第一章 设计内容及要求

基于 MATLAB 产生 m 序列

要求：

1．通过 matlab 编程产生 m 序列的产生原理及其产生方法。

2．对特定长度的 m 序列，分析其性质，及其用来构造其它序列的方法。

第二章　m 序列设计方案的选择

2.1 方案一

MATLAB 编程非常简单，无需进行变量声明，可以很方便的实现 m 序列。

2.2 方案二

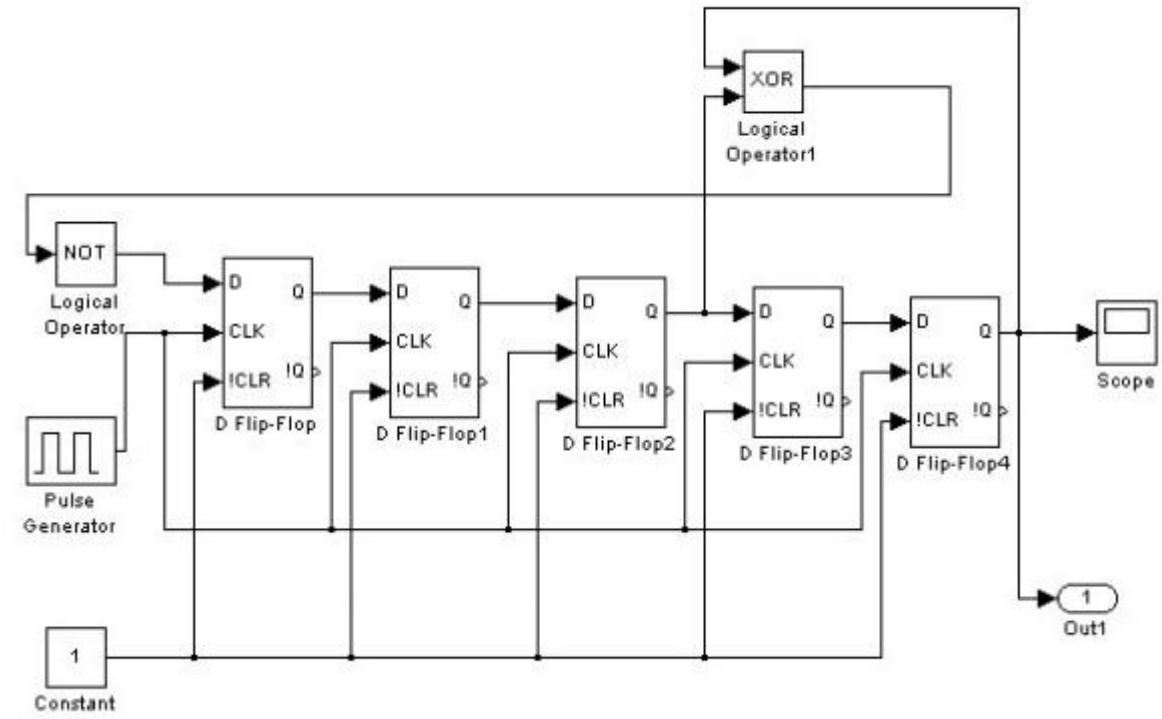


图 2.1 Simulink 实现 m 序列

Simulink 是 MATLAB 最重要的组件之一，它提供了一个动态系统建模，仿真和综合分析的集成环境。在此环境中无需大量书写程序，而只需通过简单直观的鼠标操作，就可构造出复杂的系统。Simulink 具有适应性广，结构及流程清晰及仿真精细等优点，基于以上优点，Simulink 已被广泛的运用到控制理论和数字信号处理的复杂仿真和设计。

通过比较方案一和方案二，发现方案一的有点具有通用性而方案二利用 MATLAB 的 Simulink 直接搭建模块，在移位寄存器较少的情况下利用此方法比较简单，可是当移位寄存器的个数增多时，要搭建那么多的模块就显的很繁琐了，缺乏通用性，因此本次实验选择方案一。

第三章 m 序列的产生及性质

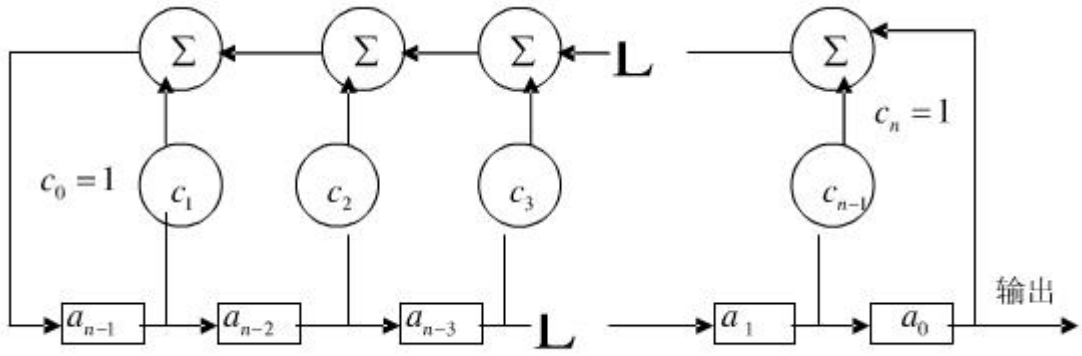
3.1 m 序列的产生原理、结构及产生

m 序列是最长线性反馈移位寄存器序列的简称，m 序列是由带线性反馈的移位寄存器产生的。

由 n 级串联的移位寄存器和反馈逻辑线路可组成动态移位寄存器，如果反馈逻辑线路只由模 2 和构成，则称为线性反馈移位寄存器。

带线性反馈逻辑的移位寄存器设定初始状态后，在时钟触发下，每次移位后各级寄存器会发生变化，其中任何一级寄存器的输出，随着时钟节拍的推移都会产生一个序列，该序列称为移位寄存器序列。

n 级线性移位寄存器的如图 3.1 所示：



图中 C_i 表示反馈线的两种可能连接方式，C_i=1 表示连线接通，第 n-i 级输出加入反馈中；C_i=0 表示连线断开，第 n-i 级输出未参加反馈。

因此，一般形式的线性反馈逻辑表达式为

$$a_n = C_1 a_{n-1} \oplus C_2 a_{n-2} \oplus \mathrm{L} \oplus C_n a_0 = \sum_{i=1}^{n} C_i a_{n-i} \pmod 2$$

------表达式 3.1

将等式左边的 a_n 移至右边，并将 $a_n=C_0a_n(C_0=1)$带入上式，则上式可以写成

$$0 = \sum_{i=0}^{n} C_i a_{n-1}$$

-------表达式 3.2

定义一个与上式相对应的多项式

$$F(x) = \sum_{i=0}^{n} C_i x^i$$

--------表达式 3.3

其中 x 的幂次表示元素的相应位置。该式为线性反馈移位寄存器的特征

多项式，特征多项式与输出序列的周期有密切关系。当 F(x)满足下列三个条件时，就一定能产生 m 序列：

（1）F(x)是不可约的，即不能再分解多项式；

（2）F（x）可整除 x^n+1，这里 $p=2^n+1$；

（3）F（x）不能整除 x^n+1，这里 q<q.

满足上述条件的多项式称为本原多项式，这样产生 m 序列的充要条件就变成了如何寻找本原多项式。

3.2 m 序列的基本性质

（1）均衡性。在 m 序列一个周期中‘1’的个数比‘0’要多 1 位，这表明

序列平均值很小。

（2）m 序列与其移位后的序列模 2 相加，所得的序列还是 m 序列，只是相位不同而已。例如：1110100 与向又移 3 位的序列 1001110 相对应模二相加后的序列为 0111010，相当于原序列向右移一位后的序列，仍为 m 序列。

（3）m 序列发生器中移位寄存器的各种状态，除全 0 状态外，其他状态只在 m 序列中出现一次。

（4）m 序列发生器中，并不是任何抽头组合都能产生 m 序列。理论分析指出，产生的 m 序列数由下式决定：

$$\Phi(2^n - 1)/n$$

--------表达式 3.4

其中 ϕ（X）为欧拉数。例如 5 级移位寄存器产生 31 位 m 序列只有 6 个。

（5）m 序列具有良好的自相关性，其自相关系数：

$$\rho(j)=\begin{cases}1 & j=0\\ -\dfrac{1}{N} & j\neq 0\end{cases}$$

--------表达式 3.5

从 m 序列的自相关系数可以看出 m 序列是一个狭义伪随机码。

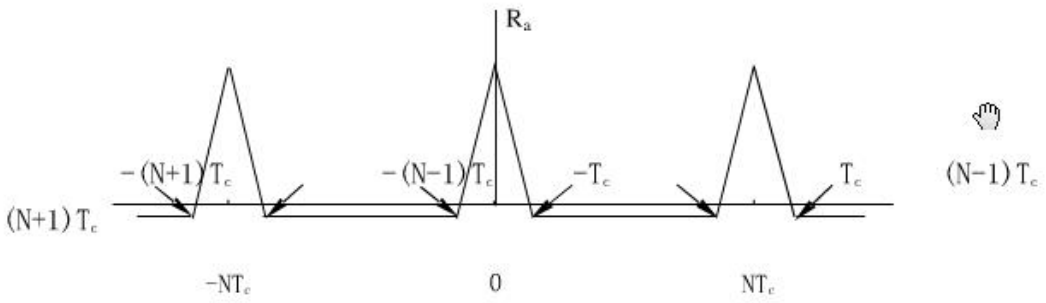


图 3.2 m 序列信号的自相关函数

3.3 生成 m 序列的模块

根据 m 序列的生成原理图，如图 3.1 所示，由图可知 m 序列是多级移位寄存器通过线性反馈再进行模二相加产生的，最后一位一位输出观察其波形图。程序中使用的代码如下：

```
N=2^length(reg1)-1;
for k=1:N
    a_n=mod(sum(reg1.*coeff1(1:length(coeff1)-1)),2);
    reg1=[reg1(2:length(reg1)),a_n];
    out1(k)=reg1(1);
end
```

其中 N 为 m 序列的长度，值为（2^6-1）。由程序已定义了移位寄存器的初始状态和抽头系数，在此基础上进行反馈，后进行模二加，所得的结果为输出的第一个值，初始状态向左移移位，而所得的输出值填补上变成新的序列，在进行第二次反馈和模二加，依次循环 N 次，产生 m 序列

第四章 m 序列构造其他序列

Gold 序列具有三值互相关函数，其值为：

$$-\frac{1}{p}t(r),-\frac{1}{p},\frac{1}{p}[t(r)-2] \qquad \text{（式 2.10）}$$

这里，

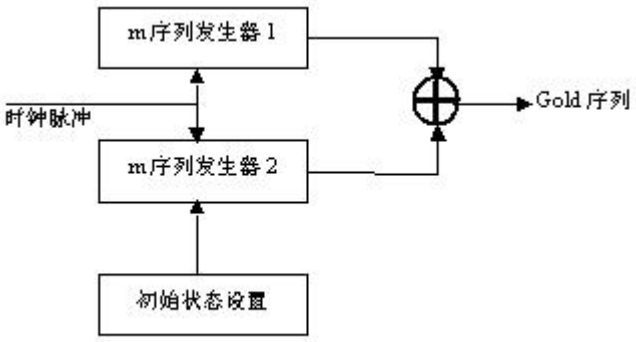
$$p=2^r-1, t(r)=\begin{cases}1+2^{0.5(r+1)}, r\text{为奇数}\\ 1+2^{0.5(r+2)}，\ r\text{为偶数但不是4的倍数}\end{cases} \qquad \text{（式 2.11）}$$

当 r 为奇数时，gold 序列中约有 50%的码序列归一化相关函数值为-1/p。当 r 为偶数但又不是 4 的倍数是，约有 75%的码序列归一化互相关函数值为 -1/p。

Gold 序列是 R.Gold 于 1967 年提出来的，它由两个 m 序列按下述方法演变而来的：把 2 个码长相同的 m 序列移位并进行模 2 加，如果相加的两个 m 序列是一对优选对，则相加的结果为一个 Gold 序列。

设有一对周期为 $N=2^r-1$ 的 m 序列优选对 {a}，{b}，以其中任意一个序列为基准序列，如 {a}，对另一个序列 {b} 进行移位 i 次，得到 {b} 的移位序列$\{b_i\}$，然后与序列{a}进行模二加得到一个新的周期为 N 的序列{c}，则称新序列 {c} 为 Gold 序列，既

$\{c_i\}=\{a\}+\{b\}$　　i=0，1，2，....,N



4.1 Gold 序列的产生方框图

产生 gold 序列的程序代码如下：

```
gold=mod(out1+out2,2);
```

第五章 程序调试及运行结果

5.1 仿真设计流程图

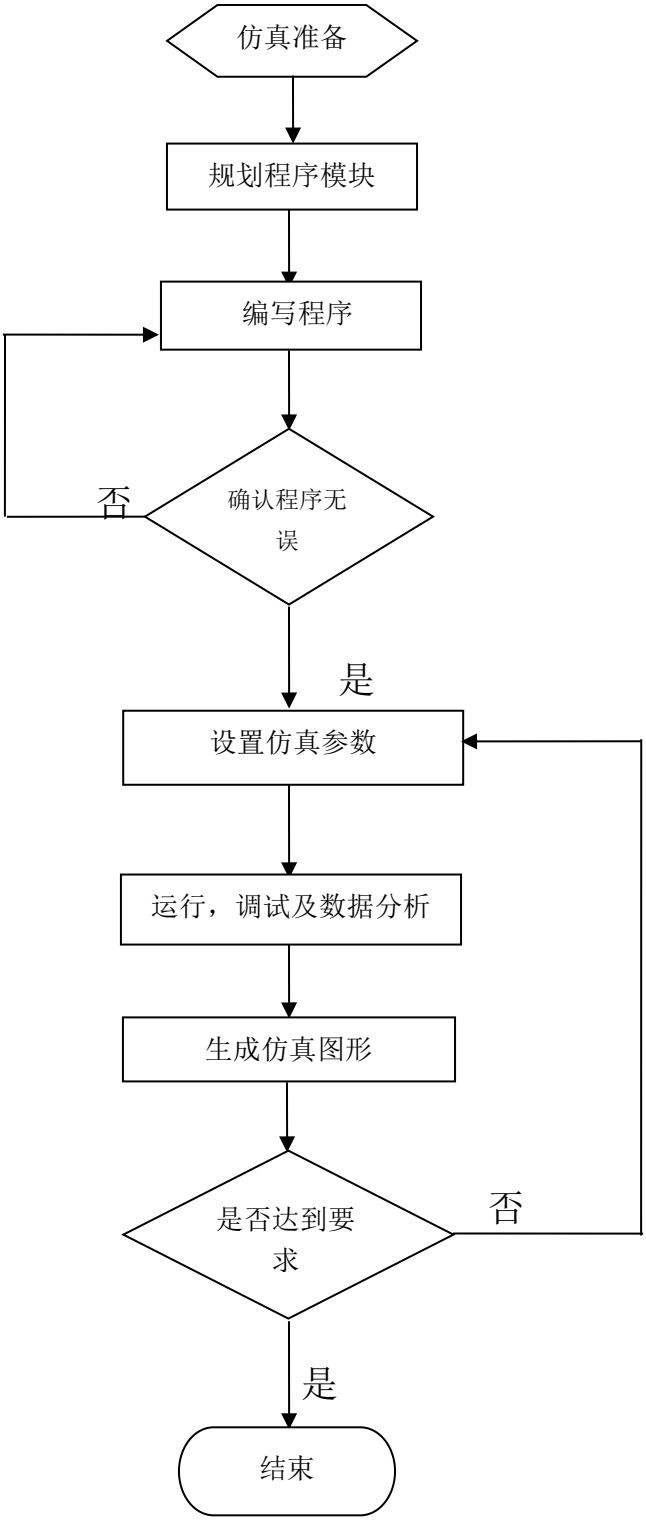


图 5.1　实验仿真流程图

5.2 实验的调试与运行结果

程序中把移位寄存器的初始值定义全为 1，抽头系数定义为[1000011]和

[1100111],根据公式 m 序列的长度=2^n-1，可知道所得的两个 m 序列的长度都为 63，所利用的移位寄存器为 6 个。代码如下：

```
reg1=ones(1,6);                    %寄存器初始状态
coeff1=[1 0 0 0 0  1 1];           %设置系数
reg2=ones(1,7);                    %寄存器初始状态
coeff2=[1 0 0  1 1 1 1];           %设置系数
```

程序检测无误后，运行程序，得到图形如下：

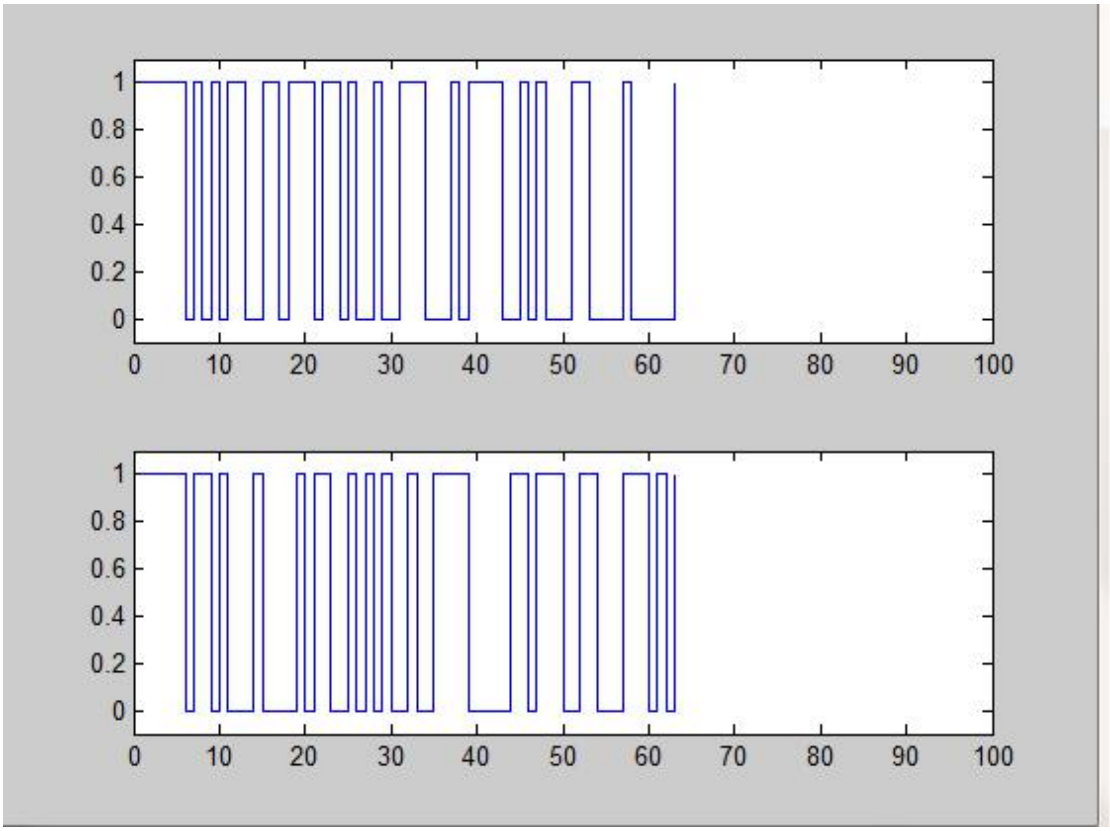


图 5.2 运行后的两 m 序列波形仿真图

根据产生 Gold 序列的原理，运行程序，得到如下 Gold 序列的仿真图；

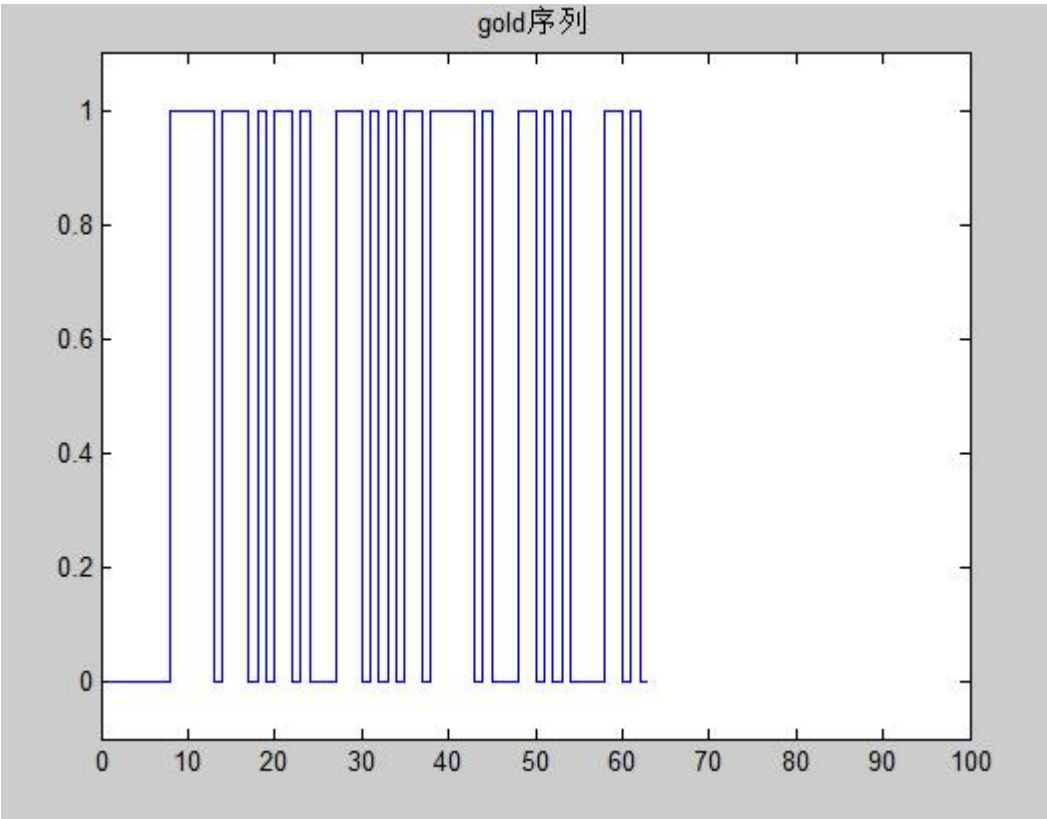


图 5.3 运行后 Gold 序列的仿真图

自相关性：首先将第一个 m 序列变成双极性的序列，在与本身进行移位相乘进行积分运算，代码如下：

```
out1=2*out1-1;          %变为双极性序列
for j=0:N-1
    rho(j+1)=sum(out1.*[out1(1+j:N),out1(1:j)])/N;
end
j=-N+1:N-1;
rho=[fliplr(rho(2:N)),rho];
figure(3)
plot(j,rho);
axis([-10 10 -0.1 1.2]);title('第一个 m 序列的自相关函数')
```

互相关性：第一个 m 序列的函数与第二个 m 序列函数的移位相乘进行积分运算。代码如下：

```
for j=0:N-1
    R(j+1)=sum(out1.*[out2(1+j:N),out2(1:j)]);
```

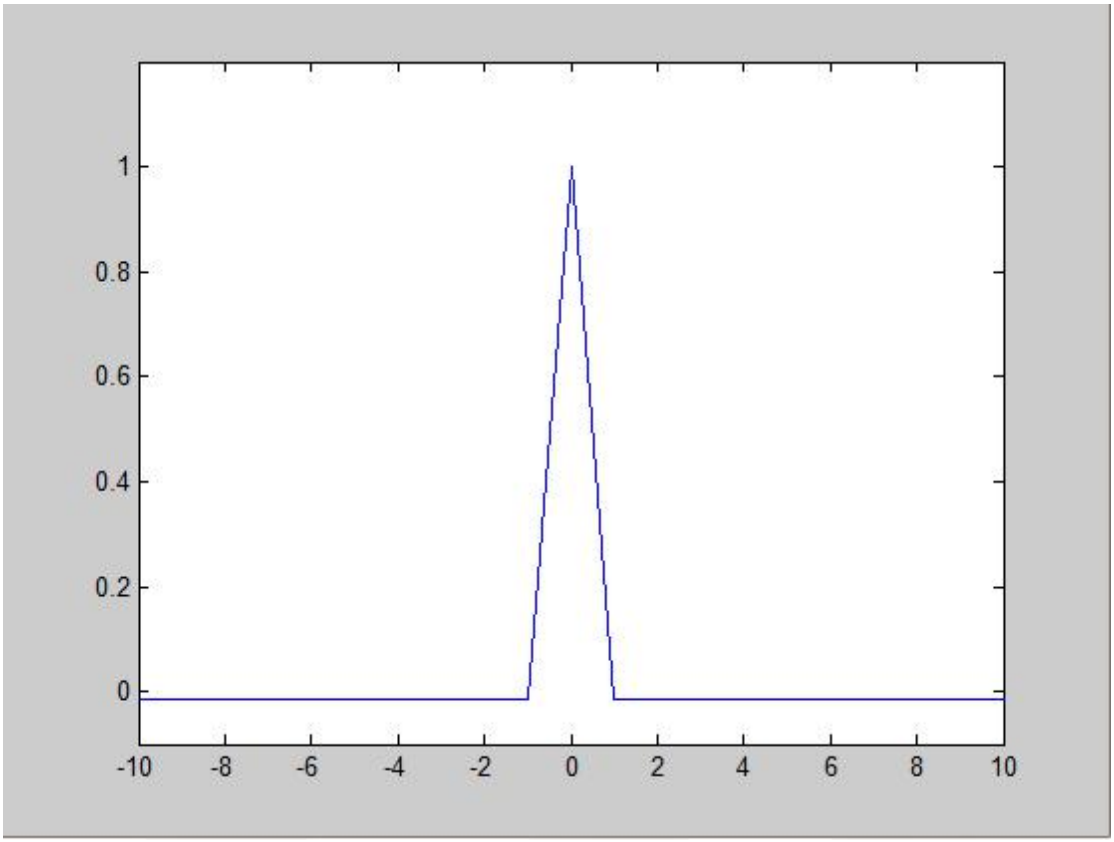


图 5.4 m 序列自相关性仿真图

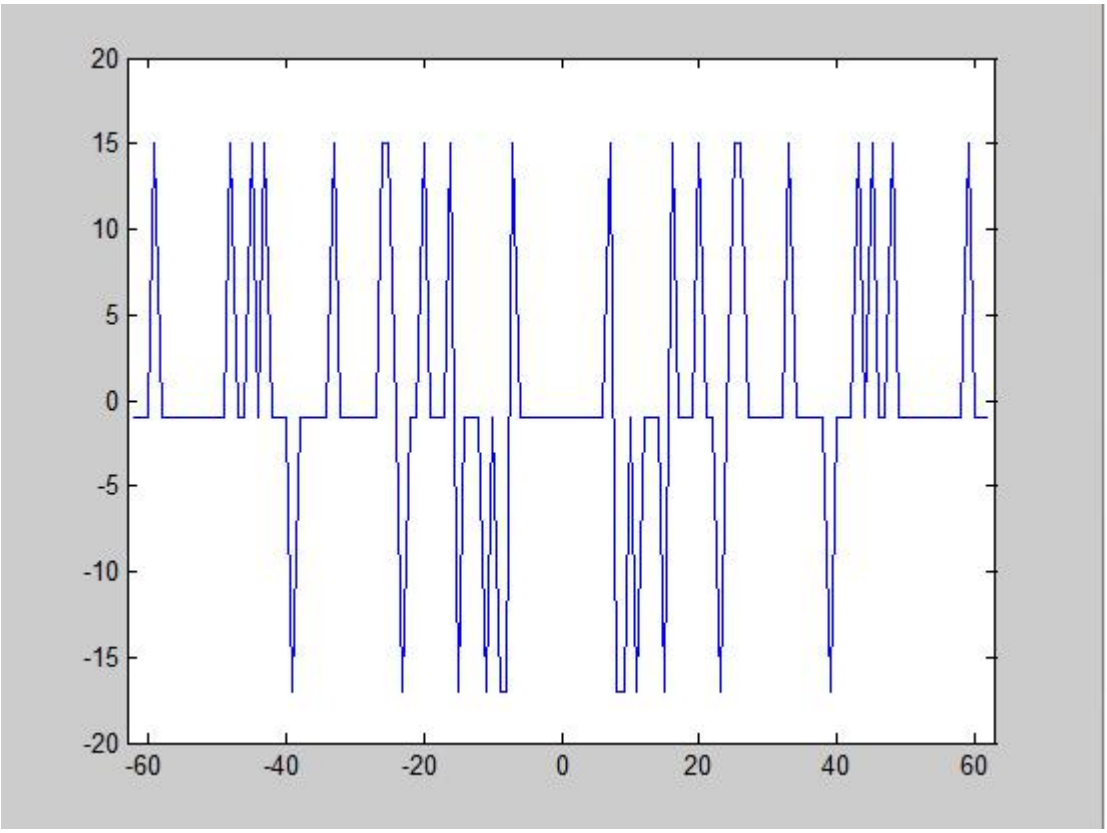


图 5.5 两 m 序列的互相关性仿真图

5.3 错误排除

实验过程中出了许多错误，特别是在对序列的自相关性上，缺乏了对自相关性的了解，导致出的波形错误，但在查找资料，修改自相关函数，终于得到了正确的仿真图。

结论

课程设计是一个十分有价值、有意义的实践活动，把一个课题设计好不是一步到位的，是经过反复修改，不断调试的过程，期间有困难也有乐趣，使对工程实践有了一个初步的认识。

本次课程设计实现了设计要求，利用软件实现 m 序列的生成，通过这次实验不但加深了对 m 序列的了解，而且对 MATLAB 的编程有了很好的掌握，虽然在仿真过程中会出现一些如程序不对或出的仿真图没有达到实验要求，如 m 序列中‘1’的个数要比‘0’的个数多 1，而实际出的仿真图‘1’和‘0’的个数却是相等的，最后在调整了顶层原理图才使得仿真图正确，在不断的程序调整中提高了自己的能力。

m 序列可以用软件实现，也可以用硬件实现，但是通过此次实验看出了软件的诸多优点。在课程设计的过程中，查询了大量的资料，通过相关资料的查询，使我对通信领域的有关知识有了一定的了解，扩大了知识面。

参考资料

[1]肖国镇，梁传甲.伪随机序列及其应用[M].北京：国防工业出版社.1985

[2]林可祥，汪一飞.伪随机码的原理与应用[M].北京：人民邮电出版社.1998.

[3]吴先用，邹学玉.一种 m 序列伪码发生器的产生方法[J].西安：西安电子科技大学出版社.2003

附录

```
clear;
reg1=ones(1,7);                    %寄存器初始状态
coeff1=[1 0 0 0 0 1 1];            %设置系数
N=2^length(reg1)-1;
%产生 m 序列
for k=1:N
    a_n=mod(sum(reg1.*coeff1(1:length(coeff1)-1)),2);
    reg1=[reg1(2:length(reg1)),a_n];
    out1(k)=reg1(1);
end
reg2=ones(1,7);                    %寄存器初始状态
coeff2=[1 10 0 1 1 1];             %设置系数
N=2^length(reg2)-1;
for k=1:N
    a_n=mod(sum(reg2.*coeff2(1:length(coeff2)-1)),2);        %移位，反
馈
    reg2=[reg2(2:length(reg2)),a_n];                         %反馈
    out2(k)=reg2(1);                                         %取第一个
值输出
end
 %产生 gold 序列
gold=mod(out1+out2,2);
c=1:N;
figure(1)
[b1,t1]=stairs(c,out1);
subplot(2,1,1);plot(b1,t1);
axis([0 130 -0.1 1.1]);title('第一个 m 序列');
[b2,t2]=stairs(c,out2);
subplot(2,1,2);plot(b2,t2);
```

```
axis([0 130 -0.1 1.1]);title('第二个 m 序列');
figure(2)
[b3,t3]=stairs(c,gold);
plot(b3,t3);
axis([0 130 -0.1 1.1]);title('gold 序列')
out1=2*out1-1;        %变为双极性序列
out2=2*out2-1;
%自相关函数
for j=0:N-1
    rho(j+1)=sum(out1.*[out1(1+j:N),out1(1:j)])/N;
end
j=-N+1:N-1;
rho=[fliplr(rho(2:N)),rho];
figure(3)
plot(j,rho);
axis([-10 10 -0.1 1.2]);title('第一个 m 序列的自相关函数')
%互相关函数
for j=0:N-1
    R(j+1)=sum(out1.*[out2(1+j:N),out2(1:j)]);
end
j=-N+1:N-1;
R=[fliplr(R(2:N)),R];
figure(4)
plot(j,R);
axis([-N N -20 20]);title('两个 m 序列的互相关函数');
```