**东南大学**

**《协作通信与网络》**

**实验报告**

**论文题目：基于AF和DF的功率分配问题的**

**研究**

**姓 名 ： 谷伟齐**

**学 号 ： 04016538**

**专业班级： 040165班**

**学院名称： 信息科学与工程学院**

**2018年10月**

## 一、实验目的

实验工具：MATLAB R2015a

实验目的：

1. 了解AF和DF协作的工作原理及误码率的计算方法；
2. 学习用MATLAB仿真AF和DF的过程；
3. 研究在源节点和中继节点功率不同时协作效率和误码率的情况并用MATLAB进行仿真。

**二、实验要求**

1、根据原理中的公式和数学表达式编写程序，得到需要的AF和DF协作下在不同源节点和中继节点功率分配比例时的SNR与BER的关系曲线；

2、列出上机的调试程序；

3、进行实验结果的分析和讨论；

4、简述实验心得体会及其他。

**三、实验内容**

1.实验原理

(1)AF协作

在AF协议下，中继节点采用模拟处理，不对接收的信号进行解调和解码，而是直接将收到的带有噪声的信号进行模拟处理，然后发送给接收端。这也就是指中继节点接收到源信号之后直接对模拟信号进行放大处理（使得中继发出信号功率恒定）。

单中继AF协作通信过程：

1. 源端广播过程

源节点S以广播的方式向周围发送信号，其中一路直接发送到目的节点D，另一路发送到中继节点R。经过信道后，则中继节点接收到的信号为:

目的节点D接收到的信号为：

其中，为源发送信号的功率，为源节点与中继节点间的信道噪声。

1. 中继放大过程

中继节点R直接将收到的来自源节点S的信号以系数β进行功率放大，然后转发给目的节点D。AF方式可以看成是具有两个发射端的重复码，唯一不同的是中继节点将自身接收到的噪声信号也放大并发送到目的节点。目的节点通过合并两路信号，对源节点的发送信号进行估计。

为保证中继节点功率受限，放大系数β应满足：

可见β取决于信道的衰落系数、源发送信号的功率和噪声功率。

那么中继放大后的信号：

目的节点接收的来自中继的信号：

其中为中继节点发送信号的功率。

1. 目的端接收处理

目的节点将收到的来自源节点的直传信号和来自中继节点的信号按照MRC方案进行合并，得到信号y：

其中，和分别为目的节点接收到的来自源、来自中继信号的加权系数，为：

式中，和分别为协作情况下源节点和中继节点的发送功率，、、分别为源节点与目的节点、源节点与中继节点、中继节点与目的节点之间的瑞利衰落信道系数的复共轭，为噪声功率。

AF被称作非再生中继方式，其本质上是一种模拟信号的处理方式。在该机制中，每个用户接收它伙伴发送过来的带有噪声的信号，接着对该信号进行放大，然后将放大的带有噪声的信号重新发送。基站将对用户和其伙伴传送来的数据进行合并判决。尽管协作者在进行放大时也放大了噪声，但是基站接收到两个独立的衰落信号最后能作出很好的判决。AF方式最简单，而且由于目的节点可接收到两路独立的衰落信号，AF可获得满分集增益，性能良好。由于中继节点在放大信号的同时也放大了源-中继信道引入的噪声，因此AF方式存在噪声传播效应。

(2)DF协作

DF称为前向译码、再生中继。在DF协议下，中继要先对接收到的信号进行解调、采样判决、存储、译码等数字处理，然后将处理后的数据进行编码调制后再转发，即中继节点接收到信号后先解码再转发。由于DF为再生中继方式，DF方式本质上是一种数字信号处理方式。DF方式通过译码，避免了噪声对下一跳的影响。但是，如果中继节点对所接收到的信号直接进行译码，之后将所译出的信号转发给目的节点，将有可能引起错误传播。这是由于中继节点可能得到错误信息，这样对协作传输反而是不利因素。

单中继DF协作通信过程

1. 源端广播过程

同AF模式一样，源节点S以广播的方式向周围发送信号，其中一路直接发送到目的节点D，一路发送到中继节点R。

1. 中继端解码重传过程

中继节点R直接将接收到的来自源节点S的信号进行解调译码，并通过某种方式校验译码是否正确，如果错误则中继不再发送该信息，如果正确则该信号重新编码调制，然后转发给目的节点D。在编码过程中，可以选择与源节点一样的编码方案，或者采用不同的编码方式。

经过中继重新编码调制的信号为，目的节点接收到的来自中继的信号变为：

1. 目的端接收处理

同AF模式一样，DF协议下的目的节点按照MRC方案的到合并信号y：

不同的是，加权系数取决于两路信号的信噪比最优值：

对于DF模式，中继节点对源节点信息解码错误所带来的误差会随着跳数的增加而不断累积，从而影响到分集效果和中继性能。这表明源-中继节点信道传输特性的好坏对DF方式协同通信系统的性能有很大影响。

2.实验流程

本实验是在上述实验原理的基础上，在AF和DF两种方式下对源节点和中继节点的功率和在不同情况下的配比进行研究，从而找到误码率最小，效率最高的配比。其中，本实验选择了和在0.2：0.8、0.3：0.7、0.4：0.6、0.5：0.5、0.6：0.4、0.7：0.3和0.8：0.2配比情况下在不同信噪比时的BER进行相互比较和与非协作系统情况下的BER进行比较。

3.实验调试程序

(1)AF协作

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%2018.12.8\_Weiqi Gu\_04016538\_SEU

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clc;

clear all;

datestr(now)

MIN\_SNR\_dB = 0;

MAX\_SNR\_dB = 10;

INTERVAL = 0.5; % SNR interval

POW\_S = [0.2 0.3 0.4 0.5 0.6 0.7 0.8]; % Power division factor,with cooperation, in order to guarantee a certain power of the total,

POW\_R=1-POW\_S;

POW = 1;

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

M = 2; % number of symbols

N = 10000; % number of bits

x = randint(1,N,M); % Random binary data stream

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

H\_sd = RayleighCH( 1 ); % between Source and Destination

H\_sr = RayleighCH( 1 ); % between Source and Relay station

H\_rd = RayleighCH( 1 ); % between Relay station and Destination

for num=1:7

snrcount = 0;

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

snrcount = snrcount + 1;

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

err\_num\_AF = 0;

for tries=0:Monte\_MAX

sig = 10^(SNR\_dB/10); % SNR, said non-dB

POW\_S\_N = POW\_S(num) / sig; % Noise power\_S

POW\_R\_N = POW\_R(num) / sig; % Noise power\_R

% 'x\_s' is transmitted from Source to Relay and Destination

y\_sd = awgn( sqrt(POW\_S(num))\*H\_sd \* x\_s, SNR\_dB, 'measured'); % Destination received the signal 'y\_sd' from Source

y\_sr = awgn( sqrt(POW\_R(num))\*H\_sr \* x\_s, SNR\_dB, 'measured'); % Relay received the signal 'y\_sr' from Source

%01:Without Cooperation,Source node transmit the signal to Destination node directly

y\_SD = demodulate(modem.pskdemod(M),H\_sd'\*y\_sd);

err\_num\_SD = err\_num\_SD + Act\_ber(x,y\_SD); % wrong number of bits without Cooperation

%02:With Fixed Amplify-and-Forward relaying protocol

% beta: amplification factor

% x\_AF: Relaytransmit the AF signal 'x\_AF'

[beta,x\_AF] = AF(H\_sr,POW\_S(num),POW\_S\_N,y\_sr);

y\_rd = awgn( sqrt(POW\_R(num))\*H\_rd \* x\_AF, SNR\_dB, 'measured'); % Destination received the signal 'y\_rd' from Relay

y\_combine\_AF = Mrc( H\_sd,H\_sr,H\_rd,beta,POW\_S(num),POW\_S\_N,POW\_R(num),POW\_R\_N,y\_sd,y\_rd); % MRC

y\_AF = demodulate(modem.pskdemod(M),y\_combine\_AF); % After demodulate, Destinationthe gains the signal 'y\_AF'

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

end;% for tries=0:Monte\_MAX

% Calculated the actual BER for each SNR

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

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

end; % for SNR\_dB=MIN\_SNR\_dB:INTERVAL:MAX\_SNR\_dB

end

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

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

semilogy(SNR\_dB,ber\_SD,'-o',SNR\_dB,ber\_AF{1},'-+',SNR\_dB,ber\_AF{2},'-\*',SNR\_dB,ber\_AF{3},'-',SNR\_dB,ber\_AF{4},'--',SNR\_dB,ber\_AF{5},'-^',SNR\_dB,ber\_AF{6},'-.',SNR\_dB,ber\_AF{7},'-<');

legend('Direct','S:R=0.2:0.8','S:R=0.3:0.7','S:R=0.4:0.6','S:R=0.5:0.5','S:R=0.6:0.4','S:R=0.7:0.3','S:R=0.8:0.2');

grid on;

ylabel('The AVERAGE BER');

xlabel('SNR(dB)');

title('the actual BER of Direct and Direct, AF');

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

(2)DF协作

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%2018.12.8\_Weiqi Gu\_04016538\_SEU

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clc;

clear all;

datestr(now)

MIN\_SNR\_dB = 0;

MAX\_SNR\_dB = 10;

INTERVAL = 0.5; % SNR interval

POW\_S = [0.2 0.3 0.4 0.5 0.6 0.7 0.8]; % Power division factor,with cooperation, in order to guarantee a certain power of the total,

POW\_R=1-POW\_S;

POW = 1;

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

M = 2; % number of symbols

N = 10000; % number of bits

x = randint(1,N,M); % Random binary data stream

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

H\_sd = RayleighCH( 1 ); % between Source and Destination

H\_sr = RayleighCH( 1 ); % between Source and Relay station

H\_rd = RayleighCH( 1 ); % between Relay station and Destination

for num=1:7

snrcount = 0;

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

snrcount = snrcount + 1;

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

err\_num\_DF = 0;

for tries=0:Monte\_MAX

sig = 10^(SNR\_dB/10); % SNR, said non-dB

POW\_S\_N = POW\_S(num) / sig; % Noise power\_S

POW\_R\_N = POW\_R(num) / sig; % Noise power\_R

% 'x\_s' is transmitted from Source to Relay and Destination

y\_sd = awgn( sqrt(POW\_S(num))\*H\_sd \* x\_s, SNR\_dB, 'measured'); % Destination received the signal 'y\_sd' from Source

y\_sr = awgn( sqrt(POW\_R(num))\*H\_sr \* x\_s, SNR\_dB, 'measured'); % Relay received the signal 'y\_sr' from Source

%01:Without Cooperation,Source node transmit the signal to Destination node directly

y\_SD = demodulate(modem.pskdemod(M),H\_sd'\*y\_sd);

err\_num\_SD = err\_num\_SD + Act\_ber(x,y\_SD); % wrong number of bits without Cooperation

%03:With Fixed Decode-and-Forward relaying protocol

x\_DF = DF(M,y\_sr,x);

y\_rd = awgn( sqrt(POW\_R(num))\*H\_rd \* x\_DF, SNR\_dB, 'measured');

y\_combine\_DF = Mrc( H\_sd,H\_rd,POW\_S(num),POW\_S\_N,POW\_R(num),POW\_R\_N,y\_sd,y\_rd);

y\_DF = demodulate(modem.pskdemod(M),y\_combine\_DF);

err\_num\_DF = err\_num\_DF + Act\_ber(x,y\_DF); % wrong number of bits with DF

end;% for tries=0:Monte\_MAX

% Calculated the actual BER for each SNR

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

ber\_DF{num}(snrcount) = err\_num\_DF/(N\*Monte\_MAX);

end; % for SNR\_dB=MIN\_SNR\_dB:INTERVAL:MAX\_SNR\_dB

end

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

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

semilogy(SNR\_dB,ber\_SD,'-o',SNR\_dB,ber\_DF{1},'-+',SNR\_dB,ber\_DF{2},'-\*',SNR\_dB,ber\_DF{3},'-',SNR\_dB,ber\_DF{4},'--',SNR\_dB,ber\_DF{5},'-^',SNR\_dB,ber\_DF{6},'-.',SNR\_dB,ber\_DF{7},'-<');

legend('Direct','S:R=0.2:0.8','S:R=0.3:0.7','S:R=0.4:0.6','S:R=0.5:0.5','S:R=0.6:0.4','S:R=0.7:0.3','S:R=0.8:0.2');

grid on;

ylabel('The AVERAGE BER');

xlabel('SNR(dB)');

title('the actual BER of Direct and Direct, DF');

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

## 四、实验结果

1.AF协作

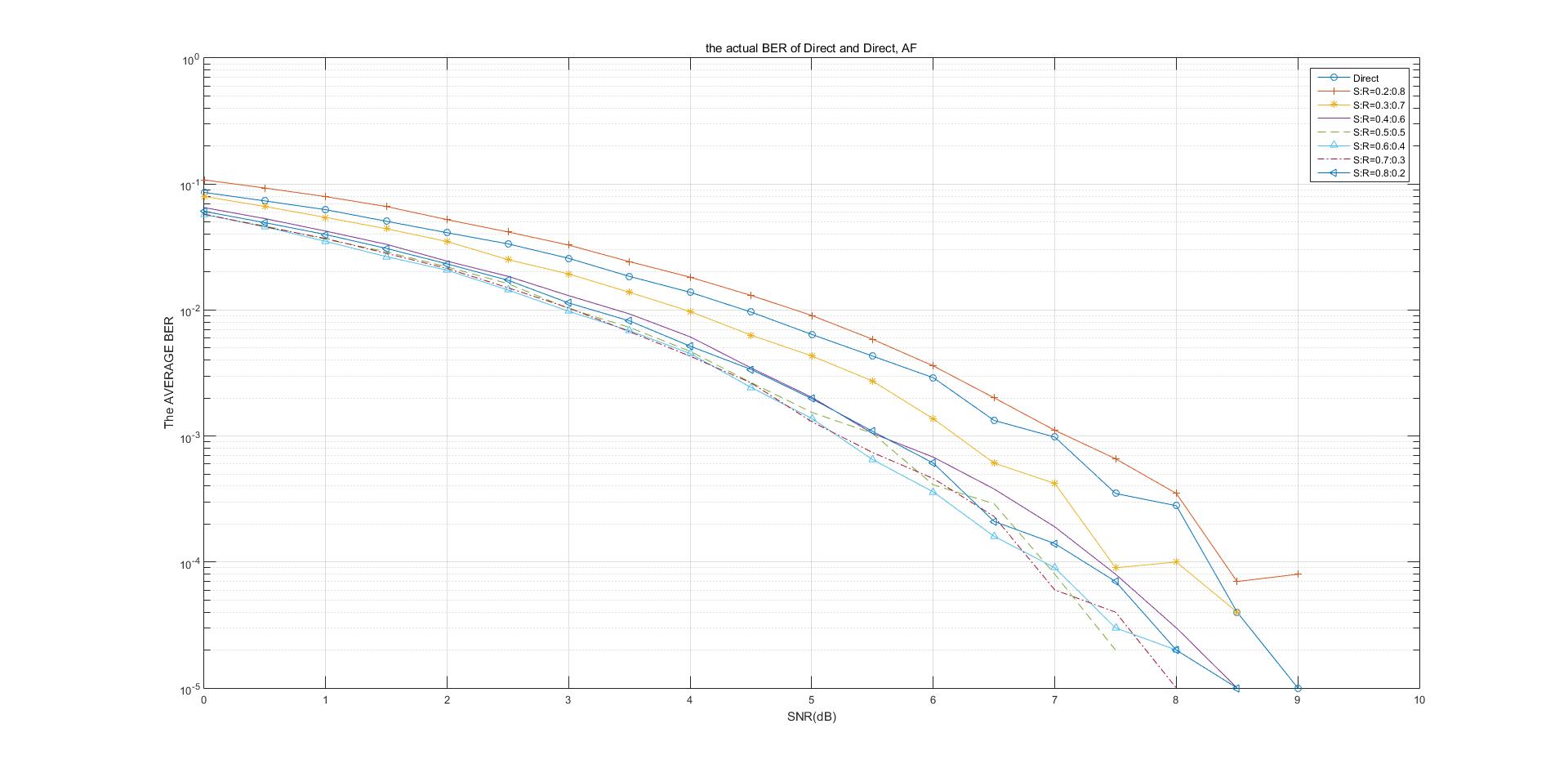


图 1 AF协作

2.DF协作

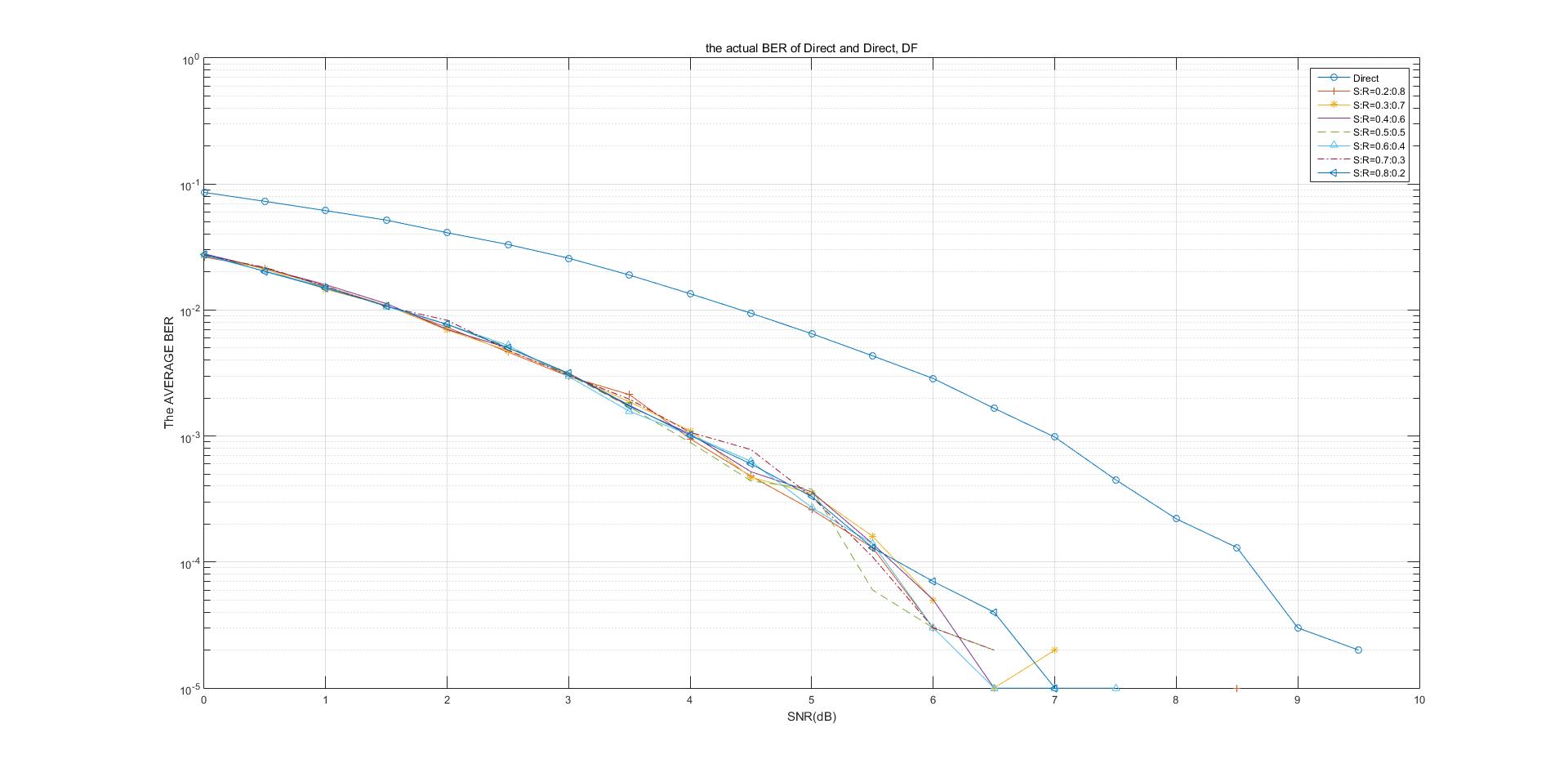


图 2 DF协作

## 五、实验总结

图1所示为AF协作时在源节点和中继节点的功率不同配比情况下SNR与BER的关系图。由图可知，除配比为0.2：0.8之外，其他配比情况均比无协作通信时的误码率要低，传输效率高，且在SNR较小时，配比0.4：0.6、0.5：0.5、0.6：0.4、0.7：0.3、0.8：0.2的情况均很可观且效果差不多，当SNR大于7之后，源节点和中继节点的功率相等的情况（0.5：0.5）的效果较其他几种更为出色。

图2所示为DF协作时在源节点和中继节点的功率不同配比情况下SNR与BER的关系图。由图可知，几种配比情况下的误码率均比无协作通信时的误码率要低很多且互相之间基本相差不多。SNR在5-6的范围时两功率分配相等的效果更好，大于6时可1️以选择源-中继为0.6：0.4的分配方式得到满意的结果。同时我们可以看出，在DF通信中，源节点和中继节点的功率分配并不是影响传输效果的主要因素。

此次实验的不足之处在于程序运行次数还不够多导致结果可能有所偏差，但分配情况的大致结果我们还是能够从实验中得出。