连续相位调制信号的非相干序列检测

研究报告

目录

1.问题背景

连续相位调制(Continuous phase modulation)是一种重要的数字通信调制方式.其数学形式可以表达为[1]:

$$s(t,\alpha)=\sqrt{\frac{2E_S}{T}}\exp\left(j2\pi h\sum_n a_n q(t-nT)\right)\quad(1)$$

上式中E_S表示每个符号的能量,T表示符号传输周期.h为调制率,能够表达为有理分数的形式$h=k/p$,其目的是为了让解码过程中的状态数有限[2]. a_n表示传输的信息符号.对于M元符号, $\{a_n\}$在$\{\pm1,\pm3\ldots\pm(M-1)\}$中取值.q(t)称为CPM调制的频率脉冲(frequncy pulse).其导数g(t)是在$(0,L_gT)$区间上取值的信号.通常的q(t)满足：

$$q(t)=\begin{cases}0 & (t<0)\\ \int_0^t g(\tau)d\tau & (0<t<L_gT)\\ \frac{1}{2} & (t>L_gT)\end{cases}\quad(2)$$

在本实验中使用到了GMSK与2RC两种脉冲成型方式,其频率脉冲的导数g(t)分别如下[2]:

$$\text{GMSK:}\quad g(t)=\frac{1}{2T}\left\{Q\left(\left[2\pi B\frac{t-\frac{T}{2}}{\sqrt{\ln 2}}\right]\right)-Q\left(\left[2\pi B\frac{t+\frac{T}{2}}{\sqrt{\ln 2}}\right]\right)\right\}\quad(3)$$

$$\text{2RC:}\quad g(t)=\begin{cases}\frac{1}{4T}\left[t-\cos\left(\frac{\pi t}{T}\right)\right] & (0\le t\le 2T)\\ 0 & \text{else}\end{cases}\quad(4)$$

由(1)式可以看出,CPM信号的幅值始终恒定为$\sqrt{2E_S/T}$,其信息由信号相位承载.由此衍生出基于锁相环(phase − locked loop)的相干检测方法.但是该方法存在有诸多的缺点,如错锁(false − locks),相位滑动(phase − slips)等等.

针对此缺点引入了非相干的序列检测方法,即本文所讲的非相干序列检测方法.该方法是基于对CPM信号进行洛朗分解(Laurent decompostion),将原信号等价转化为一系列线性调制信号的线性组合,然后根据使用维特比算法(VA)对原信号进行译码操作,但该算法性能远不及相干检测方法[1]. 在此基础上,我们在进行VA译码之前先将接收到的信号通过白化匹配滤波器,然后再执行VA算法.经过实验验证发现通过增加观察序列的长度以及接收机的复杂度(包括WMF以及VA两个部分)可以将算法的性能不断往相干检测逼近[1].下面将具体介绍仿真的细节.

2.工作思路与内容

本次仿真任务主要可以分成四个模块(1)脉冲成型(2)Laurent分解(3)白化匹配滤波(WMF)(4)维特比译码模块(VA)。实验框图如下：

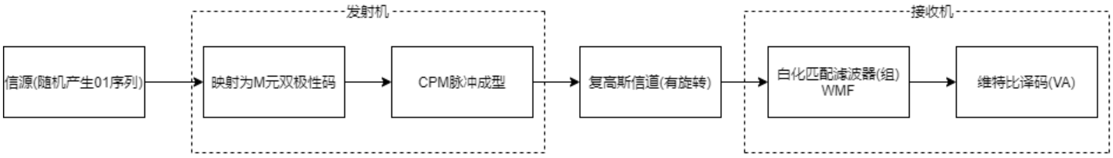

图 1 实验框图

在实际的实验当中，将信源部分和符号映射部分合二为一，即直接将M元双极性码作为传输的信息。因此经过维特比译码后得到的是系统的误符号率P_s。根据误符号率和误比特率的近似关系$P_b \approx P_s/\log_2 M$[3]得到系统的误比特率。

在文章原文以及各篇参考文献中，信号的表达形式基本上均为连续形式，不容易实现计算机仿真。因此在实验过程中一个很重要的部分便是连续时间信号的离散化。即利用按照一定的过采样率sps采集得到离散信号来近似原文中的连续时间信号。为了实现采样重建精度和计算的复杂度的折衷，在本实验中过采样率$sps = 10$。

下面将详细讲述上述各个模块的设计与实现思路。

2.1 CPM脉冲成型

由式(1)可知,CPM信号波形由参数$T, E_s, h, q(t)$决定.在本次实验中,对两组不同的CPM信号进行仿真,两组实验的参数如下:

实验 1: GMSK针对二元CPM调制.

$h = 0.5, L_g = 2, BT = 0.25, T = 1$

实验 2:2RC针对四元CPM调制

$h = 0.25, L_g = 2, T = 1$

在仿真过程中,由于不能直接针对连续信号进行仿真,因此需要经过过采样后的离散信号对连续信号波形进行近似.在实验 1,2 中过采样率$sps = 10$,即对每个符号共采样 10 个点进行近似.下面将详细讲述仿真中实验 1、2 的脉冲成型过程.

2.1.1 GMSK成型

由(1)式,GMSK信号中$g(t)$可以改写为:

$$g(t) = 0.5\int_{t_2}^{t_1} \exp\left(-\frac{\tau^2}{2}\right) d\tau = 0.5\int_{-0.5}^{0.5} \frac{1}{\sqrt{2\pi}\delta} \exp\left(-\frac{(t-\tau)^2}{2\delta^2}\right) d\tau \quad (5)$$

其中$t_1 = 2\pi B(t + \frac{T}{2})/\sqrt{\ln 2}$, $t_2 = 2\pi B(t - \frac{T}{2})/\sqrt{\ln 2}$, $\delta = \frac{\sqrt{\ln 2}}{2\pi BT}$

在仿真过程中为提高实验精度，使用SciPy中的 quad 函数计算$g(t)$的值。由于$g(t)$没有确定的解析表达式,因此为求得$q(t)$只有采取用求和代替积分的方法进行,具体如下：

$$q[n] = \int_0^{\frac{nT}{sps}} g(\tau)d\tau = q[n-1] + \frac{1}{sps} g[n] \quad (6)$$

$$q[0]=0 \quad (6)$$

由此即可得到频率脉冲序列。然后按照(1)式即可获得调制后的CPM信号s(t)

2.1.2 2RC成型

由(2)和(4)式，可得$q(t)=\frac{1}{4T}(t-\frac{T}{\pi}\sin(\frac{\pi t}{T}))$，之后直接按照过采样率$sps=10$即可得到频率脉冲序列$q[n]=q\left(\frac{nT}{sps}\right)=\frac{1}{4}\left(\frac{n}{sps}-\sin\left(\frac{\pi n}{sps}\right)\right)$ $(0\le n\le 2sps-1)$。由此即可得到频率脉冲序列。然后按照(1)式即可获得调制后的CPM信号s(t)

2.2 Laurant分解

Laurant分解即是将CPM信号的形式从(1)转换为以下形式[1]：

$$s(t,\mathbf{a})=\sum_{k=0}^{Q^{\log_2 M}(M-1)-1}\sum_{n}\alpha_{k,n}h_k(t-nT) \quad (7)$$

其中$Q=2^{L_g}-1$。$\alpha_{k,n}$称为伪符号(pseudo symbol)，由信息符号序列$\{a_n\}$以及调制参数决定。h_k为线性调制序列，由频率脉冲q(t)以及调制参数决定。可以在数学上严格证明(7)与(1)等价。

在实际工程中，通常不会取$Q^{\log_2 M}(M-1)-1$个h_k来无误的表示原信号，而是取前$K=M-1$个信号来表示。这样的截断能够有效的降低计算的复杂度同时保证误差足够的小[1], 称前$M-1$个信号为主脉冲(princpal pulse)。

下面将简要讲述如何得到(7)中的$\alpha_{k,n}$以及h_k，详细过程参见参考文献[4]

针对二元信号首先需要求得

$$u(t)=\begin{cases}\dfrac{\sin\big(2h\pi q(t)\big)}{\sin(h\pi)} & 0\le t\le LT\\ u(2LT-t) & LT\le t\le 2LT\\ 0 & \text{else}\end{cases} \quad (8)$$

之后获得二进制表示序列$\beta_{k,i}$：$k=\sum_{i=1}^{L-1}2^{i-1}\beta_{k,i}$，由此得到线性调制信号

$$h_k(t)=\prod_{i=0}^{L-1}u\big(t+iT+\beta_{k,i}LT\big) \quad 0\le k\le Q-1 \quad (9)$$

其中h_k的持续时间由$D_k=\min_i\{L\big(2-\beta_{k,i}\big)-i\}$, $0\le i\le L-1$确定。

伪符号序列：

$$\alpha_{k,n}=\exp\left\{jh\pi\left[\sum_{m=-\infty}^{n}a_n-\sum_{i=0}^{L-1}a_{n-i}\beta_{k,i}\right]\right\} \quad (10)$$

综上即可得到原来CPM信号的Laurant分解。

对于多元调制(即本实验的2RC调制)，其线性调制脉冲的计算过程相比二元调制要更为复杂。具体的计算流程详见参考文献[4]。下面将介绍在不同调制方式以及不同K取值的条件下Laurant分解的细节。

2.2.1 GMSK $K = 1$

在$K = 1$的条件下，$\beta_{0,i} \equiv 0$。因此线性调制信号可以化简为：

$$h_0(t) = \prod_{i=0}^{L-1} u(t + iT) \quad 0 \leq k \leq Q - 1 \quad (11)$$

伪符号表达式可以化简为：

$$\alpha_n = \exp\left\{j0.5\pi\left[\sum_{m=-\infty}^{n} a_n\right]\right\} \quad (12)$$

由此可以得到伪符号的递推关系式[5]：

$$\alpha_n = ja_n\alpha_{n-1}, \quad \begin{cases}\alpha_{2n} & \in \{\pm1\} \\ \alpha_{2n+1} & \in \{\pm j\}\end{cases} \quad (13)$$

2.2.2 GMSK $K = 2$

在该条件下，两个线性调制信号序列如下($sps = 10$)：

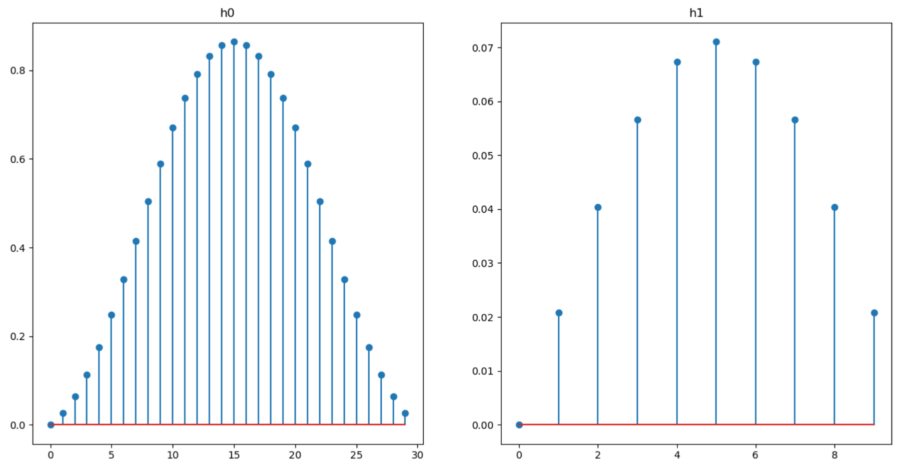

图 2 $h_0(t)$与$h_1(t)$对比图

由上图可知，Laurant分解的主要能量都集中于主脉冲$h_0(t)$中。且$h_0(t)$的持续时间为3T，$h_1(t)$的持续时间为T

在$K = 2$的条件下，伪符号序列为一个长度为sig_len的二维向量组。根据原文定义：$\boldsymbol{\alpha_n} = (\alpha_{0,n}, \alpha_{1,n})^T$其递推关系变为[5]

$$\alpha_{0,n} = ja_n\alpha_{0,n-1}, \quad \begin{cases}\alpha_{0,2n} & \in \{\pm1\} \\ \alpha_{0,2n+1} & \in \{\pm j\}\end{cases} \quad (14)$$

$$\alpha_{1,n} = ja_n\alpha_{0,n-2}, \quad \begin{cases}\alpha_{0,2n} & \in \{\pm j\} \\ \alpha_{0,2n+1} & \in \{\pm1\}\end{cases} \quad (15)$$

2.2.3 2RC K = 3

在该条件下主脉冲的计算过程为[4]

$$h_0(t) = c_0^{(0)}(t)c_0^{(1)}(t) \quad h_1(t) = c_0^{(0)}(t+T)c_0^{(1)}(t) \quad h_2(t) = c_0^{(0)}(t)c_0^{(1)}(t+T)$$

$$\alpha_{0,n} = b_{0,n}^0 b_{0,n}^1 \qquad \alpha_{1,n} = b_{0,n}^0 b_{0,n}^1 \qquad \alpha_{2,n} = b_{0,n}^0 b_{0,n}^1$$

其中$c_0^{(0)}(t)$, $c_0^{(1)}(t)$的计算与二元情况中$h_0(t)$类似，b_n^0, b_n^1的计算与二元情况中的α_n类似。

$$c_0^{(l)}(t) = \prod_{i=0}^{L-1} u^{(l)}(t+iT) \quad l = 0,1 \quad (16)$$

$$b_{0,n}^{(l)} = \exp\left\{jh^l\pi\left[\sum_{m=-\infty}^{n}\gamma_{m,l}\right]\right\} \quad l = 0,1 \quad (17)$$

在该条件下共有三个主脉冲princpal pulse，其时域图如下所示：

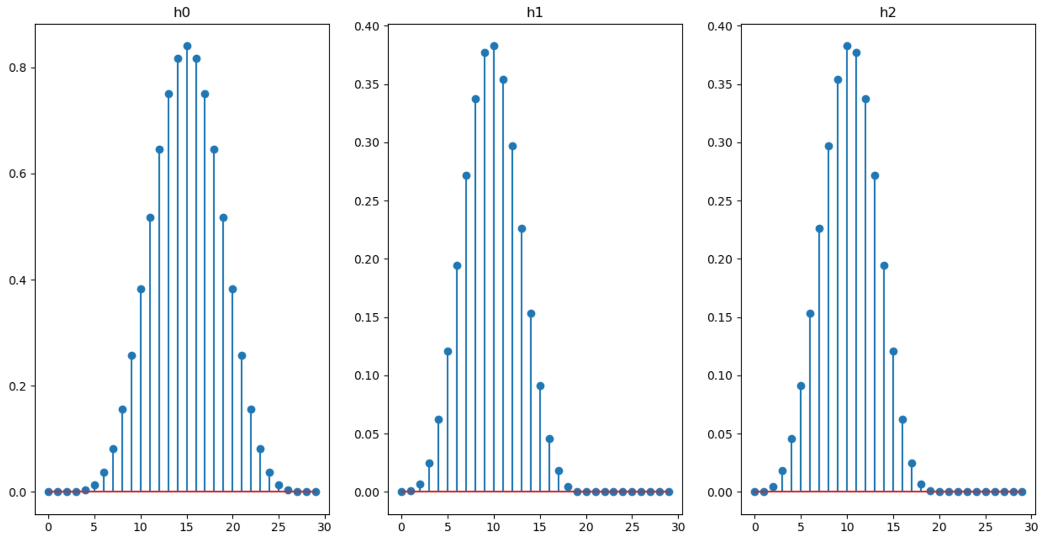

图 3 $h_0(t)$与$h_1(t)$, $h_2(t)$对比图

由上图可知，$h_0(t)$与$h_1(t)$, $h_2(t)$为princpal pulse，其持续时间分别为3T, 2T, 2T。同时从上图可以看出，$h_1(t)$与 $h_2(t)$的波形十分接近，因此在后续处理中可以将两者求平均进行计算。

2.2.4 2RC K = 5

在 2.2.3 的基础上增加了两个次要脉冲$h_3(t)$, $h_4(t)$以及伪符号 $\alpha_{3,n}$ $\alpha_{4,n}$，其计算公式如下：

$$h_3(t) = c_0^{(0)}(t+2T)c_0^{(1)}(t) \qquad h_4(t) = c_0^{(0)}(t)c_0^{(1)}(t+2T)$$

$$\alpha_{3,n} = b_{0,n-2}^0 b_{0,n}^1 \qquad \alpha_{0,n} = b_{0,n}^0 b_{0,n-2}^1$$

在该条件下五个线性调制信号时域图如下：

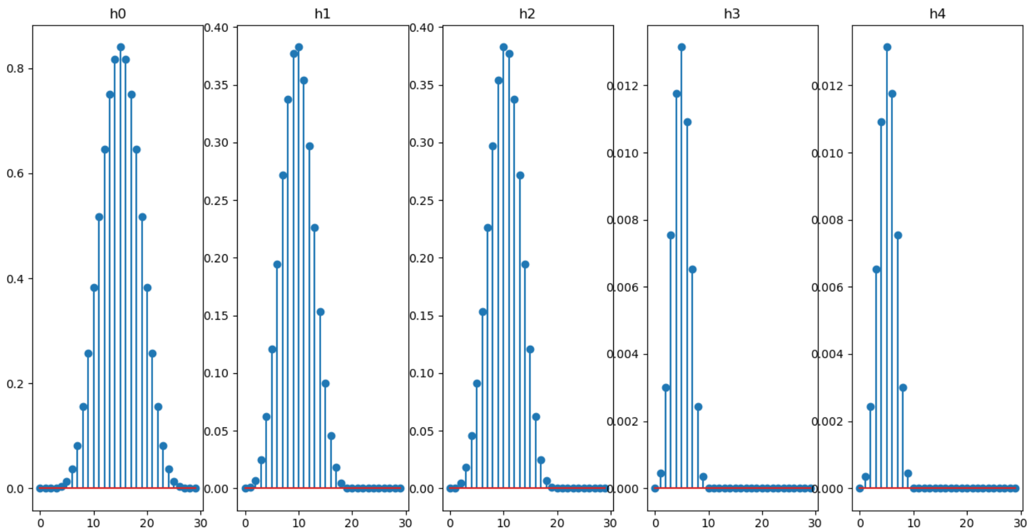

图 4 线性调制脉冲对比图

由上图可以进一步得到在四元2RC调制中线性调制信号的能量主要集中在$h_0(t)$与$h_1(t), h_2(t)$中。类似的由，上图可以看出在时域上$h_3(t), h_4(t)$十分相似，因此可以进行类似$h_1(t), h_2(t)$求平均的操作以简化接收机的复杂度。

2.3 白化匹配滤波器(WMF)

由原文献可知，信号s(t)经过信道后到达接收端的信号r(t)可以表示为：

$$r(t) = s(t,\boldsymbol{\alpha})e^{j\theta} + w(t) \quad (18)$$

其中$\theta \sim U(0,2\pi)$,w(t)为功率谱密度为n_0的高斯白噪声。在具体实现的过程中，w(t)为长度与$s(t,\boldsymbol{\alpha})$相等高斯分布序列，其方差为噪声的功率。因此$\sigma^2 = n_0 W$。根据Nyqusit定理，数字通信的带宽应该满足$\frac{R_s}{2} \le W \le R_s$。因此在本实验中取带宽$W = R_s = sps$，因此$\sigma^2 = n_0 sps$。

WMF由两部分组成，匹配滤波器以及白化滤波器。其结构如下：

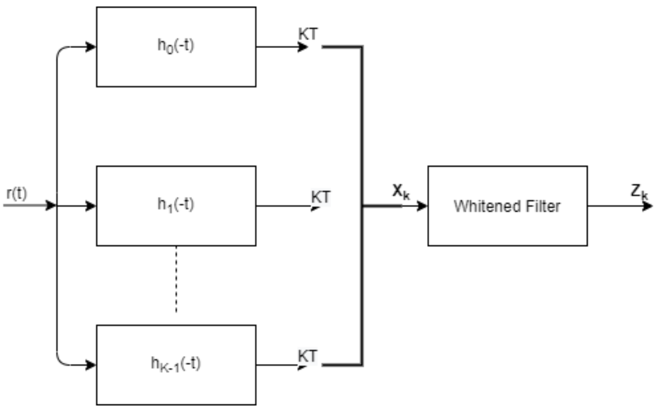

图 5 WMF结构图

将Laurant分解得到的脉冲分量$h_k(t)$反转即可得到匹配滤波器$h_k(-t)$。在具体实现中，通过numpy中的convovle函数实现匹配滤波。需要注意的是，在匹配滤波之后还需

要进行采样。令xt为匹配滤波后得到的序列。则送入到自化滤波模块的$\mathbf{x}$满足$\mathbf{x}[\mathrm{n}] = \mathrm{xt}[\mathrm{n} \cdot \mathrm{sps} + \mathrm{sps} - 1]$

白化滤波器WF是$\mathrm{K} \times \mathrm{K}$滤波器组，令其传递函数$\mathrm{H(z)}$。为实现白化功能，先考虑线性调制脉冲$\{h_k(t)\}$自相关的z变换$\mathrm{G(z)}$。$\mathrm{G(z)}$具有对称性，即$\mathrm{G(z^{-1})} = \mathrm{G^T(z)}$。经过匹配滤波后，噪声的功率谱密度变为$\Phi_n(z) = 2N_0G^T(z)$。考虑对$G^T(z)$进行离散谱分解，即找到$F(z)$使得$G^T(z) = F(z^{-1})F^T(z)$。具体的谱分解过程详见参考文献[6]。其核心思想在于递归的消去$G^T(z)$特征多项式零点，最终将$G^T(z)$转化为一个仅包含常数的对称矩阵。因此白化滤波器的传函即可表示为$F^{-1}(z^{-1})$。需要注意的是，为保证系统的稳定性，需要保证$\det(F(z^{-1}))$的根包含在单位圆内[1]。

在本次实验中，自相关计算中所用到的积分用求和来近似。下面将展示四次仿真中的$G^T(z)$以及分解得到的$F(z)$。

(1)GMSK $\mathbf{K=1}$

在该条件下能够严格求解白化滤波器。$G^T(z)$为一维，可以表示为$G(z) = \sum_{i=-2}^{i=2} g_i z^{-i}$，$g_i$为$h_0$的自相关。因此可以先求出$G(z)$的零点$z_i (i = 1:4)$。然后取在单位圆内的两个零点$z_0, z_1$，计算卷积$a = \mathrm{conv}([1, -z_0], [1, -z_1])$,系数$k = \mathrm{sum}(g_i)/((1 - z_0)(1 - z_1))^2$。则白化滤波器的传递函数分母即为$\sqrt{k}a$,分子为 1。

(2)GMSK $\mathbf{K=2}$

在$G^T(z)$大于一维的情况下不能严格求解白化滤波器，只有采用数值方法[7]求解，从而一定的误差。在本次实验中使用SymPy进行相关的符号计算。该条件下$G^T(z)$与分解得到的$F(z)$如下：

$$G^T(z)\text{:} \begin{bmatrix} 0.0356z^2 + 0.4483z + 0.882 + 0.4483z^{-1} + 0.0356z^{-2} & 0.01173 + 0.03656z^{-1} + 0.01173z^{-2} \\ 0.01173 + 0.03656z + 0.01173z^2 & 0.002465 \end{bmatrix}$$

$$F(z) = \begin{bmatrix} 0.08024 + 0.6558z^{-1} + 0.4442z^{-2} & 0.2305 + 0.45977z^{-1} \\ 0.0263 & 0.0421 \end{bmatrix}$$

(3)2RC $\mathbf{K=2}$

该条件下$G^T(z)$与分解得到的$F(z)$如下：

$$G^T(z) = \begin{bmatrix} 0.0825z + 0.503 + 0.0825z^{-1} & 0.1195z + 0.1195 + 0.0007z^{-1} \\ 0.0007z + 0.1195 + 0.1195z^{-1} & 0.0045 + 0.0829 + 0.0045z^{-1} \end{bmatrix}$$

$$F(z) = \begin{bmatrix} 0.11897 + 0.69347z^{-1} & -0.09355 \\ 0.17232 + 0.12821z^{-1} & -0.11299 + 0.15584z^{-1} \end{bmatrix}$$

(4)2RC $\mathbf{K=3}$

该条件下$G^T(z)$与分解得到的$F(z)$如下：

$$G^T(z)\text{:} \begin{bmatrix} 0.0852z + 0.503 + 0.0852z^{-1} & 0.1195z + 0.1195 + 0.0007z^{-1} & 1.73e-4 + 0.00406z \\ 0.0007z + 0.1195 + 0.1195z^{-1} & 0.0045 + 0.0829 + 0.0045z^{-1} & 6.9e-4 + 6.9e-4z \\ 1.73e-4 + 0.00406z^{-1} & 6.9e-4 + 6.9e-4z^{-1} & 5.44e-5 \end{bmatrix}$$

$$F(z)=\begin{bmatrix}0.1532+0.6105z^{-1} & -0.0302+0.2923z^{-1} & -0.1278995\\ 0.20435+0.0379z^{-1} & -0.019193+0.1778z^{-1} & -0.079034\\ 0.0053 & 0.0028 & 0.0043\end{bmatrix}$$

2.4 维特比算法(VA)

维特比算法是本次实验的一大核心，其算法中的状态体现在

$$y_n=\sum_{l=0}^{L-1}F_l^T\boldsymbol{\alpha}_{n-l}\quad(19)$$

以及使用的分支距离(branch metrics)

$$\lambda(a)=|\sum_{k=0}^{K-1}\sum_{i=0}^{N-1}z_{k,n-i}y_{k,n-i}^{*}|-|\sum_{k=0}^{K-1}\sum_{i=1}^{N-1}z_{k,n-i}y_{k,n-i}^{*}|-0.5\sum_{k=0}^{K-1}|y_{k,n}|^2\quad(20)$$

分支距离是基于最大似然估计对状态进行比较，在算法执行时应当选取累积分支距离最大的状态作为最优的估计。

其中$y_{k,n}$为基于当前状态假设下根据(19)式计算得到的结果，因此在不考虑状态减少(state reduced)的情况下，一个状态进行一次分支距离计算需要$\{y_n \dots y_{n-N+1}\}$参与计算。根据(19)式计算$\{y_n \dots y_{n-N+1}\}$需要$\{\boldsymbol{\alpha}_n \dots \boldsymbol{\alpha}_{n-N+2-L}\}$参与计算。由于伪符号$\boldsymbol{\alpha}_n$完全由信息符号所决定，因此对于M元信息符号其状态数为$S=M^{N+L-1}$。

当N与M较大时，状态数S呈指数级上升，因此需要引入状态化简来使系统计算复杂度降低。状态简化的详细过程详见参考文献[7]。通常的状态化简需要将状态进行分类，本实验为了简单起见本实验采用DFSE估计器，其计算流程为对于状态数S，每一个状态代表了$K'=\log_2(S)$个信息符号，用$\{\mathbf{a}_n \dots \mathbf{a}_{n-K'+1}\}$表示。由于分支距离的计算需要用到$\{\mathbf{a}_n \dots \mathbf{a}_{n-N+2-L}\}$，因此$\{\mathbf{a}_{n-K'} \dots \mathbf{a}_{n-N+2-L}\}$需要采用最大似然估计的方法得到。将估计得到的$\{\mathbf{a}_{n-K'} \dots \mathbf{a}_{n-N+2-L}\}$代入到(19)式中，计算得到分支距离。从而实现状态数减少同时保证算法的有效性。在本实验中的最大似然估计以以下方式进行，即对于每一步译码，每个状态保留在该步译码比较前到达该状态的路径所经过的各个状态对应的伪符号$\boldsymbol{\alpha}_n$，经过状态转移比较后将新的伪符号序列用迭代的方式传递给下一个状态，以实现最大似然估计。

维特比算法在执行过程中可以分为三个时期：初始时期，启动时期，稳定时期。初始时期从零状态出发，给转移的状态赋上累积距离初值。启动时期状态转移关系每一步均在变化，且不涉及到状态之间的比较。稳定时期状态转移关系不变，且状态转移涉及到状态间的比较。每一次状态转移之后需要记录当前状态时转移后状态在当前时刻的上一个状态，以便之后进行状态的回溯。

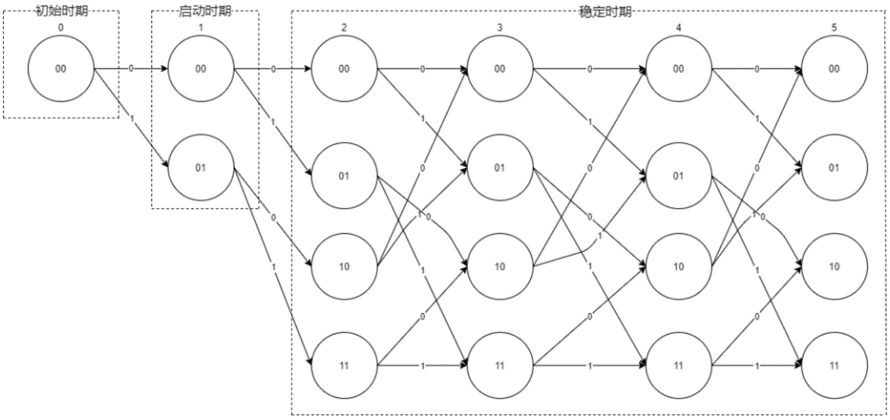

图 6 $M=2,S=4$维特比译码状态网格

在本次实验中，状态的编码采用进制编码的方式。以$S=4, M=2$为例，令编码bit0代表$\mathbf{a_n}=-1$，编码bit1代表$\mathbf{a_n}=1$。则状态$\{-1,-1\}$用$S=0$表示，状态$\{-1,1\}$用$S=1$表示，状态$\{1,-1\}$用$S=2$表示，状态$\{1,1\}$用$S=3$表示。使用进制编码能够很方便的通过移位、取余等运算实现状态间的转移。

3.仿真结果与分析

3.1 Fig2 仿真结果

根据上述实验实际，针对GMSK调制，$h=0.5$，$BT=0.25$，调制深度$L=2$的情况下误比特率随着信噪比E_b/n_0的变化关系如下：

3.1.1 $K=1$条件下

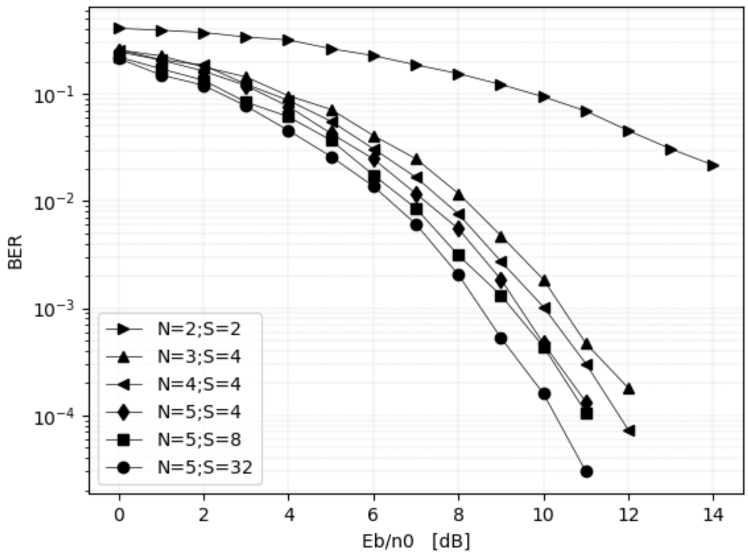

图 7 Fig2($K=1$)仿真结果

3.1.2 $K=2$条件下

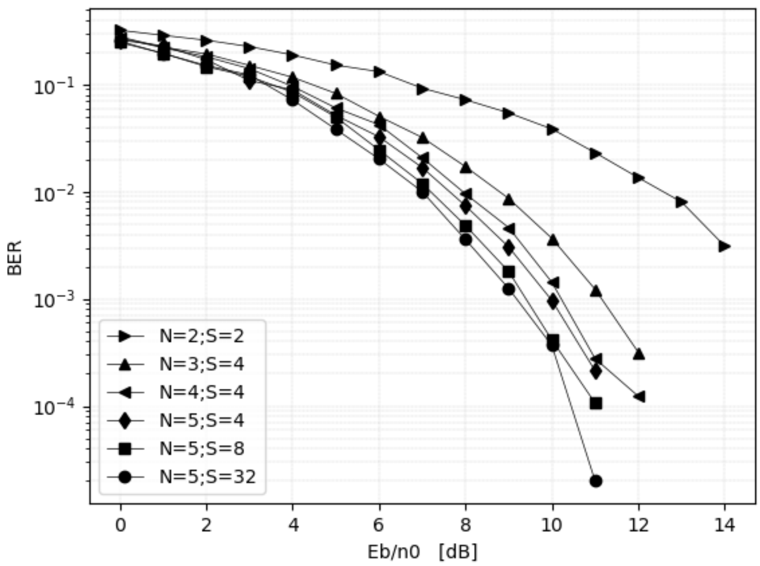

图 8 Fig2($K=2$)仿真结果

由上图可知，随着信噪比的增大，系统的误比特率降低。在信噪比相同的情况下，误比特率随着N和S的增大而减小。

对比$K=1$与$K=2$的结果，发现两条曲线除了在$N=2,S=2$条件下差异较大外其余情况基本相同。其原因在于由图 2 可知线性调制信号的主要能量集中在$h_0(t)$中，因此差别不大。而在$N=2,S=2$条件下，维特比网格图过于简单，因此译码效果并不好，增大K的值能够给出更多信息，较为显著影响译码效果，因此图 8 中译码效果优于图 7 中的效果，但均比不上原图中$N=2,S=2$的情况。其余部分中在某些条件下$K=1$的情况优于$K=2$的情况，其原因在于$K=1$条件下白化滤波器是严格求解得到的，而$K=2$条件下是通过数值方法求得的，因此存在有误差。

和原图相比，除去$N=2,S=2$，其余情况下仿真结果要得到相同误比特率需要信噪比增加 0.5dB，基本满足要求。误差的来源为sps有限以及白化滤波器计算的精度等。

3.2 Fig3 仿真结果(此处K指 WF 阶数)

3.1.1 $K=2$条件下

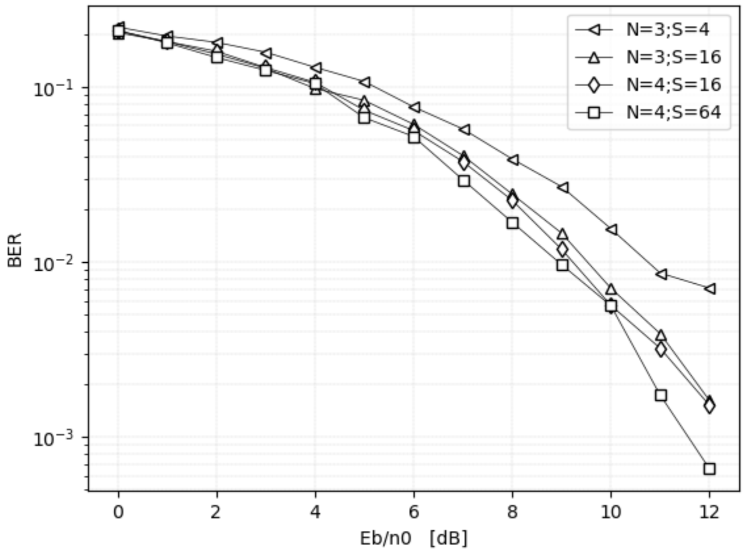

图 9 Fig3($K=2$)仿真图

3.1.2 $K=3$条件下

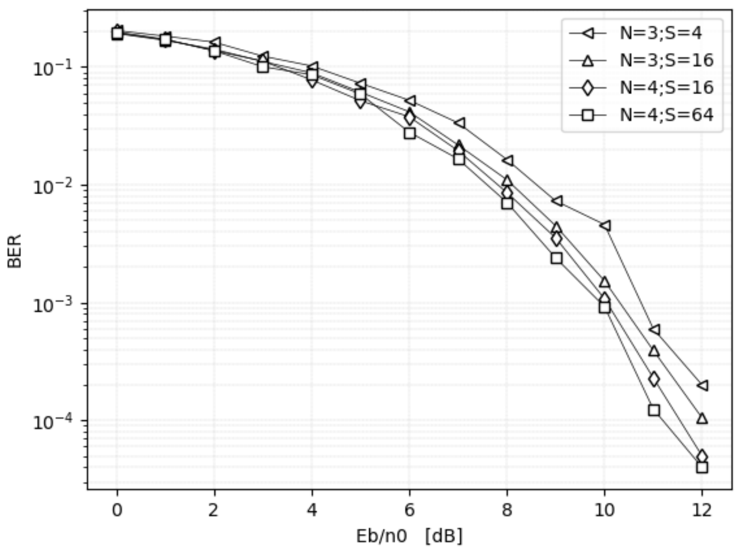

图 10 Fig3($K=3$)仿真图

由上图可知，同样的，随着信噪比的增大，系统的误比特率降低。在信噪比相同的情况下，误比特率随着N和S的增大而减小。

对比$\mathrm{K}=2,\mathrm{K}=3$的情况，发现$\mathrm{K}=3$的情况远好于$\mathrm{K}=2$的情况，其原因在于在计算$\mathrm{K}=2$情况下白化滤波器系数的过程中计算过程存在有舍入的误差。多步计算累积之后导致系数误差较大，从而经过白化滤波器后噪声并不是白的。理论上图 9 与图 10 的区别应当不大，之后可以进一步精确计算$K=2$时的白化滤波器来进行验证。

对比图 10 和原图中 Fig 3，二者基本相同，达到了预期的目标。

• 参考文献

[1] G. Colavolpe and R. Raheli, "Noncoherent sequence detection of continuous phase modulations," in IEEE Transactions on Communications, vol. 47, no. 9, pp. 1303-1307, Sept. 1999, doi: 10.1109/26.789664.

[2] J. B. Anderson, T. Aulin, and C.-E. Sundberg, *Digital Phase Modulation*. New York: Plenum, 1986.

[3]《通信与网络》课程课件，数字基带调制(3)

[4] U. Mengali and M. Morelli, "Decomposition of M-ary CPM signals into PAM waveforms," in IEEE Transactions on Information Theory, vol. 41, no. 5, pp. 1265-1275, Sept. 1995, doi: 10.1109/18.412675..

[5] G. Colavolpe and R. Raheli, "Reduced-complexity detection and phase synchronization of CPM signals," in IEEE Transactions on Communications, vol. 45, no. 9, pp. 1070-1079, Sept. 1997, doi: 10.1109/26.623071.

[6] P. R. Mothyka and J. A. Cadzow, “The factorization of discrete process spectral matrices,” IEEE Trans. Automat. Contr., vol. AC-12,pp. 698–707, Dec. 196

[7] M. V. Eyuboglu and S. U. H. Qureshi, "Reduced-state sequence estimation with set partitioning and decision feedback," in IEEE Transactions on Communications, vol. 36, no. 1, pp. 13-20, Jan. 1988, doi: 10.1109/26.2724.