第二次实验报告

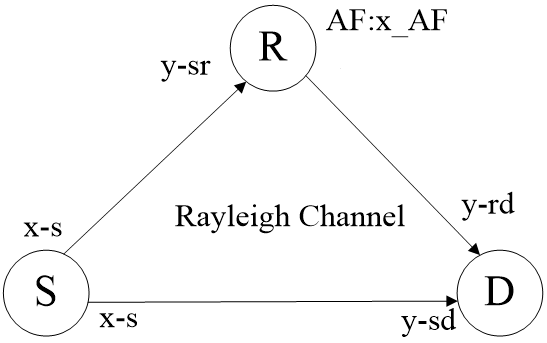
姓名：张书琪 学号：04016606

程序目的：本文档旨在说明协作通信网络中，中继采用AF协作网络的结构框架。

以及非协作系统中AF的实际和理论误码率曲线

运行环境：Matlab R2018b

通信模型：三端点通信模型：

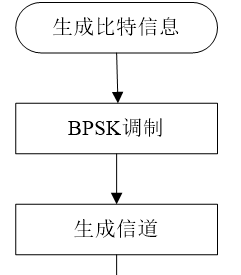


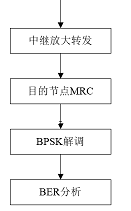
程序入口：code.m

程序结构图：

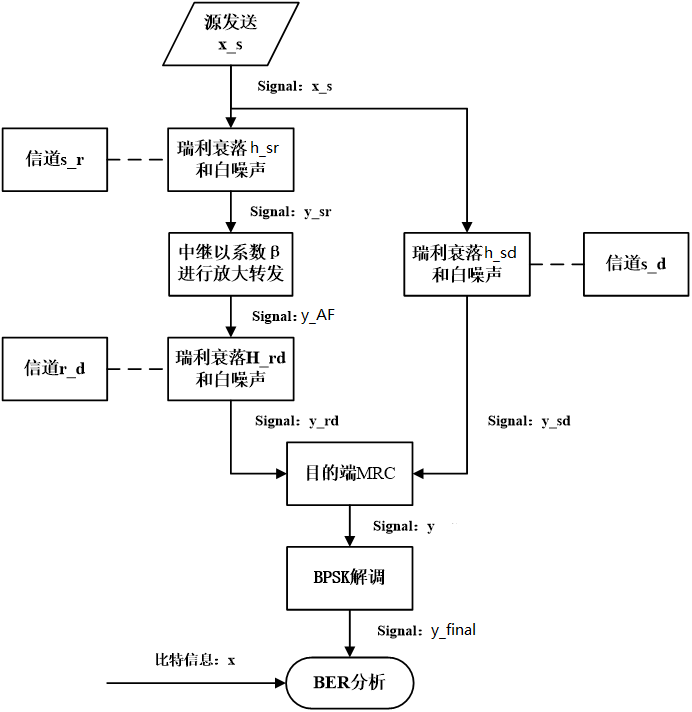


程序流程图：





AF策略流程图



程序代码：

%% original definition

MIN\_SNR\_dB = 0;

MAX\_SNR\_dB = 14;

INTERVAL = 0.5; % SNR interv

POW\_DIV = 1/2; % Power division factor功率分配系数,with cooperation, in order to guarantee a certain power of the total,

% respectively, the Source using the 1/2 of the power to send signals to the Relay and Destination

POW = 1; % without cooperation,Source send signals directly to the Restination with full power

Monte\_MAX=10^1; % the times of Monte Carlo,Limited to the computer configuration level, select the number to 10

%% (Signal Source) Generate a random binary data stream

sigma = 2; %方差开方

M = 2; % number of symbols

N = 10000; % number of bits

x = randi([0,1],1,N); % Random binary data stream %产生一个1\*N的矩阵，矩阵中元素取值范围为[0,(M-1)]

%% Modulate using bpsk

h = modem.pskmod(2);%产生2psk调制器

x\_s= modulate(h,x); %调制产生源信号

%x\_s = modulate(pskmod(M),x); % The signal 'x\_s' after bpsk modulation

%%

h\_sr = Rayleigh(1); %源与中继节点 信道衰落系数

h\_sd = Rayleigh(1); %源与目的节点 信道衰落系数

h\_rd = Rayleigh(1); %中继与目的节点 信道衰落系数

%% In different SNR in dB

snrcount = 0;

for SNR\_dB=MIN\_SNR\_dB:INTERVAL:MAX\_SNR\_dB

snrcount = snrcount+1; % count for different BER under SNR\_dB

%err\_num\_SD = 0; % Used to count the error bit

err\_num\_AF = 0;

%err\_num\_DF = 0;

for tries=0:Monte\_MAX

sig = 10^(SNR\_dB/10); % SNR, said non-dB 信噪比 非分贝单位

POW\_S = POW\_DIV; % Signal power信号功率

POW\_N = POW\_S / sig; % Noise power噪声功率

%SNR\_dB = 1.5;

%n\_sr = 0.5; %源与中继节点之间的信道噪声

%n\_sd = 0.6; %源与目的节点之间的信道噪声

%n\_rd = 0.65; %中继与目的节点之间的信道噪声

%y\_sr = h\_sr\*(Ps^0.5)\*x\_s + n\_sr; %中继节点接收到的信号

%y\_sd = h\_sd\*(Ps^0.5)\*x\_s + n\_sd; %目的节点首先接收到的信号

y\_sr = awgn( h\_sr\*(POW\_S^0.5)\*x\_s, SNR\_dB, 'measured'); %中继节点接收到的信号

y\_sd = awgn( h\_sd\*(POW\_S^0.5)\*x\_s, SNR\_dB, 'measured'); %目的节点首先接收到的信号

beta = sqrt( POW\_S/(POW\_S\*(h\_sr)^2 + POW\_N) ); %放大系数β

y\_AF = beta\*y\_sr; %中继放大后的信号y\_AF

y\_rd = awgn( h\_rd\*(POW\_S^0.5)\*y\_AF, SNR\_dB, 'measured');

Pr = 0.5\*y\_AF.^2; %中继放大后的信号功率

%y\_rd = [0.5\*(y\_AF)^2]^0.5\*h\_rd\*y\_AF + n\_rd; %目的节点接收到来自中继节点的信号

a1 = ((POW\_S)^0.5\*h\_sd)/POW\_N; % 目的接收来自源的加权系数

a2 = (beta\*Pr.^0.5\*h\_sr\*h\_rd)/((beta^2\*h\_rd^2+1)\*POW\_N); %目的接收来自中继的加权系数

y = a1.\*y\_sd + a2.\*y\_rd; %按照MRC方案进行合并，得到信号y （未解调）

y\_final = demodulate(modem.pskdemod(M),y); %BPKS解调

err\_num\_AF = err\_num\_AF + Act\_ber(x,y\_final); % wrong number of bits with AF

end

% Calculated the actual BER for each SNR %通过统计蒙特卡罗的误码数，与全部比特数目作对比

ber\_AF(snrcount) = err\_num\_AF/(N\*Monte\_MAX);

% Calculated the theoretical BER for each SNR %调用自定义函数得到

theo\_ber\_AF(snrcount) = Theo\_ber(h\_sd,h\_sr,h\_rd,POW\_S,POW\_N,POW\_S,POW\_N);

end

%% draw BER curves

SNR\_dB = MIN\_SNR\_dB:INTERVAL:MAX\_SNR\_dB;

disp('theo\_ber\_AF=');disp(theo\_ber\_AF);

figure(1) % the actual BER of Direct and AF,DF

semilogy(SNR\_dB,theo\_ber\_AF,'r-o',SNR\_dB,ber\_AF,'b-\*');%semilogx用半对数坐标绘图,x轴是log10，y是线性的；semilogy用半对数坐标绘图,y轴是log10，x是线性的

legend('理论','实际');

grid on; %增加网格

ylabel('The AVERAGE BER');

xlabel('SNR(dB)');

title('the theoretical and actual BER of AF ');

axis([MIN\_SNR\_dB,MAX\_SNR\_dB,10^(-5),1]);

程序仿真图：

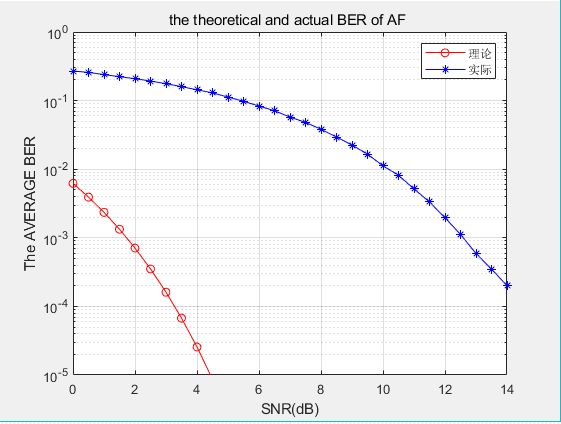


图1 非协作系统AF的理论和实际误码率曲线

多次运行

