《现代通信技术概论》期末考核

**1、（60分）选择一种面向未来的通信技术，结合课程内容，详细回答：**

**（1）技术产生的背景；**

**（2）技术的基本原理；**

**（3）技术的可能应用场景。**

新一代卫星互联网

背景：

起始于20世纪80年代末，至今已经进行了三次迭代：

1. 第一代使用 C、L、S 频段，以话音和物联网服务为主，定位是全面替代地面通信系统
2. 第二代使用 C、L、S、Ka 频段，以升级带宽和拓展综合服务为主，定位是与地面通信系统并存
3. 第三代，即新一代使用 Ku、Ka、V 频段，以提供高速低延时的互联网服务为主，定位是与地面通信系统互补

2015年后，新一代卫星互联网技术趋向小型化、大容量演进。以 SpaceX 的星链计划、一网公司的 OneWeb 计划和加拿大电信卫星公司的 Telstar 计划为首，使用大量低轨宽带/高通量卫星实现大规模组网，从而为全球提供高速低延时的互联网服务。

基本原理：

一条完整的通信链路包括地面系统、上行下行链路和通信卫星；分为空间段、地面段和用户段。

空间段：

* 若干通信卫星组成的星座。
* 通信卫星主要作用是转发无线电信信号。
* 卫星中的有效载荷基于特定的频段，其中的天线分系统接受上行信号后，经过转发器分系统对信号进行放大🡪变频🡪放大，转换为下行信号，再通过天线分系统传送至地面；
* 一颗卫星可以携带多个转发器，每个转发器可以同时接收/转发多个地面信号；在功率和带宽固定时，转发器数量和卫星容量正比。

地面段：

* 用于连接卫星网络和地面网络。
* 用于实现卫星与地面、终端与终端之间的互联，以及管控卫星网络。
* 包括关口站、地面卫星控制中心、遥测和指令站等，以及主站和陆地链路匹配的接口。

用户段：

* 各类用户终端设备。

应用场景：

1. 物联网。卫星互联网覆盖广，受天气和地理影响小，因此可以用于：1）农业上可通过其收集大面积农场的相关数据；2）工程上可通过其进行偏远地区的土木工程作业监控；3）海洋运输方面则可通过其对运输过程全程监控，提高运输效率和安全。
2. 生态环境监测和应急通信。低轨卫星互联网可以提高数据和预警的传送速度，从而实现实时监控，提高防护监测工作的效率；在灾害发生时，其收地理影响小的特性能帮助救灾人员和相关情况的通信与传输。
3. 机载/船载Wi-Fi。由于地面基站等设施的局限，飞机飞行过程中的乘客无法自由使用互联网，而新一代卫星互联网的商业化可以解决用户在飞机飞行过程中/轮船行驶过程中无法使用互联网的问题，还能提供高速率的带宽。

**2、（40分）MIMO技术是4G和5G移动通信系统的核心关键技术之一。现有一对收发机，各配有4根天线，它们之间的信道为H，发射机的发射协方差矩阵为Q，则信息速率可表示为R=log2det(I+HQHT)。回答：**

**（1）假设H确知且它的奇异值分解为H=UAVT，其中A=diag(a1,a2,a3,a4)；矩阵Q未知且它的特征分解为Q=SBST，其中S为酉矩阵，B=diag(b1,b2,b3,b4)且bi>=0（i=1,2,3,4）。运用线性代数的知识，确定S的取值，使得MIMO信道可以分解为4条平行的子信道，并写出此时的信息速率表达式（用ai和bi来表示）。**

已知 MIMO 系统可以表示为：

由于题目要求为四条平行的子信道，那么 为：

对上述式子左右分别乘上 并根据相关性质得到：

令 和 得到：，同时可以得到：

由于原始信号的信道之间互相独立，则不同信道之间的协方差应为 0

并且对于协方差矩阵 ， 是酉矩阵，结合上面的式子可以得到公式为：

通过上式可以对应得到 和

此时可以计算信息速率为：

**（2）假设b1+b2+b3+b4=P，且P为一个已知的非负常数。运用最优化理论（如拉格朗日乘子法）的知识，确定bi的表达式，使得信息速率R最大。**

使用拉格朗日乘子法，在此题中的表述应为：

对 分别进行求导可以得到：

当上式结果为 0 时，可以得到期望的结果，此时：

又根据题目的 可以得到：

假设 ，可以得到如下判断：

（3）若H的每一个元素都独立服从（0,1）高斯分布。请编写程序（C++或者MATLAB，并附上可运行代码且代码要有详细注释）随机产生100个相互独立的H，画出P=1, 10, 102, 103, 104时，信息速率R的平均值（横坐标为P，纵坐标为R的平均值）。

由于使用的是基于标准正态分布的随机生成，因此每次运行代码的结果都有些许不同，生成的图片也会有些许不同，其中一张图片如下：

图表, 折线图

描述已自动生成

对应的输出为：

>> main  
  
avarageRate =  
  
 13.4775 17.2342 22.8919 34.2229 46.6311

代码如下：

% 主函数  
function main()  
 Ps = [1, 10, 1e2, 1e3, 1e4]; % 横坐标  
 avarageRate = [0, 0, 0, 0, 0]; % 纵坐标，平均速率  
 H\_nums = 100; % 100个独立的矩阵  
  
 for P\_index = 1:5  
 P = Ps(P\_index);  
  
 for H\_index = 1:H\_nums  
 [~, ~, A, ~] = generateMatrix(); % 生成矩阵  
 A\_ = sum(A); % 得到对角线的元素  
 A\_\_ = sort(A\_); % 进行排序  
  
 maxR = 0; % 最大速率  
  
 % 分情况讨论并计算 B  
 for index = 1:4  
 [lambda, B\_] = getLambda(A\_, P, index);  
  
 % 如果 lambda 值有效，那么计算速率并更新最大值  
 if (lambda ~= -1)  
 R = caculateRate(A\_\_, B\_);  
 maxR = max(R, maxR); % 更新最大值  
 end  
  
 end  
  
 avarageRate(P\_index) = avarageRate(P\_index) + maxR; % 累加求平均值  
 end  
  
 end  
  
 avarageRate = avarageRate / 100;  
  
 avarageRate  
  
 % 画图  
  
 semilogx(Ps, avarageRate, '-\*b');  
 axis([0, 1e4, 0, 55]);  
 set(gca, 'XTick', Ps);  
 set(gca, 'YTick', [0:10:55]);  
 xlabel('P')  
 ylabel('R')  
  
end  
  
% 生成矩阵  
function [H, U, A, V] = generateMatrix()  
 H = randn(4, 4); % 以标准正态分布生成 4\*4 的矩阵  
 [U, A, V] = svd(H); % 进行 SVD 分解  
end  
  
% 获取不同情况下的 lambda 和对应的 B  
function [lambda, B\_] = getLambda(A\_, P, index)  
 % 分情况进行讨论，index 与 lambda <= a\_index 对应  
  
 A\_\_ = sort(A\_); % 进行排序，方便讨论情况  
  
 switch index  
 case 4 % a3 < lambda <= a4  
 lambda = 1 / (P + 1 / (A\_\_(4)^2));  
  
 if ~(A\_(3) < lambda && lambda <= A\_(4))  
 lambda = -1;  
 end  
  
 case 3 % a2 < lambda <= a3  
 lambda = 2 / (P + 1 / (A\_\_(4)^2) + 1 / (A\_\_(3)^2));  
  
 if ~(A\_(2) < lambda && lambda <= A\_(3))  
 lambda = -1;  
 end  
  
 case 2 % a1 < lambda <= a2  
 lambda = 3 / (P + 1 / (A\_\_(4)^2) + 1 / (A\_\_(3)^2) + 1 / (A\_\_(2)^2));  
  
 if ~(A\_(1) < lambda && lambda <= A\_(2))  
 lambda = -1;  
 end  
  
 case 1 % lambda <= a1  
 lambda = 4 / (P + 1 / (A\_\_(4)^2) + 1 / (A\_\_(3)^2) + 1 / (A\_\_(2)^2) + 1 / (A\_\_(1)^2));  
  
 if ~(lambda <= A\_(1))  
 lambda = -1;  
 end  
  
 otherwise % 其他情况  
 lambda = -1;  
 end  
  
 B\_ = [0 0 0 0]; % 基础的 B  
  
 % 如果 lambda 有效，那么计算对应的 b\_i  
 if ~(lambda == -1)  
  
 for index = 1:4  
 B\_(index) = max(1 / lambda - 1 / (A\_\_(index)^2), 0);  
 end  
  
 end  
  
end  
  
% 计算信息速率  
function R = caculateRate(A\_, B\_)  
 AB\_ = (A\_.^2) .\* B\_ + 1;  
 R = sum(log2(AB\_));  
end