通信信号处理大作业

无 97 张凯 2019011159

一、实验要求

作业1

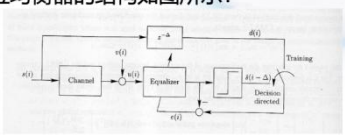
符号{$s(i)$}通过信道的传输附加复高斯白噪声{$v(i)$}。接收信号{$u(i)$}通过FIR均衡器对输入序列进行估计得到{$\hat{s}(i-\Delta)$}，并被送入判决器。均衡器有两种操作模式：训练模式（输入信号的延时作为参照序列）及面向判决模式（判决装置的输出作为参照序列）。输入序列{$s(i)$}为QAM 星座点（如4-QAM、16-QAM、64-QAM、256-QAM）。

$\varepsilon-$NLMS算法是一种基于LMS算法的改进算法：其中的更新过程变为：

$$\mathbf{C}_{k+1}=\mathbf{C}_k+\frac{\mu e_k^* x_{k-n}}{\varepsilon+\sum_{i=-N}^{N}|x_{k+i}|^2}, n=\pm1,\pm2,\cdots,\pm N$$

作业1

自适应均衡器的结构如图所示：



a. 写一个程序，用500个QPSK符号训练一个自适应均衡器，然后传输5000个16-QAM符号时采用面向判决模式。信噪比SNR为30dB，$\Delta=15$，均衡器长度$L=35$。采用$\varepsilon-$NLMS算法训练均衡器，其中步长$\mu=0.4$，校正项$\varepsilon=10^{-6}$。画出{$s(i),u(i),\hat{s}(i-\Delta)$}的散点图。

通信信号处理 27

作业1

b. 和（a）中的设置相同，在训练的迭代次数分别为150、300和500次时，画出并比较均衡器输出的散点图。取$\mu=0.001$用LMS算法再进行仿真。

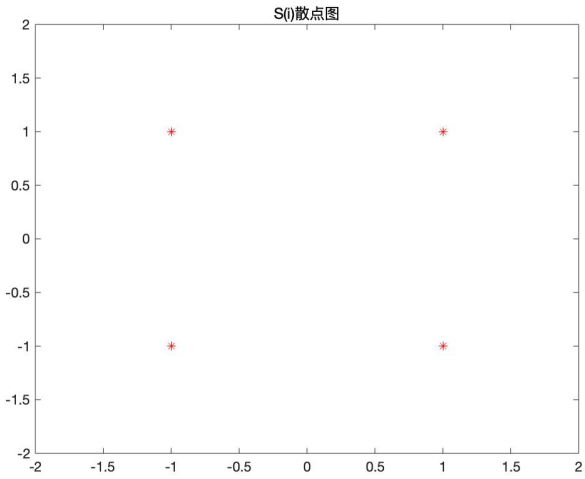
c. 将输入信号改为256-QAM的星座点，使用$\varepsilon-$NLMS算法，用500个训练符号，画出均衡器的输出散点图。

d. 输入为4-QAM、16-QAM、64-QAM、256-QAM调制数据，信噪比从5dB到30dB变化，变化步长为1dB。画出$\varepsilon-$NLMS算法对应的SER-SNR曲线（SER：symbol error rate）。

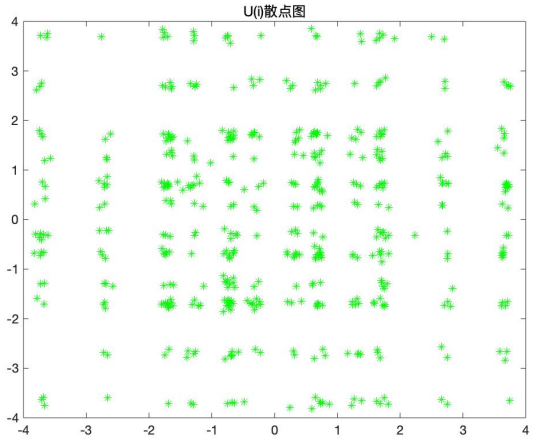
二、实验内容和分析

(a)训练均衡器并作图验证

采用 Matlab 进行代码编写，并调用了 Communication Tool 工具箱的相关函数，可以很容易地生成 QPSK，MQAM 等信号及加入 AWGN 噪声。因此，首先，进行 QPSK 的画图验证。



生成的 QPSK 星座图无误，再通过信道，即序列与信道传递函数序列[0.5,1,1.2,-1]进行卷积，之后调用 awgn()函数加入噪声，得到 U(i)序列，散点图如下：

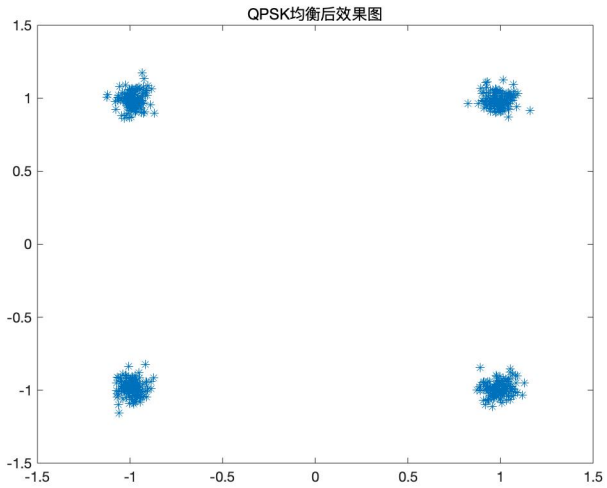


可以看到，经过信号和加入白噪声之后，星座图完全不再是原来的形状，但仍有一定的对称性。下面按照 LMS 算法公式，对 500 个点的 QPSK 序列进行训练，训练一个延时为$\Delta = 15$ 的均衡器。

核心公式为：

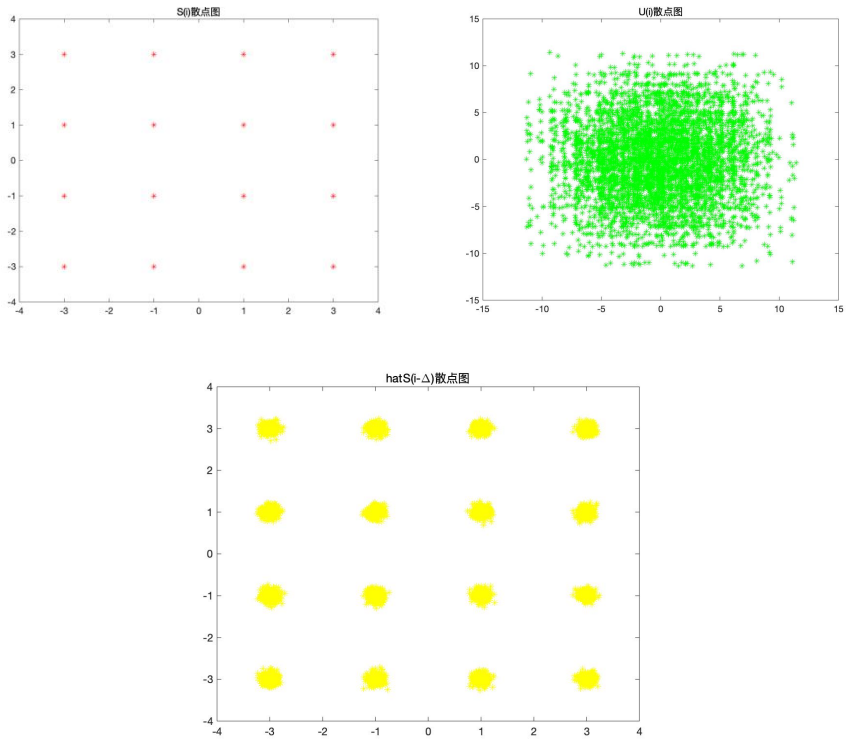
$$C_{k+1} = C_k + \frac{\mu e_k^* x_{k-n}}{\epsilon + \sum_{i=-N}^{N} |x_{k+i}|^2}$$

核心训练代码见 train_a.m 文件，其中由于存在$\Delta = 15$ 的延时，在计算卷积时有补零等的细节操作，这里不再详细阐述，训练 QPSK 后的均衡器系数 C 对 500 个已知序列做均衡，结果图如下：



可见，通过均衡器后，星座图又恢复了原来的形状，训练还是有效的。

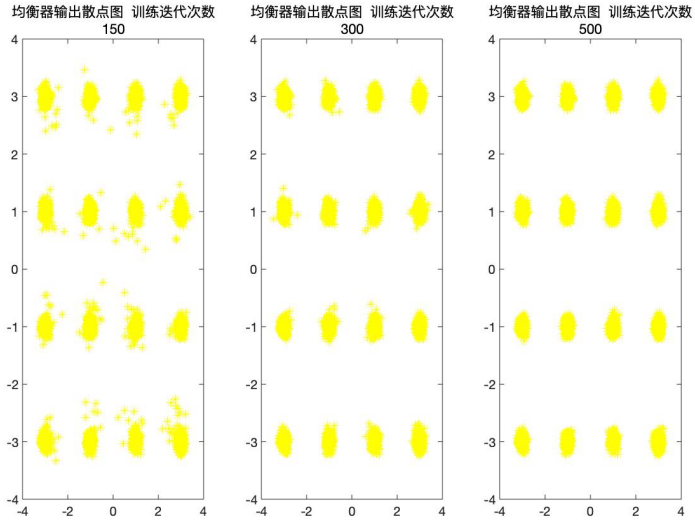
现在，按照题目要求，分别画出 5000 个 16-QAM 测试序列的 S(i)，U(i)，\hat{S(i-Δ)}的图像，其中判决时进入面向判决模式，代码见 test_a.py，结果如下所示：



其中，前两幅图分别是原始的 16-QAM 符号和通过信号+噪声之后的星座图，可见，通过信号和加入噪声之后，16-QAM 星座图完全不再是原来的形状，经过均衡器后，星座图又基本恢复了原来的形状，由图易知，没有点判决错误，即此时 SER=0。

(b)修改 NLMS 迭代次数，进行仿真；比较 LMS 算法

修改 NLMS 训练迭代次数，分别为 150，300，500，核心代码见 train_b.py 和 test_b.py 依然采用(a)中 5000 点 16-QAM 序列进行判决，结果如下：

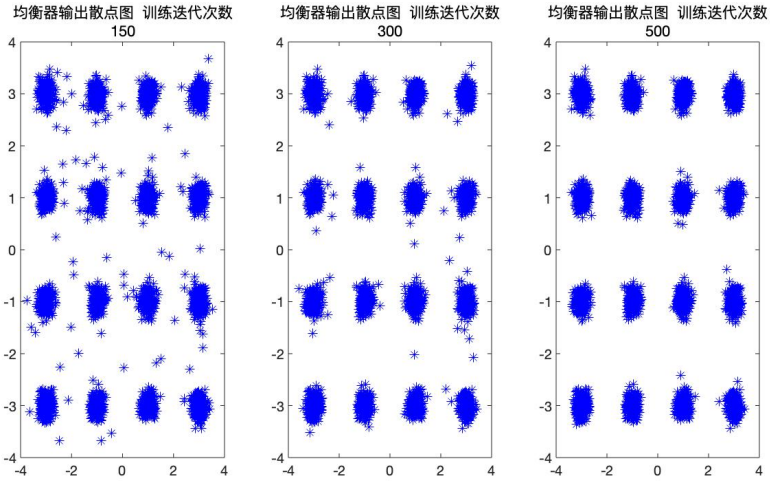


如图可见，采用同样的算法，训练迭代次数越多，均衡后星座图越理想，判决越准确。

之后，在核心代码 train_b.py 和 test_b.py 中，修改 mode 参数，改为 LMS 算法模式，同时修改相关参数和公式，公式上只是把归一化分母去掉，参数需要修改μ为 0.001，核心公式如下：

$$C_{k+1} = C_k + \mu e_k^* x_{k-n}$$

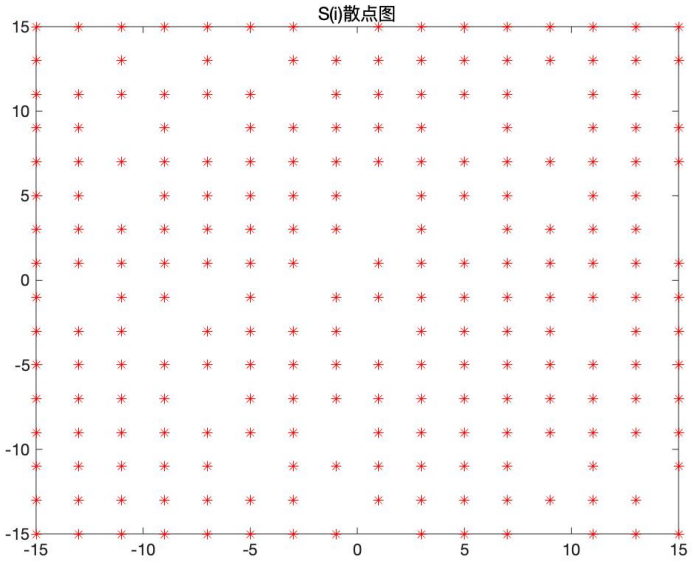
依然采用(a)中 5000 点 16-QAM 序列进行判决，结果如下：

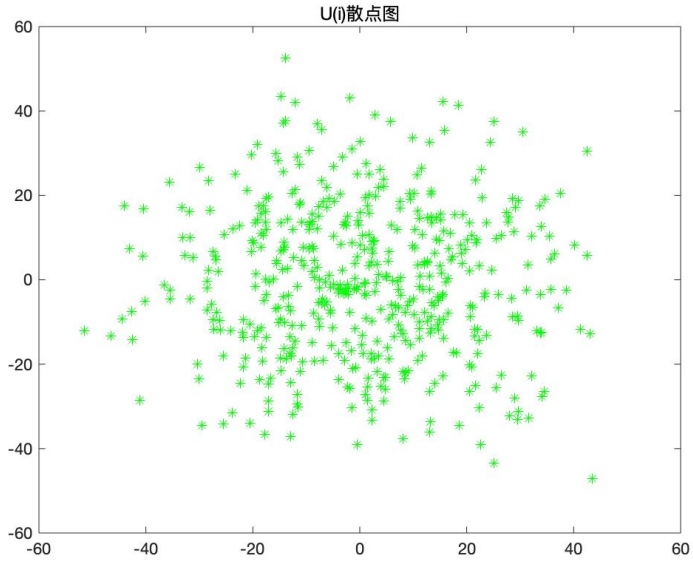


可见，在同样的训练迭代次数下，LMS 算法效果不如 NLMS 理想，收敛速度更慢，但是当达到较高的迭代次数时，两者差别逐渐缩小。

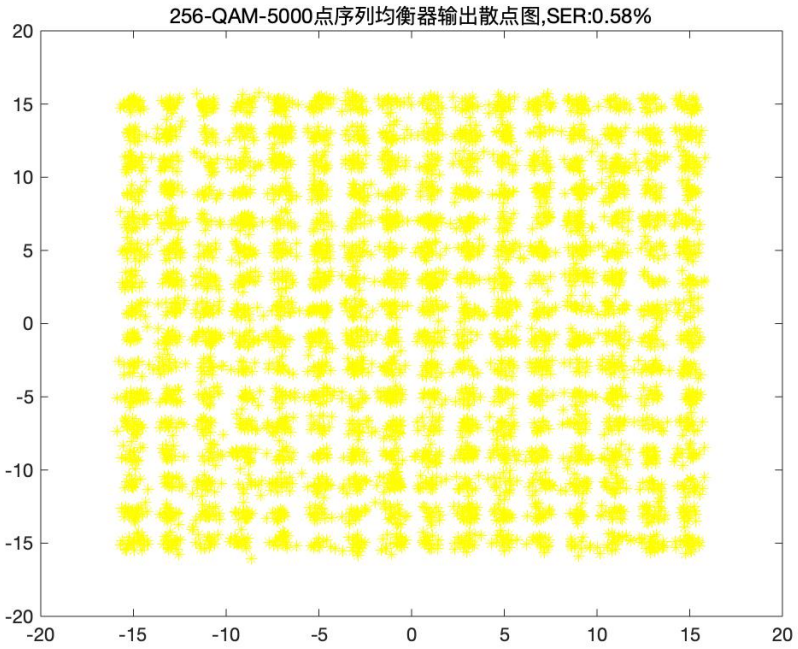
(c)256-QAM　500 点　NLMS 训练

对于 256-QAM 来说，500 点的训练确实太少，生成随机序列画出来无法保证星座图都有位置。画图如下：





训练 500 点后，进入判决模式，序列为 5000 点，效果如下：



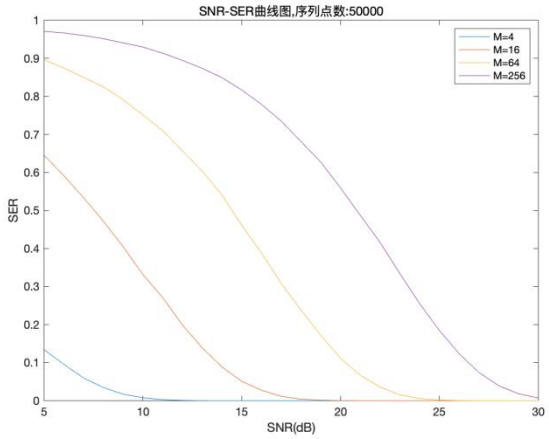
可见效果依然非常好，所以虽然训练序列较短，训练后的均衡器还是相当有效的，至少所有的点均衡后都能在星座点附近，计算所得 SER = 0.58%，还是相当有效的，也进一步验证了均衡器训练结果的正确性。(c)全部代码均见 train_and_test_c.m。

(d)画出不同 QAM 调制方式下不同信噪比的 SER-SNR 曲线

之前的结果都极为理想，可能是因为 SNR 太高了，因此，现在调节不同的 SNR，来比较 SNR 对于均衡和判决结果 SER 的影响：

均衡器的训练均采用(a)中的训练方式，500 个 QPSK 进行训练，之后，针对不同的 MQAM 及 SNR，进行不同参数下的判决模式，计算 50000 点序列判决的 SER 并作图，具体代码见 test_d_SER.m。

结果如图所示：



由图可见，随着 SNR 逐渐上升，SER 逐渐降低至 0(实际中不可能为 0，但可忽略不计)，同时，在相同的 SNR 下，对于不同的 M 来说，QAM 的 SER 也不同，在 SER 均不为 0 时，随着 M 增大，SER 增加，这是因为训练时主要针对 QPSK，即 4QAM 序列进行训练，随着符号数增加，受均衡器阶数和训练时序列的限制，均衡的能力会有所降低。因此，实验的结果符合理论的分析和直观的理解。并且，我们可以通过上图知道 SNR 和调制方式均能影响误符号率，因此，在实际通信中，不但要考虑效率，也要考虑调制方式带来的误符号率的影响。

三、实验小结

本次实验中，通过编写代码进行仿真，验证了 ε -NLMS 算法的正确性，对算法的性能做了分析和对比。实验开始时，由于没有正确使用 Matlab 工具箱的相关函数, 进展较慢, 之后查阅了 Communication 工具箱的相关函数, 实现 QPSK、MQAM 等信号以及加入白噪声等细节操作就变得十分容易了，之后按照课件公式进行编写代码也十分顺利。实验通过画图的方式，更加直观地体现了均衡器的作用，也加深了我对于均衡这一部分理论知识的认识和理解。