图

### 4.3 色散、相位噪声、加性高斯白噪声

本文通过对信号直接做快速傅里叶变换 (FFT) 后的频域数据施加 1 个随频点平方关系变化的相位 (式 4-3), 最后进行一次逆快速傅里叶变换 (IFFT) 来模拟光纤中的色散效应 (图 4.4)。

$$H_s(f) = e^{i*[(\lambda^2\pi Dz/c)f^2]} \quad (4-3)$$

式中:  $\lambda$ —波长, 取 1550 nm;  
 $Dz$ —色散值;  
 $c$ —光速;  
 $f$ —频点;  
 $i$ —表示虚部.

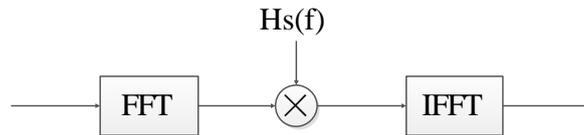


图 4.4 模拟色散效应示意图

相位噪声的模拟则是根据相位噪声的相位差满足式 4-4, 则可得到  $t$  时刻的相位噪声相位如式 4-5 所示, 其中  $\theta_t$  表示随机相位噪声。

$$d\theta = \theta_{k+1} - \theta_k = (\sqrt{2\pi \cdot LW/f_b})X_k \quad (4-4)$$

$$\theta_t = \sum_{i=1}^t d\theta_i = \sum_{i=1}^t \theta_{i+1} - \theta_i = (\sqrt{2\pi \cdot LW/f_b}) \sum_{i=1}^t X_i \quad (4-5)$$

式中:  $LW$ —一个激光器线宽指标, kHz;  
 $f_b$ —波特率;  
 $X_i$ —均值为 0, 方差为 1 的随机变量.

进一步推导得到均值  $E(\theta_t)$  和方差  $D(\theta_t)$  分别如式 4-6、4-7 所示:

$$E(\theta_t) = (\sqrt{2\pi \cdot LW/f_b}) \sum_{i=1}^t E(X_i) = 0 \quad (4-6)$$

$$D(\theta_t) = \sum_{i=1}^t D((\sqrt{2\pi \cdot LW/f_b})X_i) = (2\pi \cdot LW/f_b) \sum_{i=1}^t D(X_i) = (2\pi \cdot LW/f_b)t \quad (4-7)$$

所以得到相位噪声是一个服从均值为 0，方差为  $(2\pi \cdot LW/f_b)t$  的分布。

加性高斯白噪声是在时域上与原信号直接进行加性叠加的噪声，服从的是高斯分布。仿真中通过 MATLAB 软件中的信道函数 awgn 模拟得到。

#### 4.4 色散补偿

本文中色散补偿依据题意，先将接收到的数据通过 FFT 转换到频域，再乘式 4-3 的共轭，随后再通过 IFFT 转换回时域。

#### 4.5 相噪补偿

相噪补偿前首先需要对导频进行提取，本文通过将发送端经过 16QAM 调制后的信号在星座图上的位置用极坐标形式表示为  $g(t)=Ae^{i\varphi(t)}$ ；信号在叠加上信道中的相位噪声后表示为  $g'(t)=Ae^{i[\varphi(t)+\theta(t)]}$ ；在接收端将接收到的导频信号与已知的导频信号相除即可得到 t 时刻的信道相位噪声如式 4-8 所示，完成导频提取。

$$g'(t)/g(t) = e^{i\theta t} \quad (4-8)$$

然后通过插值计算出荷载的相位噪声估计值进而进行补偿，一般而言插值会使用最小二乘法。接收信号被同步处理后可认为信道是平坦的，则接收信号频域如式 4-9 所示：

$$\hat{S} = S e^{i\theta_t^z} + N \quad (4-9)$$

式中： $\hat{S}$ —接收到的信号；  
 $S$ —所发送的信号；  
 $\theta_t^z$ —荷载相噪估计值；  
 $N$ —加性高斯白噪声信号。

本文基于最小二乘法对荷载相噪的估计值  $\hat{\theta}_t^z$  进行确定，以使得接收和发送端的平方误差最小，即  $\hat{\theta}_t^z$  的损失函数  $F(\hat{\theta}_t^z)$  (式 4-10) 最小。对  $F(\hat{\theta}_t^z)$  进一步求导并令其为 0，则可得到式 4-11。由此可得到基于最小二乘法的信道估计值  $\hat{\theta}_{tLS}^z$  如式 4-12 所示。

$$F(\hat{\theta}_t^z) = \|\hat{S} - S\hat{\theta}_t^z\|^2 = \hat{S} \cdot \hat{S} - \hat{S} \cdot S\hat{\theta}_t^z - \hat{\theta}_t^z \cdot S \cdot \hat{S} + \hat{\theta}_t^z \cdot S \cdot S\hat{\theta}_t^z \quad (4-10)$$

$$\partial F(\hat{\theta}_t^z)/\partial \hat{\theta}_t^z = -\hat{S} \cdot S + \hat{\theta}_t^z \cdot S \cdot S = 0 \quad (4-11)$$

$$\hat{\theta}_{tLS}^z = \hat{S}/S = (S\hat{\theta}_t^z + N)/S = \hat{\theta}_t^z + S^{-1}N \quad (4-12)$$

#### 4.6 解调

16QAM 信号采用正交相干解调方法时，其解调器工作原理如图 4.5 所示。

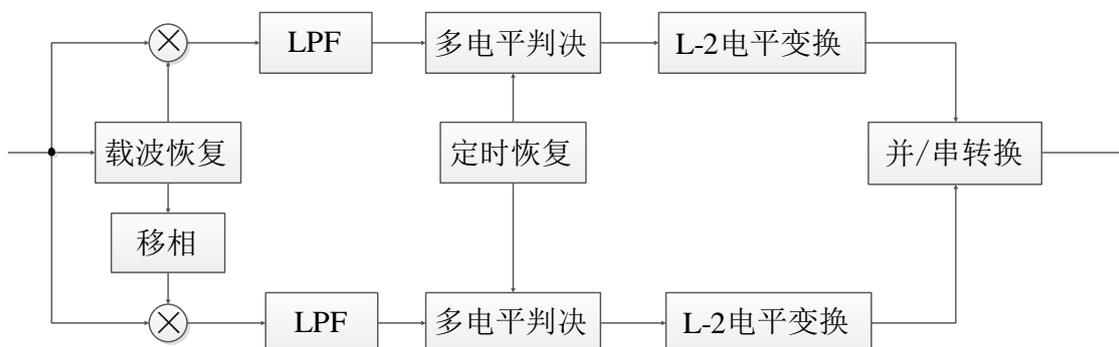


图 4.5 16QAM 解调器工作原理图

对 16QAM，它是相干解调输出的。其同相与正交支路“相关器”输出的信号加噪声混合值如式 4-13 所示，由此计算得符号错误概率  $P_e$  如式 4-14 所示。16QAM 的平均能量  $E_{av}$  如式 4-16 所示，以平均能量表示平均误符号概率如式 4-17 所示。

$$\begin{cases} x_1 = \int_0^\tau x(t)\varphi_1(t)dt = \sqrt{E_0}a_i + n_{l1} \\ x_2 = \int_0^\tau x(t)\varphi_2(t)dt = \sqrt{E_0}b_i + n_{l2} \end{cases} \quad (4-13)$$

$$P_e = 1 - P_c = 1 - (1 - P_{e1})^2 = 2P_{e1} - P_{e1}^2 = [2(L-1)\text{erfc}(\sqrt{E_0/n_0})]/L \quad (4-14)$$

$$\text{erfc}(x) = 1 - 2/(\sqrt{\pi} \int_0^x e^{-t^2}) \quad (4-15)$$

$$E_{av} = 2 \left[ (2E_0/L) \sum_{i=1}^{0.5L} (2i-1)^2 \right] = 2(M-1)E_0/3 = 10E_0 \quad (4-16)$$

$$P_e \approx 2(1 - L^{-1})\text{erfc}\left\{\sqrt{3E_{av}/[2(M-1)n_0]}\right\} = 1.5\text{erfc}\left[\sqrt{E_{av}/(10n_0)}\right] \quad (4-17)$$

## 5. 任务一：Pilot 开销最小的 CR 算法设计

### 5.1 问题分析

任务一的本质是信号模型建立问题，需要本文在 RSNR 代价 < 0.3 dB 的目标下，重点对导频插入方式和相噪补偿进行 Pilot 开销最小的算法设计。以往研究都是基于工程角度考量，一些细节工作未深入开展。本任务中加性高斯白噪声影响较直观，而色散补偿和相位噪声的耦合机制对信号的影响较大。这种耦合关系具体体现在：

首先发送端发送出的 16QAM 调制信号如式 5-1 所示：

$$S_1(t) = A(t)e^{i\varphi(t)} \quad (5-1)$$

该信号经过信道中的色散效应后，被叠加上了色散噪声，表现为在频域乘上了一个随频率变化的相应  $H(\omega)$ ，见式 4-3。此时信号  $S_2(t)$  如式 5-2 所示，其中  $\mathcal{F}^{-1}$  表示 IFFT， $\mathcal{F}$  表示 FFT。

$$S_2(t) = \mathcal{F}^{-1} \{ \mathcal{F} [S_1(t)] \cdot H(\omega) \} \quad (5-2)$$

再叠加相位噪声  $\theta(t)$ ，得到的信号  $S_3(t)$  如式 5-3 所示：

$$S_3(t) = \mathcal{F}^{-1} \{ \mathcal{F} [S_1(t)] \cdot H(\omega) \} \cdot e^{i\theta(t)} \quad (5-3)$$

最后叠加上加性高斯白噪声得到接收端信号  $\hat{S}(t)$  模型如式 5-4 所示：

$$\hat{S}(t) = \mathcal{F}^{-1} \{ \mathcal{F} [S_1(t)] \cdot H(\omega) \} \cdot e^{i\theta(t)} \quad (5-4)$$

依题意，在接收端接收到信号后首先要进行一个色散补偿处理，该处理是将接收到的信号在频率域乘上已知的色散相应的共轭。为了方便进行分析，将信号改为在频域表示，可以得到进行色散补偿后的信号  $\hat{S}_1(t)$  如式 5-5 所示：

$$\mathcal{F} [\hat{S}_1(t)] = \{ \mathcal{F} [S_1(t)] \cdot H(\omega) \} * \mathcal{F} [e^{i\theta(t)}] \cdot \text{conj}[H(\omega)] \quad (5-5)$$

由色散效应的定义与傅里叶变换的时移特性公式 5-6 可知，发送端信号在经过色散效应后，实际上是在信号的不同频率分量在信道传播中被施加了不同的时移，且频率越高，被施加的时移会越大。

$$\mathcal{F} [f(t - t_0(\omega))] = F(\omega) e^{-i\omega t_0(\omega)} \quad (5-6)$$

同时，由傅里叶变换的频移特性（式 5-7）可知，在时域上乘以的相位噪声信号实际上对信号的不同频率分量施加了一个频率偏移。

$$\mathcal{F} [f(t) e^{i\omega_0 t}] = F(\omega - \omega_0) \quad (5-7)$$

这就使得色散补偿后色散噪声虽得到解决，但相位噪声因此变得更加复杂而难以补偿。故本文将该部分作为重点处理对象，结合导频插入方式的优化。

### 5.2 模型建立

经过模型准备，信号传输系统框架已构建完毕。根据本任务具体要求，整体上对接收信号进行模型建立。本模型主要分为两部分——导频插入方式的优化和相位噪声补偿优化，其中相位噪声补偿优化分为在色散补偿后对于被色散补偿影响的相位噪声进行修正，以及相位噪声补偿两部分。

信号的傅里叶变换在时域和频域上的分解示意图如图 5.1 所示<sup>[3]</sup>。傅里叶变换能够将原信号分解成为不同频率分量的叠加，而本题中的色散效应是对信号的不同频率分量叠加上了与频率的平方相关的时延。

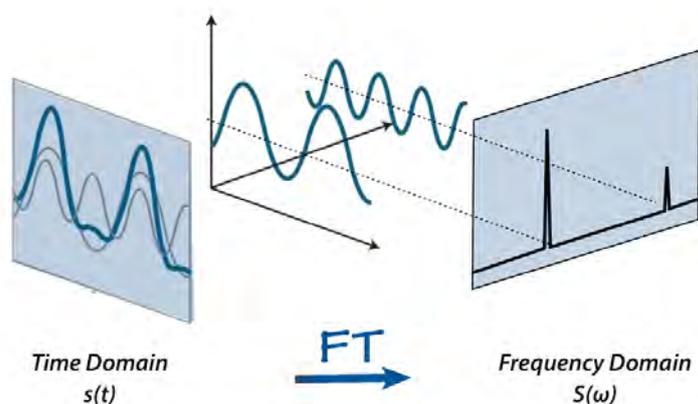


图 5.1 信号的傅里叶变换在时域和频域上的分解示意图

在叠加上了相位噪声后，会对原信号叠加上一个频率移动，使得频率分量的频率发生变化。这样就使得先后叠加了色散与相噪的信号的频率分量的时延实际上并不是正比于频率的平方，而是有一个相噪影响发生频移前后的频率的平方差的误差。

而在进行色散补偿后，会对当前信号的频率分量施加与叠加色散效应符号相反的时延，但是由于相位噪声的影响使得频率分量的位置已经产生了移动，所以色散补偿后的信号的不同频率分量还是会有一个与前后两个频率平方差相关的误差。故在进行相噪补偿前需要先根据相位噪声产生的频移来对信号做一次这样的色散与相噪的去耦合补偿。该补偿的频率响应如式 5-8 所示：

$$H_s(f) = e^{i[(\lambda^2 \pi D z / c)[f^2 - (f - \omega_0)^2]} \quad (5-8)$$

### 5.3 模型求解

对相位噪声的演化进行仿真，得到了如图 5.2 结果。根据模型简化原理，本文以傅里叶变换窗口  $W(\text{FFT})$  内的与采样点数线性相关的随机噪声为研究对象。

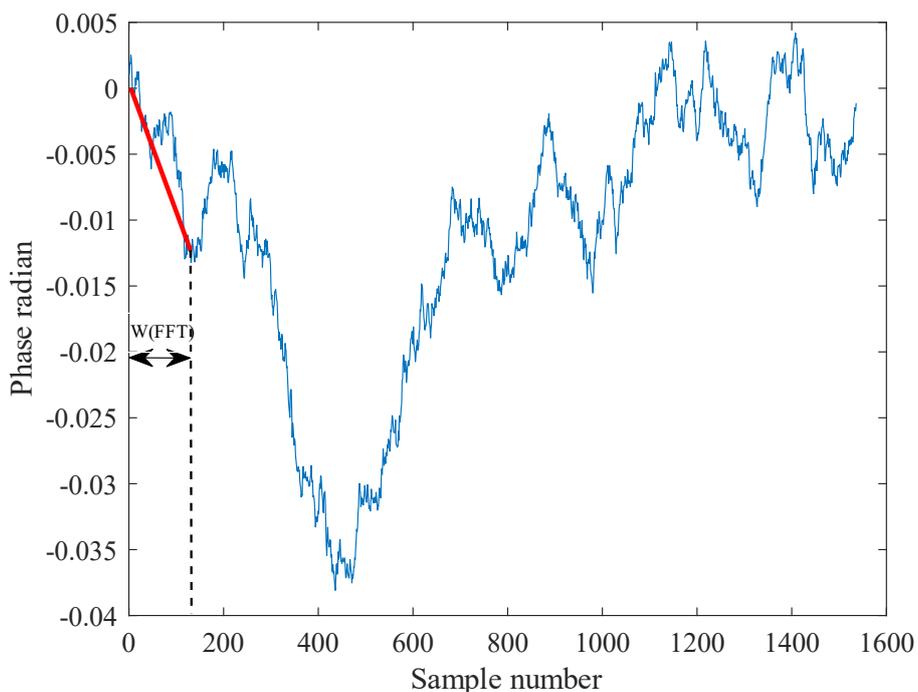


图 5.2 相位噪声演化曲线

在相位噪声在傅里叶变换窗口是线性变化的前提下，可以得到色散与相噪的去耦合补

偿中频移的表达式如式 5-9 所示：

$$\omega_0 = (\theta_l - \theta_1)/l \quad (5-9)$$

因此本文采用在窗口首尾插入导频的方式，同时一个窗口的长度与一帧数据的长度相同，作为本任务导频插入方式的优化结果。

本文依据题设信号条件进行了仿真实验，对导频提取后的相位差进行了计算，计算出了 3150 个样本相位差值，并进一步得到相邻导频相位差间轨迹夹角柱状及对应正态分布图如图 5.3 所示。观察得到大部分夹角在  $0^\circ$  附近，且最大夹角不超过  $11^\circ$ ；查阅相关资料得到以往研究认为两点间夹角小于  $30^\circ$  即可使用直线近似拟合其运动轨迹<sup>[4]</sup>，故本文假定通过最小二乘法拟合的函数为一次函数，即对相邻两导频相位噪声间的星座图运动轨迹采用线性拟合。

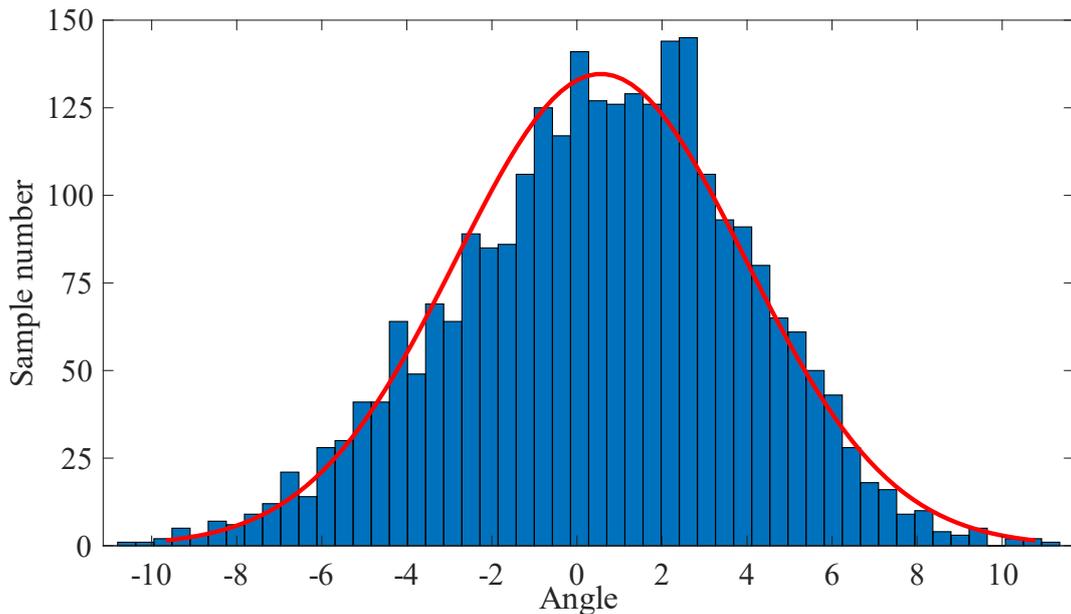


图 5.3 相邻导频相位差间轨迹夹角柱状及对应正态分布图

插值是离散函数逼近的重要方法，利用它可通过函数在有限个点处的取值状况，估算出函数在其他点处的近似值<sup>[5]</sup>。综上所述，本文中相噪补偿采用是分段线性插值算法。具体计算流程为设已知坐标  $(x_0, y_0)$  与  $(x_1, y_1)$ ，通过式 5-10 则可得到  $[x_0, x_1]$  上某一  $x$  在该映射上的  $y$  值。

$$\frac{y-y_0}{y_1-y_0} = \frac{x-x_0}{x_1-x_0} \quad (5-10)$$

相噪的去耦合补偿部分是根据提取出的导频的相位差得到对应频偏，并根据得到的频偏给信号不同频率分量对应的补偿。经过这个处理后本文认为信号的色散效应已经完全去除了。之后的相噪补偿根据之前的假设与推到本文选择使用分段线性插值进行补偿。

本算法采用将导频在窗口首位插入的方式，同时由于本文的线性相位假设，插入多于两个的导频对于性能提升影响很小，因此算法的 Pilot 开销只与窗口长度  $l$  相关。本文对于在题目给定的场景下 RSNR 代价与窗口长度  $l$  进行了仿真，得到图 5.4。由于要求算法并行度为 128，因此窗口长度  $l$  应该为 2 的幂次方，这样才能让硬件电路最大效率工作。从图中可以看出，在窗口长度为 256 能够在满足 RSNR 代价  $< 0.3$  dB 的条件下达到最大的窗口长度，即最小的 Pilot 资源开销，此时开销为  $2/256$ 。

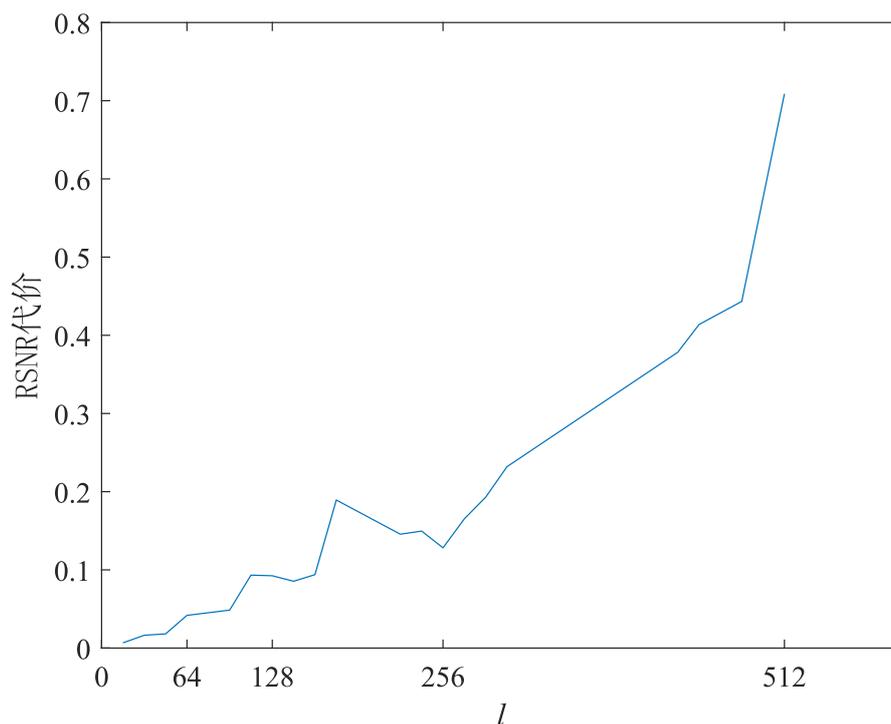


图 5.4 RSNR 代价与窗口长度的关系

#### 5.4 ASIC 算法设计（硬件设计）

本文整体 ASIC 算法设计框架如图 5.5 所示，其中的运算处理主要由一系列图中的相噪补偿 PE 组成的 PE 阵列进行。每一个相噪补偿 PE 同时进行窗口长度  $l$  个计算，因此在问题一的算法并行度 128 的要求下，由  $128/l$  个 PE 构成 PE 阵列。相噪补偿 PE 主要分为两个部分，即色散耦合补偿与相噪补偿。其中色散耦合补偿由导频提取，频移计算，色散与相噪去耦合补偿计算，相乘器阵列，FFT 与 IFFT 构成。相噪补偿模块由导频相噪提取、荷载相噪求解、荷载相噪补偿构成。假设输入信号经过了串/并转换，得到的信号存储在图 5.5 的 Buffer 模块中。导频对应的星座图坐标为  $3+3i$ 。

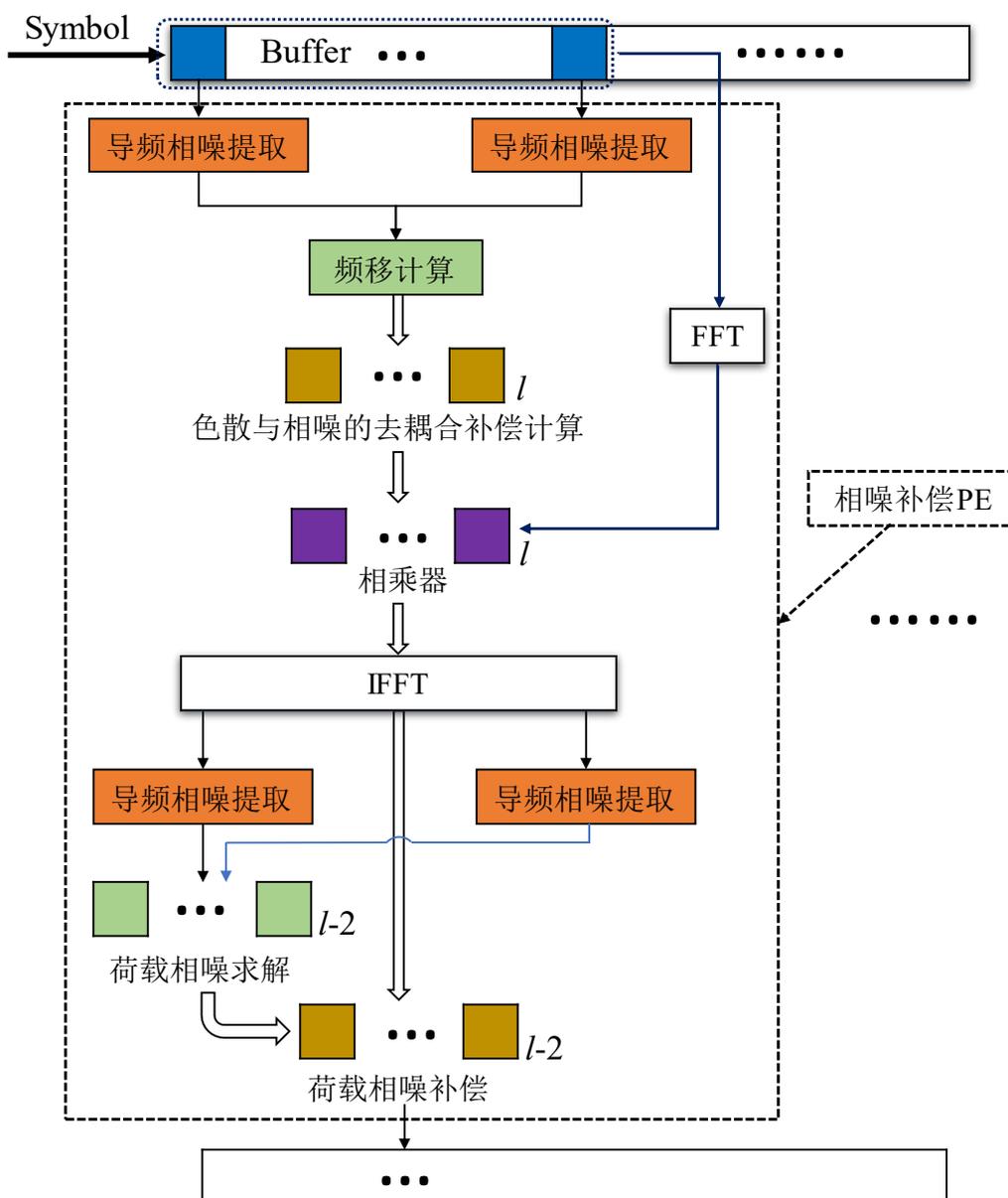


图 5.5 ASIC 算法设计框架图

#### 5.4.1 导频相噪提取

根据式 4-8 可以使用接收序列中的导频信号除以已知的导频信号来提取出传输过程中叠加上的相噪信号，在硬件上为了方便实现使用  $x+iy$  ( $i$  表示虚部) 来表示信号。计算如式 5-11 所示：

$$\frac{x+iy}{3+3i} = \frac{(x+iy)(3-3i)}{(3+3i)(3-3i)} = \frac{x+y+i(y-x)}{6} \quad (5-11)$$

该模块电路结构如图 5.6 所示：

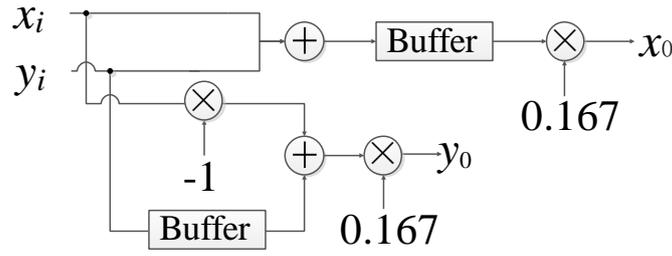


图 5.6 导频相噪提取模块电路结构

#### 5.4.2 频移计算

根据式 5-9 可以求出相噪产生的频移的估计值，这里分为两步计算：让两个导频信号相除，得到的信号包含的相位是两个导频信号相位噪声的差，如式 5-12 所示；再从信号中提取出相位信号，这里采用反正弦函数求得（式 5-13），使用 LUT 查表实现，最后将提取出的相位信号除以窗口长度。其中的除法都采用 LUT 查表加乘法器实现，该模块电路结构如图 5.7 所示。

$$x_0 + iy_0 = [x_{p+1}x_p + y_{p+1}y_p + i(x_p y_{p+1} - x_{p+1}y_p)] / (x_p^2 + y_p^2) \quad (5-12)$$

$$\omega_0 = \arcsin(y_0/x_0)/l \quad (5-13)$$

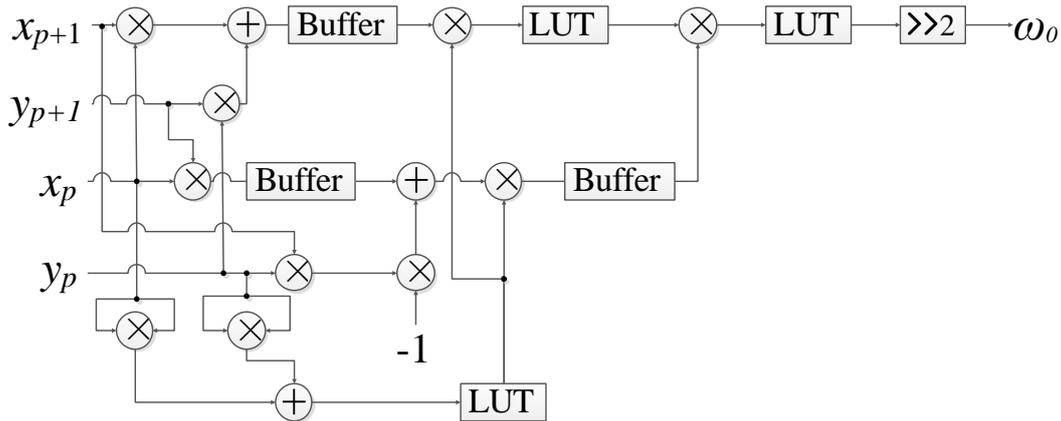


图 5.7 频移计算模块电路结构

#### 5.4.3 色散去耦合计算

根据式 5-8 可以得到对应的色散去耦合响应如式 5-14，其中  $\lambda^2 \pi D z / c$  是个常数，我们可以直接预存在电路中。而  $f$  代表的是傅里叶变换后的频率分量，它只与采样率有关，因此在傅里叶变换窗口长度与系统的参数确定后，这里将  $\lambda^2 \pi D z / c$  与  $f$  在电路中直接作为常数处理， $\lambda^2 \pi D z / c$  等于常数  $B$ 。同时由于计算中还需要进行指数运算，故采用一个 LUT 查找表结构来实现。该模块电路结构如所示。

$$H_s(f) = e^{i * [(\lambda^2 \pi D z / c) [f^2 - (f - \omega_0)^2]]} = e^{i * B [f^2 - (f - \omega_0)^2]} \quad (5-14)$$

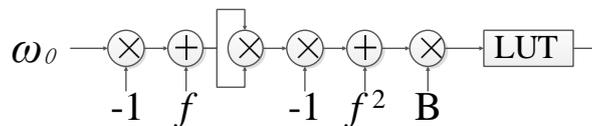


图 5.8 色散去耦合模块电路结构

#### 5.4.4 荷载相噪求解

根据式 5-11 可由导频的相噪推出载荷相噪  $e^{i\theta_n^z}$  如式 5-15 所示：

$$e^{i\theta_n^z} = \frac{(e^{i\theta_{p+1}} - e^{i\theta_p})n}{M} e^{i\theta_p} = \frac{[x_{p+1} - x_p + i(y_{p+1} - y_p)](x_p + iy_p)n}{8} \quad (5-15)$$

式中  $n$  表示两个相邻导频间载荷的编号，其数值从 1 到  $l-2$  范围变化。同样，其中的分母是 2 的幂次方。载荷相噪求解模块由  $l-2$  个单元组成，载荷相噪求解单元模块如图 5.9 所示。

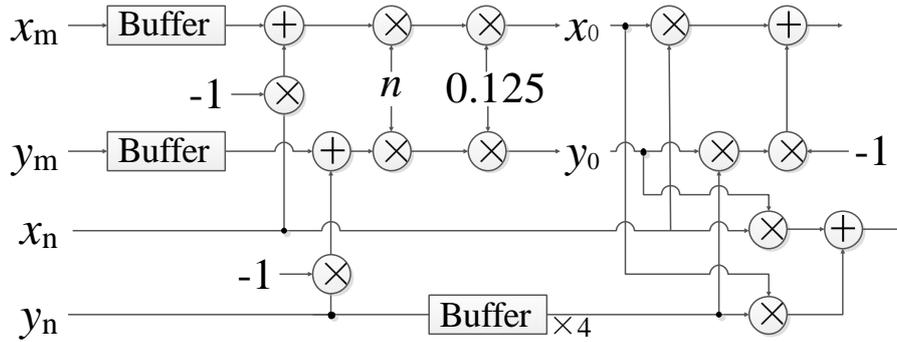


图 5.9 载荷相噪求解单元模块

#### 5.4.5 载荷相噪补偿

在得到载荷相噪估计后，将接收载荷  $\widehat{S}_n^z(t)$  与估计值的共轭  $e^{-i\theta_n^z}$  进行相乘以对载荷进行相位噪声补偿，如式 5-16 所示。

$$\begin{aligned} S_n^z(t) &= \widehat{S}_n^z(t) \cdot e^{-i\theta_n^z} = (\widehat{x}_n^z + i\widehat{y}_n^z)(x_n^z - iy_n^z) \\ &= \widehat{x}_n^z x_n^z + \widehat{y}_n^z y_n^z + i(x_n^z \widehat{y}_n^z - \widehat{x}_n^z y_n^z) \end{aligned} \quad (5-16)$$

载荷相噪求解模块由  $l-2$  个单元组成，载荷相噪补偿单元模块如图 5.10 所示：

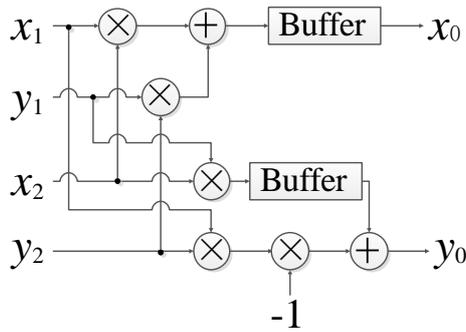


图 5.10 载荷相噪补偿单元模块

#### 5.4.6 FFT 与 IFFT

FFT 指的是快速傅里叶变换，IFFT 指的是逆快速傅里叶变换；在本算法中的作用是使用乘加运算将信号在时域与频域之间进行转换。

## 6. 任务二：色散、线宽与 Pilot 开销间的量化

### 6.1 问题分析

信号在传输过程中经过了色散效应，相位噪声，高斯白噪声的乘性或加性的叠加影响。对于每个噪声单独定性分析，色散  $Dz$  的增大会使得色散效应变强，按照题意接收端的色散补偿处理可以完全补偿掉色散效应，但是可能会因为色散效应与相位噪声的耦合而使得对于相位噪声的影响变大。同时线宽是直接影响相位噪声的，线宽越大，等价于直接增加相位噪声的系数，必然会使得相位噪声直接增大。本题中要求了  $RSNR$  代价的上限，因此为了保证  $RSNR$  代价不高于阈值，需要调整 CR 算法将信噪比再次补偿回来。在相同的相位补偿处理下，随着导频间隔减小，即导频密度的增加，相噪补偿的效果应该会提升，即相应的信噪比会增加，即猜测随着色散与线宽的增加，在保证  $RSNR$  代价不超过阈值的情况下，Pilot 开销会增加。

### 6.2 模型建立

为求解色散、线宽与 Pilot 开销之间的定量关系，为减少计算复杂度，首先需要确定一种导频最佳的插入方式。通过在不同导频间距设置下的仿真实验结果发现，当导频符号设置在每一帧的首部和尾部时，对应的 BER 最低，即相噪恢复的效果最好。仿真实验得到的导频间距与 BER 的关系如图 6.1 所示。

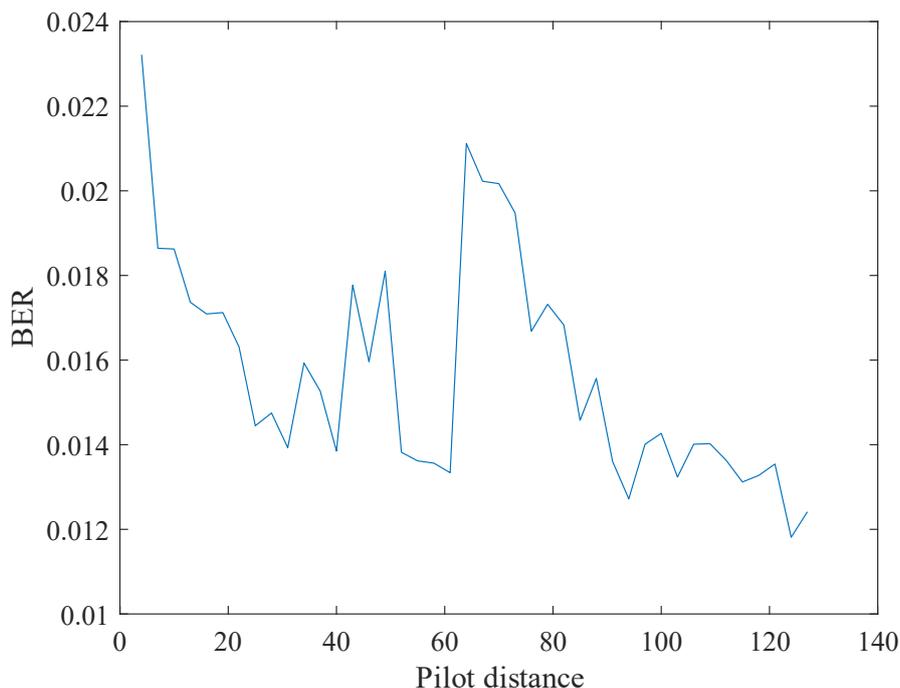


图 6.1 不同导频间距条件下的 BER

因此将导频符号固定在每一帧的首尾，并固定导频长度为 1，即每帧的第一帧和最后一帧为导频符号。该帧结构下的 Pilot 开销仅与窗口长度  $l$  有关。

$$Pilot_{overhead} = 2/l \quad (6-1)$$

$$Pilot_{usage} = (l - 2)/l \quad (6-2)$$

于是该问题可转化为当 SNR、导频插入方式确定的条件下，求解色散、线宽和为达到某一 BER（此处为  $2 \times 10^{-2}$ ）所需的窗口长度（即 Pilot 开销）的关系。

### 6.3 模型求解

在 SNR=14、窗口长度为 128 的情况下，通过仿真实验模拟不同色散  $Dz$ 、线宽  $LW$  组合下的 BER，得到图 6.2 所示的结果。通过多变量线性回归可知 BER 与线宽呈显著线性

关系，与  $Dz$  无显著关系。回归多项式如式 6-3 所示：

$$BER = 0.0123LW + 9.2493 \times 10^{-10}Dz + 9.3142 \times 10^{-6} \quad (6-3)$$

均方根误差为 0.0010。

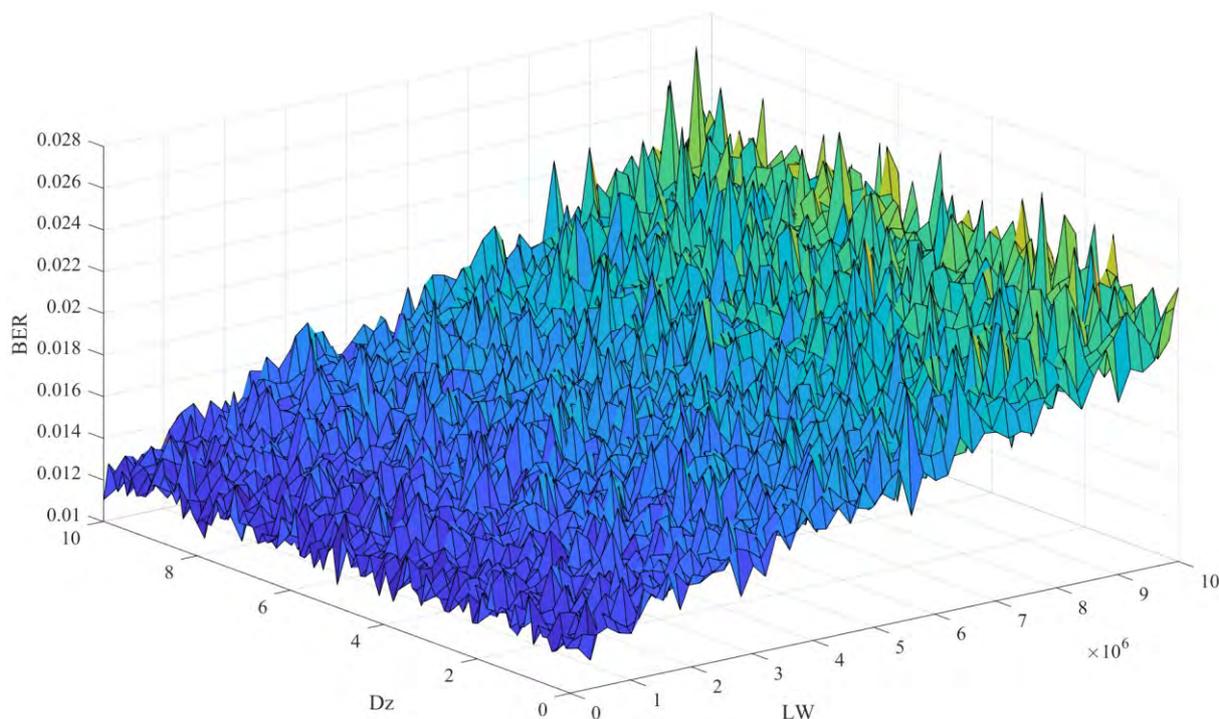


图 6.2 色散  $Dz$ 、线宽  $LW$  变化时对应的 BER (SNR=14)

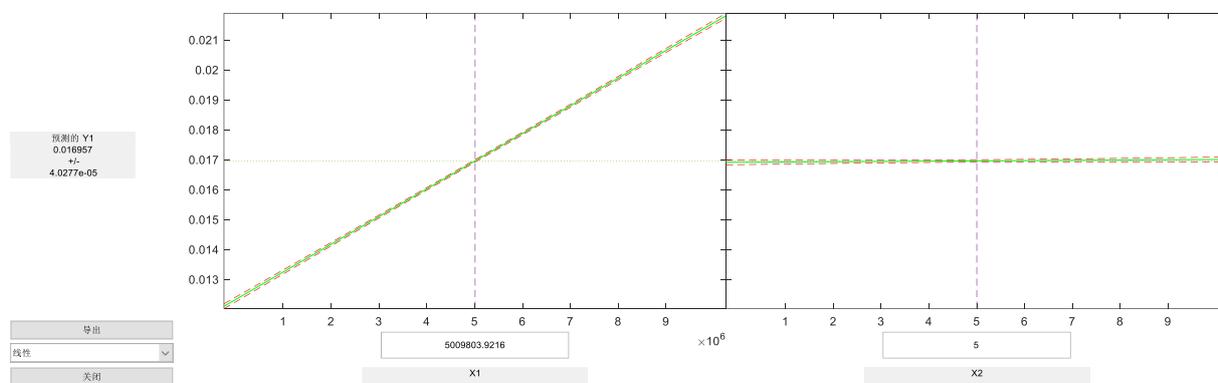


图 6.3 多变量线性回归实验

通过多变量线性回归实验得到的  $Dz$  的系数为  $9.2493 \times 10^{-10}$ ，而  $Dz$  的变化范围为  $0 \sim 10000$  ps/nm，或  $0 \sim 10$  s/m（单位换算后），故色散  $Dz$  对 BER 影响可忽略，即色散补偿过程可较好地消除色散对信号的影响。

忽略色散影响后，在不同线宽和窗口长度条件下通过实验模拟得到 BER，结果如图 6.4 所示。从图 6.4 可知，当线宽在  $10^4 \sim 10^6$  Hz 范围内，BER 总体呈随窗口长度增加而下降的趋势；当线宽大于等于  $5 \times 10^6$  Hz 时，BER 呈随窗口长度增加先下降后上升的趋势。在不同窗口长度条件下通过实验模拟得到 RSNR 代价，结果如图 6.5 所示，RSNR 代价随窗口长度的变化关系与 BER 和窗口长度的变化关系相似，也是当线宽大于或等于  $5 \times 10^6$  Hz 时，RSNR 代价随窗口长度的增加而显著增加，表明当线宽较大时还需要考虑 RSNR 代价限制条件的影响。

产生这种现象的原因可能是相噪的大小与线宽有关，当线宽较小时，相噪对 BER 的影

响与加性白噪声相比不显著，窗口长度对 RSNR 代价影响不显著；当线宽较大时，相噪对 BER 的影响较显著，RSNR 代价随窗口长度增加而增加。因而在线宽较大时，窗口长度减小有利于 BER 的降低，同时 RSNR 代价也会减小。

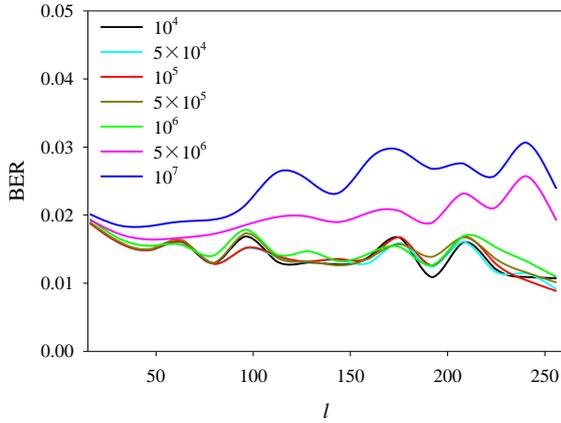


图 6.4 不同线宽下 BER 与窗口长度的关系 (SNR=14)

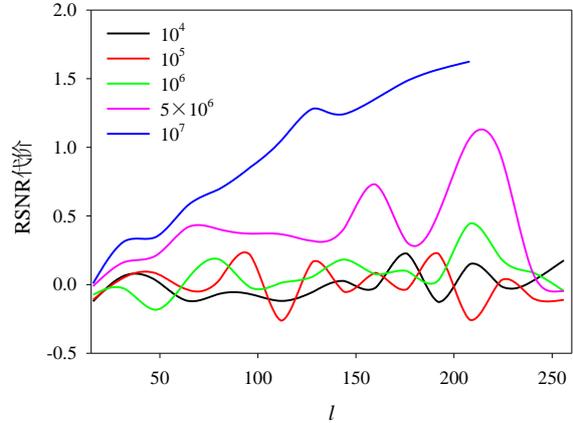


图 6.5 RSNR 代价与窗口长度的关系

从图 6.4 和图 6.5 可知，当设定目标 BER 为  $2 \times 10^{-2}$  时，满足 RSNR 代价  $< 0.3$  dB 条件的不同线宽下对应的最优 Pilot 开销如表 6-1 所示，线宽和 Pilot 开销的关系为：随着线宽的增加，Pilot 开销整体呈下降趋势；但当线宽大于或等于  $5 \times 10^6$  Hz 时，Pilot 开销显著增加，这是因为 RSNR 代价随窗口长度的增加而显著增加，说明当线宽较大时还需要考虑 RSNR 代价限制条件的影响。

表 6-1 线宽和 Pilot 开销的关系

线宽(Hz)	窗口长度	Pilot 开销
$10^4$	256	0.78%
$5 \times 10^4$	256	0.78%
$10^5$	256	0.78%
$5 \times 10^5$	256	0.78%
$10^6$	128	1.56%
$5 \times 10^6$	64	3.13%
$10^7$	32	6.25%

鉴于本模型通过多变量线性回归实验得到的色散 ( $Dz$ ) 系数很小 ( $9.2493 \times 10^{-10}$ )，且题干给出的  $Dz$  变化范围乘上该系数后变化几乎可以忽略。因此本文认为在  $Dz \in (0, 10000$  ps/nm) 时， $Dz$  与 BER 没有关系，进而与 Pilot 开销没有关系。

## 7. 任务三：资源最低的 CR 算法设计

### 7.1 问题分析

在任务一与任务二的基础上，为了能够让算法更好地在硬件上实现，需要对于设计出的算法模型进行定点量化。但是在量化的过程中必定会产生误差，即在数据中引入量化噪声。量化噪声应该与量化间隔成正比，即量化间隔越小，引入的量化噪声就越小，但是随着量化间隔的减少，运算中需要的比特位会增加，这样会带来消耗资源的直接增长。同时，算法在计算过程中不可避免会产生小数，所以小数点固定在第几位也是量化过程中很重要的问题。综上，在定点量化的过程中重点解决两个问题：小数点的位置与量化间隔的选择。与此同时，为了让进行量化后的模型在问题二的场景中同样达到 RSNR 代价 < 0.3 dB 的要求，需要尽量让量化噪声小；为了让算法模型消耗的资源尽量少，需要让量化长度尽量短，而这两者是互相矛盾的，因此需要根据实际信号情况在性能与资源进行综合考量。

### 7.2 模型建立

本任务主要进行信号的量化：设信号的取值范围为  $(a, b)$ ，量化电平数  $M$ ，可以得到量化间隔为  $\Delta v = (b-a)/M$ 。设量化后的信号为  $m_1, m_2, \dots, m_i$ 。当输入信号在取值区间内均匀分布时，可以得到量化噪声功率的平均值如式 7-1 所示<sup>[6]</sup>：

$$\begin{aligned} N_q &= \sum_{i=1}^M \int_{m_{i-1}}^{m_i} (m_k - q_i)^2 f(m_k) dm_k = \sum_{i=1}^M \int_{m_{i-1}}^{m_i} (m_k - q_i)^2 \left(\frac{1}{2a}\right) dm_k \\ &= \sum_{i=1}^M \int_{-a+(i-1)\Delta v}^{-a+i\Delta v} \left(m_k + a - i\Delta v + \frac{\Delta v}{2}\right)^2 \left(\frac{1}{2a}\right) dm_k = \sum_{i=1}^M \left(\frac{1}{2a}\right) \left(\frac{\Delta v^3}{12}\right) \\ &= M\Delta v^3/24a \end{aligned} \quad (7-1)$$

因为  $M\Delta v = 2a$ ，所以有：

$$N_q = \Delta v^2/12 \quad (7-2)$$

由于信号是均匀分布的，所以可以得到信号功率为：

$$Q = \int_{-a}^a (m_k)^2 \left(\frac{1}{2a}\right) dm_k = M^2 \Delta v^2/12 \quad (7-3)$$

得到平均信号量化噪声比为：

$$S/N_q = M^2 \quad (7-4)$$

可以看出量化比特数对于信号量化信噪比的提升是很大的。

上述的量化方法是在对信号的量化过程中量化间隔是固定的，采用的是均匀量化方式。而有些信号可能会有着时变的强度，当信号强度小时，信号的量化信噪比也就变小了，所以，上述量化方法有着局限性。在面对有着时变的强度的信号是，可以采用非均匀量化的量化方式，该量化方式的量化间隔随着信号的大小不同而变化。当信号值小时，量化间隔  $\Delta v$  也小；当信号值大时，量化间隔  $\Delta v$  也增大。

在本文中，采用何种量化方式，多长的量化长度，小数部分与整数部分分别取多少长度需要对于算法中的运算与数据进行综合考量。对于相位补偿算法部分，分析相噪补偿部分的运算，该过程中，主要的运算变量有：输入信号，导频相噪信号（即用提取出的导频除以已知导频得到），相噪补偿信号（用导频相噪信号分段线性插值得到），输出信号（用相噪补偿信号乘输入信号中的荷载信号）。

上述信号均是复数信号，观察输入信号的星座图可知，在进行了色散补偿后，输入相噪补偿模块的信号的实部与虚部基本都分布在  $[-4, 4]$  范围内。因此，对于整数部分考虑分配

量化长度为 3，最高比特位作为符号位，正好对应[-4,4]范围。

同时，对于导频相噪信号与相噪补偿信号进行分析，它们是集中在复平面(1,0)附近的复数信号，且模为 1。因此可以认为它们的信息主要集中在虚部。图 7.1 对于它们的虚部信号进行了统计，它们的最小值在 0.007 左右，导频相噪信号与相噪补偿信号中主要包含着信道中的相位噪声信息，为了量化中尽量少地损失信息，考虑将小数部分量化长度设置为 7 位，此时量化间隔为  $1/2^7=0.0078125$ 。

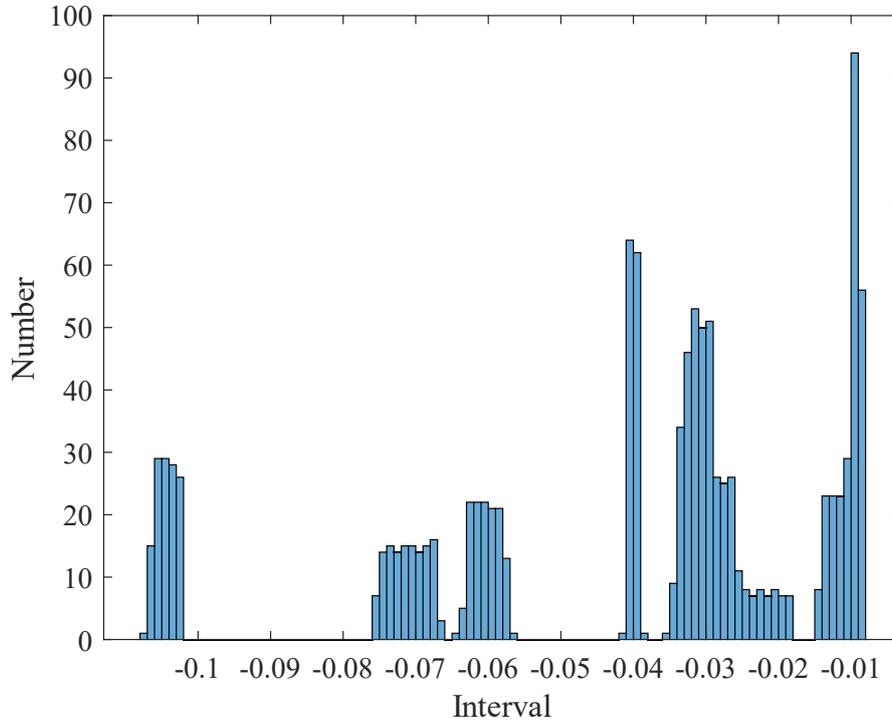


图 7.1 相噪信号虚部统计直方图

输出信号是星座图点与相位补偿信号的叠加，因此它不含有额外的信息，采用 7 位小数的方案对其进行量化不会产生额外的信息损失。

### 7.3 模型求解

本任务求解需要从资源与性能两方面进行分析。对于一个算法消耗资源的分析往往会需要借助相关 EDA 工具进行 FPGA 电路设计后进行综合得到，并不是本题的考察范围，因此只能根据题意进行定性分析。对于性能的分析可以由算法仿真得到。

本文对于定点量化的算法模型进行了仿真验证，如图 7.2，图中除了无定点量化外其他量化模型都是使用的 3 bit 整数量化，区别在于小数部分的量化长度。可以看到 7 bit 小数量化模型的误码率 (BER) 性能与无量化模型基本相同，图上的曲线完全重合在了一起，而 4 bit 量化模型在高信噪比 (SNR) 时误码率会有少量提升。由此可以看出，7 bit 小数量化在性能上胜过 4bit 模型，但是在资源消耗上 4 bit 小数模型的运算消耗的资源相比 7 bit 小数模型会大大降低，因此还需要继续进行分析。

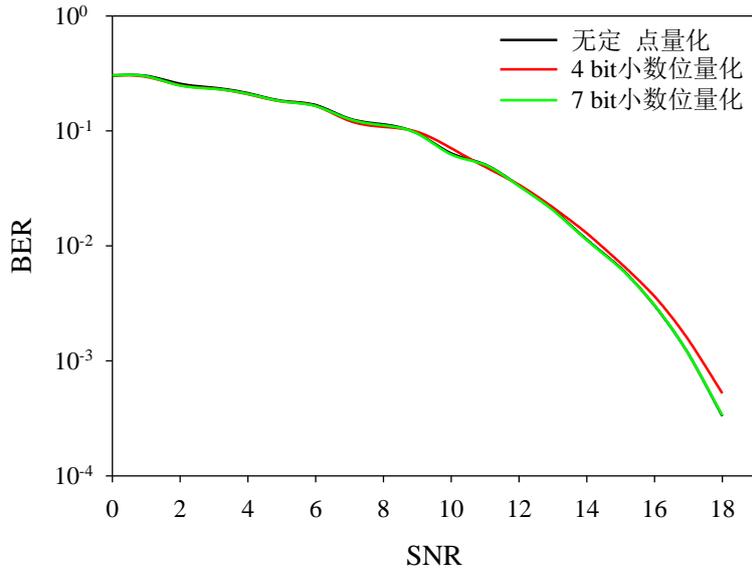


图 7.2 定点量化下的 BER 与 SNR 的关系

由于本算法采用将导频在窗口首位插入的方式，同时由于本文的线性相位假设，插入多于两个的导频对于性能提升影响很小，因此算法的 Pilot 开销只与窗口长度  $l$  相关。题中要求净荷流量大于 145 G baud，因此可以计算出对应的窗口长度要求为  $l > 60$ 。因此只需要窗口长度大于 60 即可满足题目要求。

本文对于 4 bit 小数位模型进行了 10 kHz~10 MHz 线宽下的算法仿真，得到了不同线宽下，BER 与窗口长度的关系，如图 7.3。可以看到，在线宽较小时，BER 与窗口长度呈反相关，即窗口越大，BER 越低，也就是性能越好；而当线宽较大时，BER 与窗口长度呈正相关。而这两种情况分界的线宽值相比第二问中图 6.4 对于无量化模型的仿真似乎变小了，猜测这是因为量化模型使得噪声中叠加上了量化噪声，因此相位噪声增加，等价于线宽增加了。

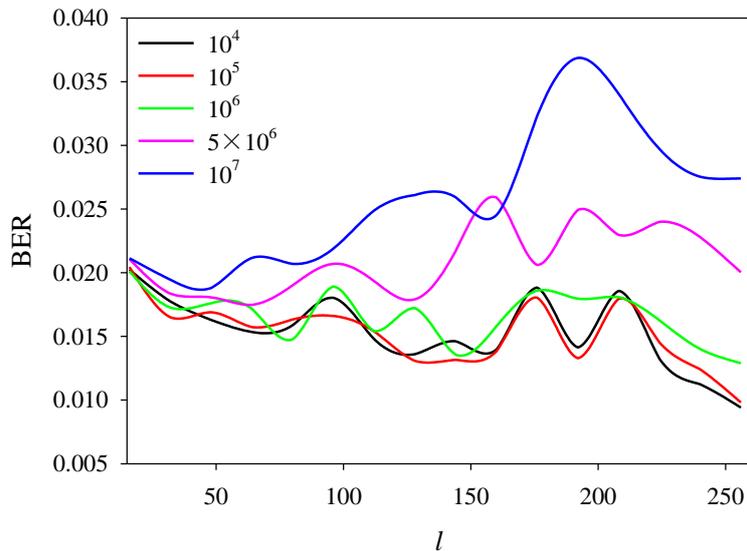


图 7.3 不同线宽下 BER 与窗口长度的关系 (SNR=14)

本文还对于不同线宽下，RSNR 代价与窗口长度的关系，如图 7.4。从图中观察到，在线宽较小时，不论采用哪种窗口长度，总是能够满足 RSNR 代价  $< 0.3$  dB；而在线宽足够大时，RSNR 代价开始与窗口长度正相关，但是仍然有能够满足 RSNR 代价  $< 0.3$  dB 的窗口长度。

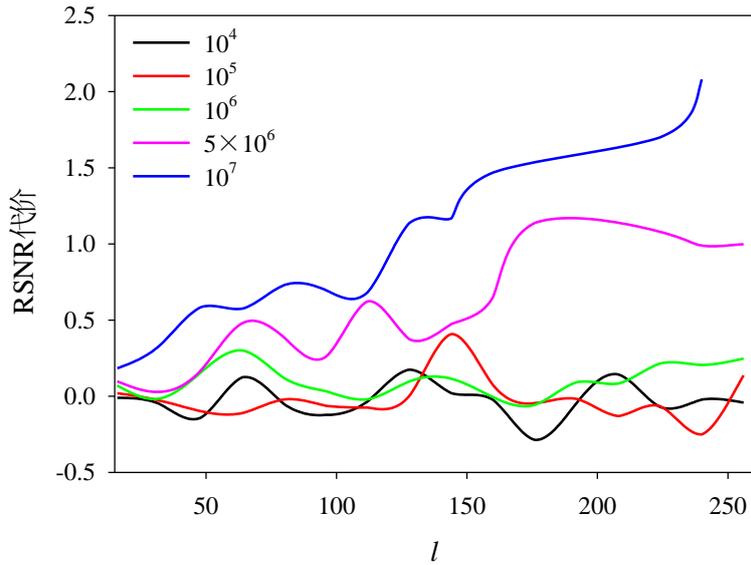


图 7.4 RSNR 代价与窗口长度的关系

由于在任务二中对于色散与 Pilot 的开销进行了论证，并得出色散对于 Pilot 开销没有影响的结论，因此此处不予讨论。

因此，4 bit 小数位量化模型能够在任务二的场景下达到 RSNR 代价 $<0.3$  dB 的目标。并在高线宽的情况下有一个 Pilot 开销最小值。而 7 bit 小数位量化模型有着比 4 bit 量化模型更优的性能，因此也能够满足任务二场景下的要求。

对于不同模型的资源消耗进行讨论。加法器的资源消耗与输入位宽成正比，乘法器的资源消耗与输入位宽的积成正比，查表 LUT 的资源消耗与 2 的输入位宽幂次方成正比。因此使用 3 位整数 4 位小数的模型相比 3 位整数 7 位小数的模型会节约 0.3 倍的加法器资源，1 倍的乘法器资源，以及海量的查找表 LUT 资源。

综上，4 bit 小数部分量化模型与 7 bit 小数部分量化模型都能够满足任务的要求，而 4 bit 小数部分量化模型能够节约大量的芯片资源。

#### 7.4 ASIC 算法设计（硬件设计）

在量化模型的基础上，对于对应的硬件电路设计还有一些可以改进的地方。

ASIC 上由于是二进制定点数表示，乘以或除以 2 的幂次方相当于对小数点进行移位，不会带来额外的资源代价。因此可以在某些乘法和除法操作上的参数进行设计，使得其消耗的资源大大减少。因此本文对于任务一中设计的电路模块进行了优化。令导频对应的星座点为  $2+2i$ 。可以得到导频相噪提取模块的运算变为：

$$\frac{x+iy}{2+2i} = \frac{(x+iy)(2-2i)}{(2+2i)(2-2i)} = \frac{x+y+i(y-x)}{4} \quad (7-5)$$

其中的除法运算分母由 6 变为了 4，这样就可以从乘法运算变为移位运算。对应的电路图如图 7.5 所示：

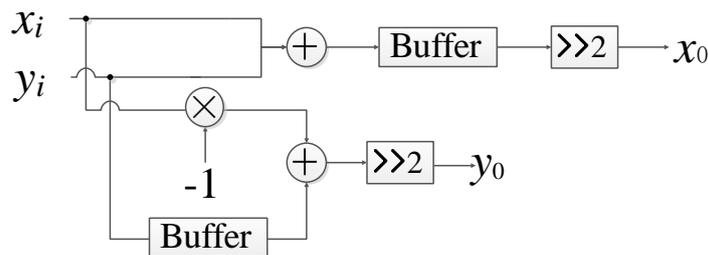


图 7.5 改进导频相噪提取模块电路结构

同理，载荷相噪求解模块中的除法运算也可以变为移位运算，其电路结构如图 7.6 所示：

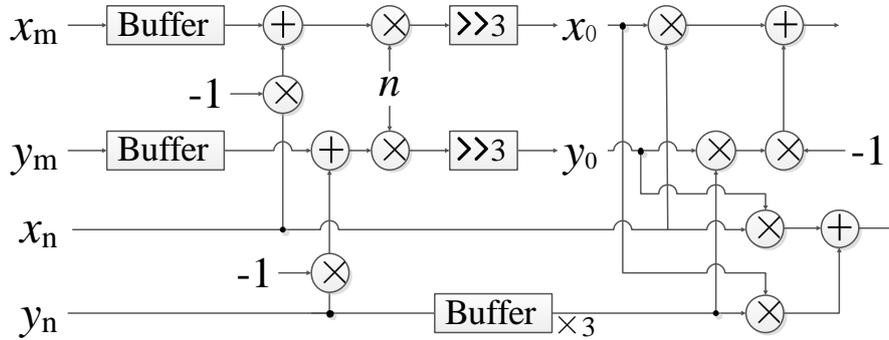


图 7.6 改进载荷相噪求解单元模块

FFT 进行蝶形运算将信号转换到频域，该模块需要进行  $N \log_2 N / 2$  次复数乘法，以及  $N \log_2 N$  次复数加法。而一次复数乘法等价于四次实数乘法与两次实数加法，一次复数加法等价于两次复数加法，因此实际等价于该模块需要进行  $2N \log_2 N$  次实数乘法与  $3N \log_2 N$  次实数加法。由此本文对于在窗口长度为 64，量化比特数为 7bit 时的 CR 算法进行了资源消耗情况统计如表 7-1 所示：

表 7-1 资源消耗统计

	乘法器	加法器	缓存 buffer	LUT	模块数量
导频相噪提取	1	2	2	0	4*2
频移计算	10	3	3	3	1*2
色散去耦合	4	2	0	1	64*2
载荷相噪求解	11	4	2	0	62*2
载荷相噪补偿	5	2	2	0	62*2
乘法器阵列	64	0	0	0	2
FFT	768	1152	0	0	2
IFFT	768	1152	0	0	2
总计	5724	5204	518	134	

根据给出的不同运算对应的资源消耗表则可以得到消耗的芯片资源统计。

表 7-2 不同运算对应的资源消耗表

基本操作	8+8 bit 加法器	8*8 bit 乘法器	8 bit-8 bit 查表	8 bit, 每 2048 符号延时
资源	1 U	8 U	128 U	1 U

同时还知道，加法器的资源消耗与输入位宽成正比，乘法器的资源消耗与输入位宽的积成正比，查表 LUT 的资源消耗与 2 的输入位宽幂次方成正比。因此 7 bit 加法器消耗的资源为  $7/8 U$ ，7 bit 乘法器消耗的资源为  $49/64 U$ ，7 bit-7 bit 查找表消耗的资源为  $56 U$ ，7 bit 延时 2048 符号消耗资源为  $7/8 U$ 。因此 7 bit 量化的模型消耗的资源为：

$$C=5724*49/64+5204*7/8+518/2048*7/8+134*56=16440 U \quad (7-6)$$

## 8. 任务四：自动优化位宽和实现性设计方案的制定

### 8.1 问题分析

位宽优化主要包括信号范围分析和精度分析两个步骤<sup>[7]</sup>：

Step 1 范围分析：用于确定系统中信号的整数位宽，以防止定点信号发生上溢；

Step 2 精度分析：用于寻找最优的小数位宽组合，除了防止信号出现下溢外，目标是在满足系统输出精度需求的情况下最小化系统的实现代价。

位宽优化的方法可以分为两大类，如图 8.1 所示：

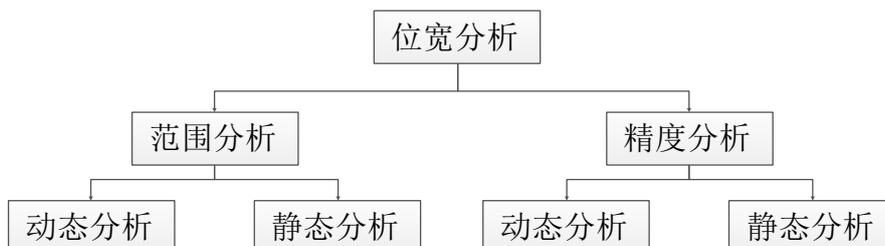


图 8.1 位宽优化的方法分类

1. 动态测试法：通过重复生成大量输入信号进行蒙特卡罗仿真，统计系统中信号的变化范围和达到输出性能要求所需要的精度。该方法的优点是可以相对真实地反映系统实际工作中信号的变化范围和有效精度，但缺点是运算时间通常较长，并且其结果无法保证覆盖到所有情况；

2. 静态测试法：通过分析系统中的数据流，计算信号的区间范围变化和量化噪声传播，在分析基础上，寻找优化的小数位宽组合以满足系统输出精度需求。由于位宽优化是一个 NP-hard 问题，除了极少数简单系统外，对信号的所有小数位宽组合可能性进行详尽搜索都难以实现，目前最好的方法都是借助各种优化算法来进行小数位宽优化。

### 8.2 模型建立与求解

为综合考虑性能与资源的影响，本文以不同位宽向量  $\vec{b}=(b_1, b_2, \dots, b_n)$  下的 RSNR 代价与资源使用情况量化函数分别赋予权重因子  $(w_1, w_2)$  组成一个目标函数，优化目标问题可表示如式 8-1 所示：

$$\min z = w_1 \cdot \text{RSNR 代价} + w_2 \cdot \text{Cost}(\vec{b}) \quad (8-1)$$

$$s. t. \{ \text{RSNR 代价} < 2 \times 10^{-2}$$

其中资源使用情况量化函数  $\text{Cost}$  与给定位宽向量下的芯片实现面积、功耗有关，本文简化为与位宽向量  $\vec{b}$  的 1-范数呈线性关系，如式 8-2 所示。

$$\text{Cost}(\vec{b}) = k \|\vec{b}\|_1 \quad (8-2)$$

目标问题为在满足 RSNR 代价要求的情况下，搜索最优化位宽组合，以使目标综合评价函数达到最优。

位宽优化问题可通过基于禁忌搜索算法的位宽自动优化方法以得到性能和代价综合评价目标函数的最优 Pareto 解集<sup>[8]</sup>。通过不断改变位宽组合，以使目标函数朝最优值前进。禁忌搜索算法寻找最优位宽组合的主要步骤如下：

Step 1 设定初始值，基于仿真系统的数据流信号区间变化情况，使用贪婪算法结果产生初始解；

Step 2 判断迭代次数是否达到限制，如果达到限制则输出当前最优解并终止流程；

Step 3 产生当前解的所有邻域解，可采用插入移动或交换移动的方法，构建当前解的所有邻域解；

Step 4 计算当前解所有邻域解对应的适配值，记录目标函数最小且满足 RSNR 代价的邻域解。如果该最小的邻域解对应的目标函数值  $z_l$  小于当前最优解对应的目标函数值  $z_{cur}$ ，则令  $z_{cur} \leftarrow z_l$ ，并将两个解的位置顺序交换，作为最早进入禁忌表的禁忌对象，转至 Step 6；

Step 5 判断邻域解对应的各对象的禁忌属性，在选择非禁忌对象中选取使得最小的邻域解，并与之前的解交换位置顺序，作为最早进入禁忌表的禁忌对象；

Step 6 迭代次数  $\leftarrow$  迭代次数 + 1，转至 Step 2。

在经过一定迭代次数后，可得到使目标函数最优的整数位宽和小数位宽的组合。

为验证算法的有效性，并简化任务求解计算量，本文将相噪补偿中涉及到的定点数位宽统一为相同的整数位宽和小数位宽，从而在 SNR=14 的条件下得到如图 8.2 所示的 BER 与整数位宽和小数位宽的关系，此时满足 BER 要求和 RSNR 代价要求的最小位宽组合为 (3,3)。

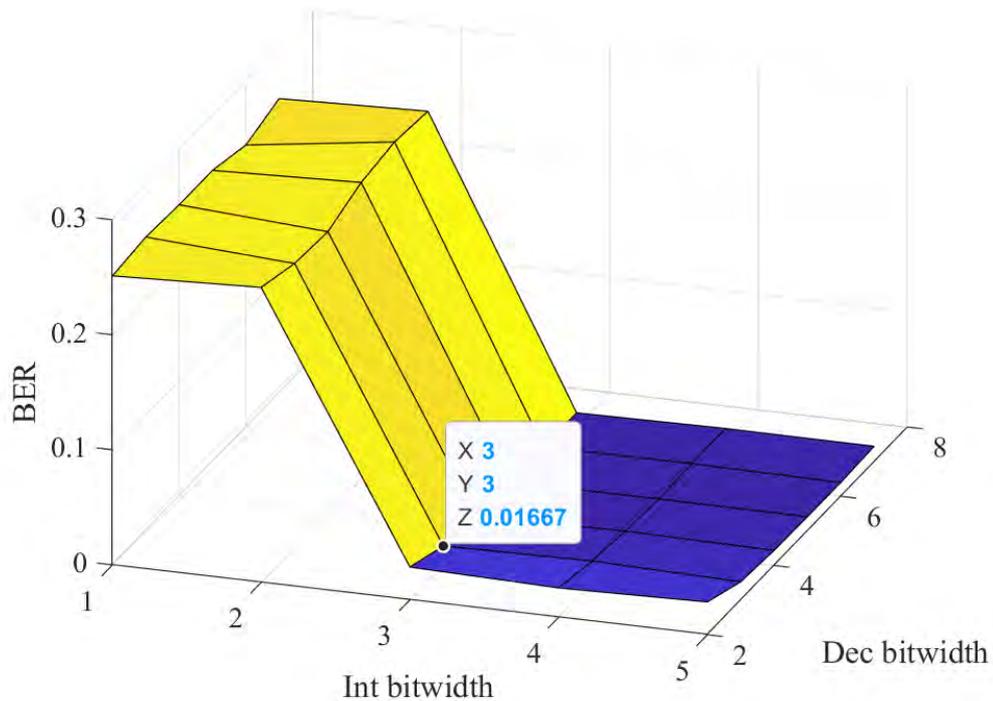


图 8.2 BER 与整数位宽和小数位宽的关系

## 9. 模型检验与评价

### 9.1 模型检验

本文对相位噪声提取效果进行了验证，如图 9.1 所示，通过仿真实验进行了相位噪声提取，如图是本文得到的仿真结果，蓝色折线是仿真中在信道生成的导频处相位偏移，红色折线是仿真中在接收端进行了色散补偿处理与色散与相噪的去耦合补偿后提取出来的导频的相位偏移，可以看出本文提出的算法对于相位噪声的估计与真实噪声非常接近，在使用该相位噪声的估计值进行补偿后，能够基本将相位噪声消除。表明本文的算法成功地将色散的影响基本消除，并从原信号中提取出了对应的相位噪声。

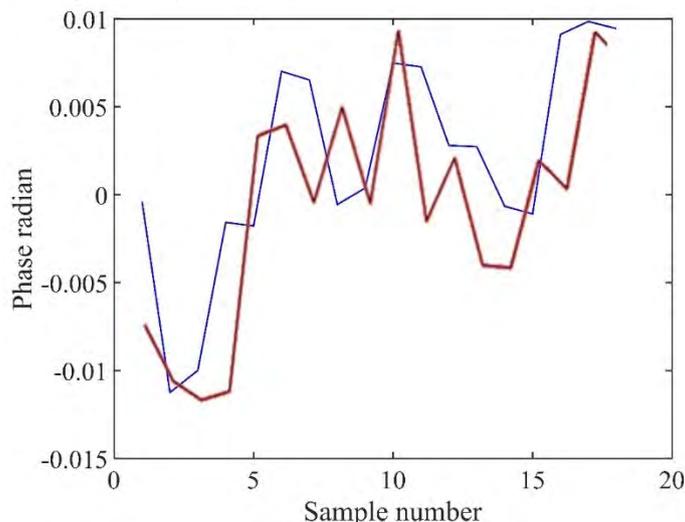


图 9.1 信道产生噪声与接收端提取噪声对比图

图 9.2 为本文仿真系统中调制后准备发送、增加白噪声后、解调前的的信号在星座图上的分布情况。可以看到信号在经过信道后信号在信道上叠加的噪声被本文设计的相噪补偿算法比较好地去除了。

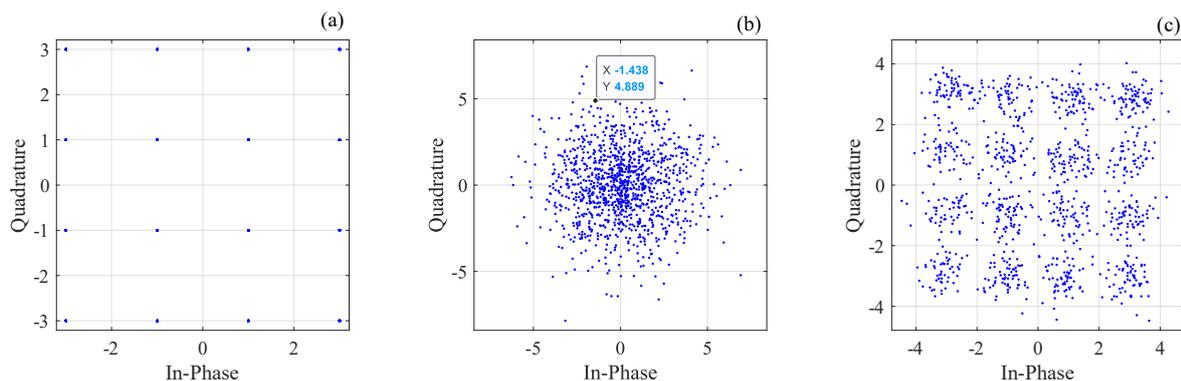


图 9.2 (a) 调制后准备发送 (b) 增加白噪声后 (c) 解调前的的信号在星座图上的分布情况

### 9.2 模型评价

#### 9.2.1 模型优点

本文提出的相噪补偿算法基于首尾导频插入与线性插值方法，相噪补偿效果较好；需要的硬件资源较少；对信号中的加性高斯白噪声也具有一定的补偿效果；且本文完成了硬件设计。

#### 9.2.2 模型缺点

本文提出的相噪补偿算法对于线宽较大的情况下补偿效果不是很好；没有考虑导频宽度变化对相噪补偿带来的影响。

## 参考文献

- [1] 张雄伟. DSP 芯片的原理与开发应用[M]. 北京: 电子工业出版社, 1997.
- [2] 姜俊迪, 林如俭. 一种 OFDM 时域导频插入的最小二乘估计方法[J]. 电子测量技术, 2008(03): 34-37.
- [3] Mastriani M J a P A. Quantum spectral analysis: frequency in time, with applications to signal and image processing[J], 2016.
- [4] 周颖, 程荫杭. 基于曲线拟合的地图匹配算法[J]. 交通运输系统工程与信息, 2004(02): 68-70.
- [5] 司守奎, 孙玺菁. 数学建模算法与应用[M]. 北京: 国防工业出版社, 2011.
- [6] 曹志刚, 钱亚生. 现代通信原理[M]. 北京: 清华大学出版社, 1992.
- [7] 张林生. 数字信号处理系统的定点化技术研究[D]. 哈尔滨: 哈尔滨工业大学, 2010.
- [8] Nguyen H N, Ménard D, Sentieys O. Novel Algorithms for Word-length Optimization[C]. European Signal Processing Conference, 2015.

## 附 录

### 附录1 主程序代码

```
function [Ber,pilot_num] = pilot_qam_enhanced_loop(noise,SNR,P_f_inter,P_f_length,xls_filename,N_fft,LW,is_quantize,qpath)
% clc
% clear
% noise=1;
% SNR=10:15;
% P_f_inter=8;
% P_f_length=1;
% xls_filename='test.xlsx';

%% 参数设置
% LW = 1e5; % 激光宽度
f_b = 150e9;
% N_fft=128;          % FFT 长度
M=16;                %16QAM调制
% SNR=0:1:24;        %仿真信噪比
% P_f_inter=6;       %导频间隔

%LLength = N_fft*16e3;
LLength = 256*16;
Ber=zeros(1,length(SNR));
Err_n = zeros(1,length(SNR));

MAX_ERR_N = 1000;

for jj=1:length(SNR)
    temp_error_n = 0;
    cycle_count = 0;
    while temp_error_n < MAX_ERR_N
        data_station=[];
        %% 基带数据数据产生
        P_data = randi([0 1],1, LLength);
        %% QAM调制
        data_temp1 = reshape(P_data,log2(M),[]);%以每组2比特进行分组, M=4
        data_temp2 = bi2de(data_temp1);          %二进制转化为十进制
        modu_data = qammod(data_temp2,M); % 16QAM调制

        %% 插入导频
        if 1
            [data,data_col,pilot_num,P_f_station,pilot_seq,data_station] = ins
```

```

ert_pilot_2(modu_data,N_fft,P_f_inter,P_f_length);
    else
        data_col = length(modu_data)/N_fft;
        pilot_num=0;
        %% 串并转换
        data=reshape(modu_data,N_fft,[]);
    end

    %% FFT-色散-IFFT
    dp_Dz=20000e-3;
    data_disp=dispersion(0,data,data_col,noise,dp_Dz,N_fft,f_b);

    %% 相位噪声
    data_pn = phase_noise(data_disp,noise,N_fft,LW,f_b);

    %% 并串转换
    Tx_data = reshape(data_pn,[],1);

    %% 信道
    rx_channel=awgn(Tx_data,SNR(jj),'measured');%添加高斯白噪声

    %% 串并转换
    Rx_data1=reshape(rx_channel,N_fft,[]);

    %% 色散补偿
    Rx_data1=dispersion(1,Rx_data1,data_col,noise,dp_Dz,N_fft,f_b);

    %% 信道估计与插值(均衡)
    if 1
        %% 信道校正
        data_afterreq = remove_pilot_2(Rx_data1,N_fft,P_f_station,pilot_se
q,data_station,P_f_length,f_b,dp_Dz,noise,is_quantize,qpath);
    else
        data_afterreq = Rx_data1;
    end
    %% 并串转换
    data_afterreq=reshape(data_afterreq,[],1);
    data_afterreq=data_afterreq(1:length(modu_data));
    %% QAM解调
    data_afterreq_rec = data_afterreq;
    demodulation_data=qamdemod(data_afterreq_rec,M);
    De_data1 = reshape(demodulation_data,[],1);
    De_data2 = de2bi(De_data1);
    %% 补零

```

```

if size(De_data2,2) < log2(M)
    De_data2(:,end+1) = 0;
end
De_Bit = reshape(De_data2',1,[]);
[number, ~] = biterr(De_Bit,P_data);%译码后的误码率
temp_error_n = temp_error_n + number;
cycle_count = cycle_count + 1;
% fprintf("SNR = %d dB, error_n = %d, ber = %.20f \n",SNR(jj),number,Ber(
r(jj)));
end
Ber(jj) = temp_error_n / (cycle_count * LLength);
Err_n(jj) = temp_error_n;
% fprintf("SNR=%d dB, cycle = %d, ber = %.10f\n",SNR(jj),cycle_count,Ber(j
j));
end

data_xls = [SNR' Ber' Err_n'];%把数据保存到data中, 其中a'的'表示转置
[m p]=size(data_xls);
data_cell=mat2cell(data_xls,ones(m,1),ones(p,1));%matrix转变成cell
title={'SNR','BER','Err_n'};%添加变量名称
result=[title;data_cell];%归纳变量名称和数据
s=xlswrite(xls_filename,result);%保存文件

end

```

## 附录2 RSNR 计算程序代码

```
clc;
clear;
rng(1);

SNR=12:0.5:17;
P_f_inter = 127;
P_f_len = 1;
N_fft=[16:16:256];
LW=1e7;
is_quantize=1; % 考虑定点量化
qpath=quantizer('fixed','round','saturate',[8 4]);

nfft_list = [];
len_list = [];
SNR_WO = [];
SNR_W = [];
col_list=[];
diff_list=[];

for ii=1:length(N_fft)
    for jj=1:length(P_f_len)
        [Ber_wo_noise,~] = pilot_qam_enhanced_loop(0,SNR,P_f_inter,P_f_len(jj),["SNR_BER_won_nfft_" N_fft(ii) "_len_" P_f_len(jj) ".xlsx"],N_fft(ii),LW,is_quantize,qpath);
        [Ber_w_noise,pilot_num] = pilot_qam_enhanced_loop(1,SNR,P_f_inter,P_f_len(jj),["SNR_BER_wn_nfft_" N_fft(ii) ".xlsx"],N_fft(ii),LW,is_quantize,qpath);
        if ~ issorted(Ber_wo_noise,'monotonic') || ~ issorted(Ber_w_noise,'monotonic')
            continue
        end
        snr_wo_n = interp1(Ber_wo_noise,SNR,2e-2,'linear','extrap');
        snr_w_n = interp1(Ber_w_noise,SNR,2e-2,'linear','extrap');
        nfft_list(end+1) = N_fft(ii);
        len_list(end+1) = P_f_len(jj);
        SNR_W(end+1) = snr_w_n;
        SNR_WO(end+1) = snr_wo_n;
        col_list(end+1) = pilot_num;
        diff_list(end+1) = snr_w_n-snr_wo_n;
        fprintf("n_fft = %d, len= %d, snr_wo_n = %.10f dB, snr_w_n = %.10f dB, diff = %.10fdB\n",N_fft(ii),P_f_len(jj),snr_wo_n,snr_w_n,snr_w_n-snr_wo_n);
    end
end
```

```
data = [nfft_list' len_list' SNR_W' SNR_WO' col_list' diff_list'];  
[m p]=size(data);  
data_cell=mat2cell(data,ones(m,1),ones(p,1));  
title={'n_fft','len','snr_w','snr_wo','p_n','diff'};  
result=[title;data_cell];  
s=xlswrite('nfft_rsnr.xlsx',result);
```

### 附录3 插入导频程序代码

```
function [data,data_col,pilot_num,P_f_station,pilot_seq,data_station] = insert_pilot_2(modu_data,N_fft,P_f_inter,P_f_length)
    %导频加到首尾
    P_f=3+3*1i; %Pilot frequency
    P_f_station = [];
    for img=1:P_f_length
        P_f_station = [P_f_station,img];
    end
    for img=N_fft-P_f_length+1:N_fft
        P_f_station = [P_f_station,img];
    end

    pilot_num=length(P_f_station);%导频数量

    data_station=[];
    for img = 1:N_fft %数据位置
        if ~ismember(img,P_f_station)
            data_station = [data_station,img];
        end
    end
    data_row=length(data_station);
    data_col=ceil(length(modu_data)/data_row);

    pilot_seq = ones(pilot_num,data_col)*P_f;    %将导频放入矩阵
    data = zeros(N_fft,data_col);%预设整个矩阵
    data(P_f_station(1:end),:)=pilot_seq;%对pilot_seq按行取

    if data_row*data_col>length(modu_data)
        data2=[modu_data;zeros(data_row*data_col-length(modu_data),1)];%将数据
        矩阵补齐
    else
        data2=modu_data;
    end;

    %% 串并转换
    data_seq=reshape(data2,data_row,data_col);
    data(data_station(1:end),:)=data_seq;%将导频与数据合并
end
```

#### 附录4 插入色散程序代码

```
function data_ifft_disp = dispersion(is_conj,data,data_col,noise,dp_Dz,N_ff
t,f_b,is_sec)
    %% FFT
    data_fft=[];
    for i=1:data_col
        data_fft(:,i) = fft(data(:,i));
    end

    %% 色散
    dp_lambda = 1550e-9;
    if noise ~= 1
        dp_Dz = 0;
    end
    dp_c = 3e8;
    data_fft_dispersion = [];

    for dp_f=1:N_fft
        real_dp_f = dp_f * f_b / N_fft;
        dispersion_coeff = dp_lambda^2*pi*dp_Dz*real_dp_f^2/dp_c;
        if is_conj == 1
            dispersion_coeff = -dispersion_coeff;
        end
        data_fft_dispersion(dp_f,:) = data_fft(dp_f,:) * exp(complex(0,dispersion_coeff));
    end

    %% IFFT
    data_ifft_disp=[];
    for i=1:data_col
        data_ifft_disp(:,i) = ifft(data_fft_dispersion(:,i));
    end
end
```

## 附录5 插入相噪程序代码

```
function [data_pn,noise_mat] = phase_noise(data,noise,N_fft,LW,f_b)
    if noise == 1
        sigma = sqrt(2*pi*LW/f_b);
        [m,n]=size(data); % m=N_fft
        noise_mat(1) = complex(0,normrnd(0,sigma));
        for jj=2:m*n
            noise_mat(jj) = noise_mat(jj-1) + complex(0,normrnd(0,sigma));
        end
        noise_mat = reshape(noise_mat,m,n);
        data_pn = data .* exp(noise_mat);
    else
        data_pn = data;
    end
end
```

## 附录6 相噪补偿程序代码

```
function [data_afterreq,h,H] = remove_pilot_2(Rx_data1,N_fft,P_f_station,pilot_seq,data_station,P_f_length,f_b,dp_Dz,noise,is_quantize,qpath)
    Rx_pilot=Rx_data1(P_f_station(1:end),:); %接收到的导频
    if is_quantize == 1
        Rx_pilot = quantize(qpath,Rx_pilot); % 定点量化
    end

    h=Rx_pilot./pilot_seq;

    if is_quantize == 1
        h = quantize(qpath,h); % 定点量化
    end

    if noise == 1
        omega_diff = angle(h(end,:) - h(1,:)) / N_fft;
        if is_quantize == 1
            omega_diff = quantize(qpath,omega_diff); % 定点量化
        end

        %% FFT
        data_fft=[];
        [~,data_col] = size(Rx_data1);
        for i=1:data_col
            data_fft(:,i) = fft(Rx_data1(:,i));
        end

        if is_quantize == 1
            data_fft = quantize(qpath,data_fft); % 定点量化
        end

        %% 色散
        dp_lambda = 1550e-9;
        dp_c = 3e8;
        data_fft_dispersion = [];

        for dp_f=1:N_fft
            real_dp_f = dp_f * f_b / N_fft;
            if is_quantize == 1
                real_dp_f = quantize(qpath,real_dp_f); % 定点量化
            end

            real_real_dp_f = real_dp_f^2 - (real_dp_f-omega_diff).^2;
```

```

    if is_quantize == 1
        real_real_dp_f = quantize(qpath,real_real_dp_f); % 定点量化
    end

    dispersion_coeff = dp_lambda^2*pi*dp_Dz*real_real_dp_f/dp_c;
    if is_quantize == 1
        dispersion_coeff = quantize(qpath,dispersion_coeff); % 定点量化
    end

    data_fft_dispersion(dp_f,:) = data_fft(dp_f,:) .* exp(complex(0,dispersion_coeff));
    if is_quantize == 1
        data_fft_dispersion = quantize(qpath,data_fft_dispersion); % 定点量化
    end
end
end

%% IFFT
data_ifft_disp=[];
for i=1:data_col
    data_ifft_disp(:,i) = ifft(data_fft_dispersion(:,i));
end
if is_quantize == 1
    data_ifft_disp = quantize(qpath,data_ifft_disp); % 定点量化
end
Rx_data1 = data_ifft_disp;
end

H=interp1(P_f_station(1:end)',h,data_station(1:end)', 'linear', 'extrap');
if is_quantize == 1
    H = quantize(qpath,H); % 定点量化
end
H=H./sqrt(real(H).*real(H) + imag(H).*imag(H));
if is_quantize == 1
    H = quantize(qpath,H); % 定点量化
end

%分段线性插值
%% 信道校正
data_afterreq=Rx_data1(data_station(1:end),:)./H;
if is_quantize == 1
    data_afterreq = quantize(qpath,data_afterreq); % 定点量化
end
end
end

```

