**东南大学**

**《协作通信与网络》**

**实验报告**

**论文题目： 不同合并方案下AF协作方式的BER随信噪比变化的对比**

**姓 名 ： 苏梓恒**

**学 号 ： 04016522**

**专业班级： 040165班**

**学院名称： 信息科学与工程学院**

**2018年12月**

## 一、实验目的

实验工具：MATLAB R2015a

实验目的：在AF协作方式的模型下，改变合并方案为MRC\ERC\FRC\ESNRC，观察BER随信噪比的变化趋势。

熟练操作 MATLAB 软件。

二、实验要求

1、根据数学表达式编写程序，得到需要的关系曲线。

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

运行环境：Matlab R2011b

通信模型：三端点通信模型



程序入口：main1.m

程序结构图：



2.实验程序

1.初始化参数部分：

tic

nr\_of\_iterations=1000; %循环次数为1000次

SNR=[-10:2.5:15];%dB形式

use\_direct\_link=1;%直接传输

use\_relay=1;%通过伙伴传输

P=2;%总功率为2

global signal;%发送信号的全局变量

signal=generate\_signal\_structure;%发送信号结构体，包括传输比特数、传输字符数、每个字符包含的比特数、信号的调制方式、发送信号的比特序列、

%发送信号的字符序列,传输完毕后接收到的比特序列,用来求误码率.

signal(1).modulation\_type='QPSK';%信号调制类型

signal.nr\_of\_bits=2^10;%传输比特数

signal.position\_x=-1;%信号源所处位置横坐标

signal.position\_y=0;%信号源所处位置纵坐标

calculate\_signal\_parameter;%发送信号准备完毕

channel=generate\_channel\_structure;

channel(1).attenuation(1).pattern='Rayleigh'; %'no','Rayleigh'

channel.attenuation(1).block\_length=1;%块长(bit/block)

channel(2)=channel(1);

channel(3)=channel(1);%三信道模式相同

channel(1).attenuation.distance=1; %信号传输距离，源与目的

channel(2).attenuation.distance=0.75;%中继与源

channel(3).attenuation.distance=0.25;%中继与目的

rx=generate\_rx\_structure;%为接收参数创建结构体

rx(1).combining\_type='ESNRC';%'ERC','FRC','SNRC','ESNRC','MRC'合并方案

rx.rx\_position\_x=1;

rx.rx\_position\_y=0;%位置

rx.sd\_weight=2;%加权值为2

rx(2)=rx(1);%负责没有协同情况下的接收

global relay;

relay=generate\_relay\_structure;%为relay创建结构体

relay(1).mode='AAF';

relay.magic\_genie=0;

relay.rx=rx(1); %relay的接收部分为rx（1）

h = waitbar(0,'Please wait...');%打开或更新等待条对话框

BER=zeros(size(SNR));%比特出错概率初始化

BER\_1sender=zeros(size(SNR));

2.功率分配部分

%功率分配 [Ps,Pr]=power\_allocate(P,asd,asr,ard,'ESNRC','Best','AAF',rx(1).sd\_weight);%转换模式为AAF，以及ESNRC的合并模式

功率分配函数：

function [Ps,Pr]=power\_allocate(P,asd,asr,ard,combining\_type,al\_method,transMode,sd\_weight);

switch transMode

case 'AAF'

switch combining\_type %合并方式

case 'MRC'

for(i=1:1:size(asd,2))

switch al\_method

case 'Best'

if(((asd(i)\*ard(i)+asr(i)\*ard(i)-asd(i)\*asr(i))>0)&&((asr(i)\*ard(i)\*P-asr(i)\*asd(i)\*P-asd(i))>0))

Ps(i)=(asr(i)\*ard(i)\*P+asd(i)\*ard(i)\*P+asd(i))/(asd(i)\*ard(i)+asr(i)\*ard(i)-asd(i)\*asr(i)+sqrt((asd(i)\*ard(i)+asr(i)\*ard(i)-asd(i)\*asr(i))\*asr(i)\*ard(i)\*(asr(i)\*P+1)/(ard(i)\*P+1)));

Pr(i)=(asr(i)\*ard(i)\*P-asr(i)\*asd(i)\*P-asd(i))/(asd(i)\*ard(i)+asr(i)\*ard(i)-asd(i)\*asr(i)+sqrt((asd(i)\*ard(i)+asr(i)\*ard(i)-asd(i)\*asr(i))\*asr(i)\*ard(i)\*(ard(i)\*P+1)/(asr(i)\*P+1)));

else

Ps(i)=P/2;

Pr(i)=P/2;

end

case 'equal'

Ps(i)=P/2;

Pr(i)=P/2;

case 'onlys'

Ps(i)=P;

Pr(i)=0;

otherwise

error('No such power allocate method');

end

end

case 'ERC'

for(i=1:1:size(asd,2))

switch al\_method

case 'Best'

if(((asd(i)-ard(i))\*(asd(i)-2\*asr(i))>0)&&(2\*asr(i)\*ard(i)\*P-2\*asd(i)-2\*asd(i)\*asr(i)\*P-asd(i)\*ard(i)\*P>0))

Ps(i)=(asr(i)\*ard(i)\*P+asd(i))/(asr(i)\*ard(i)-asr(i)\*asd(i)+sqrt((asr(i)\*P+1)\*(asd(i)-ard(i))\*(asd(i)-2\*asr(i))\*asr(i)\*ard(i)/(ard(i)\*P+2)));

Pr(i)=(2\*asr(i)\*ard(i)\*P-2\*asd(i)-2\*asd(i)\*asr(i)\*P-asd(i)\*ard(i)\*P)/(2\*asr(i)\*ard(i)-2\*asr(i)\*asd(i)+sqrt((ard(i)\*P+2)\*(asd(i)-ard(i))\*(asd(i)-2\*asr(i))\*ard(i)\*asr(i)/(asr(i)\*P+1)));

else

Ps(i)=P/2;

Pr(i)=P/2;

end

case 'equal'

Ps(i)=P/2;

Pr(i)=P/2;

case 'onlys'

Ps(i)=P;

Pr(i)=0;

otherwise

error('No such power allocate method');

end

end

case 'FRC'

for(i=1:1:size(asd,2))

switch al\_method

case 'Best'

if(((asd(i)\*sd\_weight^2-ard(i))\*(asd(i)\*sd\_weight^2-sd\_weight^2\*asr(i)-asr(i))>0)&&((P\*asr(i)\*ard(i)-asd(i)\*sd\_weight^2)\*(sd\_weight^2+1)-P\*sd\_weight^2\*asd(i)\*(asr(i)\*sd\_weight^2+asr(i)+ard(i))>0))

Ps(i)=(asr(i)\*ard(i)\*P+sd\_weight^2\*asd(i))/(asr(i)\*ard(i)-asr(i)\*sd\_weight^2\*asd(i)+sqrt((asr(i)\*P+1)\*(asd(i)\*sd\_weight^2-ard(i))\*(sd\_weight^2\*asd(i)-sd\_weight^2\*asr(i)-asr(i))\*asr(i)\*ard(i)/(ard(i)\*P+sd\_weight^2+1)));

Pr(i)=((P\*asr(i)\*ard(i)-asd(i)\*sd\_weight^2)\*(sd\_weight^2+1)-P\*sd\_weight^2\*asd(i)\*(asr(i)\*sd\_weight^2+asr(i)+ard(i)))/((asr(i)\*ard(i)-asd(i)\*asr(i)\*sd\_weight^2)\*(sd\_weight^2+1)+sqrt((asd(i)\*asr(i)\*sd\_weight^2-asr(i)\*ard(i))\*(ard(i)\*P+sd\_weight^2+1)\*(sd\_weight^2\*asd(i)-sd\_weight^2\*asr(i)-asr(i))\*ard(i)/(asr(i)\*P+1)));

else

Ps(i)=P/2;

Pr(i)=P/2;

end

case 'equal'

Ps(i)=P/2;

Pr(i)=P/2;

case 'onlys'

Ps(i)=P;

Pr(i)=0;

otherwise

error('No such power allocate method');

end

end

case 'ESNRC'

for(i=1:1:size(asd,2))

switch al\_method

case 'Best'

if(((asd(i)-ard(i))\*(asd(i)-2\*asr(i))>0)&&(2\*asr(i)\*ard(i)\*P-2\*asd(i)-2\*asd(i)\*asr(i)\*P-asd(i)\*ard(i)\*P>0))

basic\_ps(i)=(asr(i)\*ard(i)\*P+asd(i))/(asr(i)\*ard(i)-asr(i)\*asd(i)+sqrt((asr(i)\*P+1)\*(asd(i)-ard(i))\*(asd(i)-2\*asr(i))\*asr(i)\*ard(i)/(ard(i)\*P+2)));

basic\_pr(i)=(2\*asr(i)\*ard(i)\*P-2\*asd(i)-2\*asd(i)\*asr(i)\*P-asd(i)\*ard(i)\*P)/(2\*asr(i)\*ard(i)-2\*asr(i)\*asd(i)+sqrt((ard(i)\*P+2)\*(asd(i)-ard(i))\*(asd(i)-2\*asr(i))\*ard(i)\*asr(i)/(asr(i)\*P+1)));

if(0.1<asd(i)\*(asr(i)\*basic\_ps(i)+ard(i)\*basic\_pr(i)+1)/(basic\_pr(i)\*asr(i)\*ard(i))&&(asd(i)\*(asr(i)\*basic\_ps(i)+ard(i)\*basic\_pr(i)+1)/(asr(i)\*ard(i)\*basic\_pr(i))<10))

Ps(i)=(asr(i)\*ard(i)\*P+asd(i))/(asr(i)\*ard(i)-asr(i)\*asd(i)+sqrt((asr(i)\*P+1)\*(asd(i)-ard(i))\*(asd(i)-2\*asr(i))\*asr(i)\*ard(i)/(ard(i)\*P+2)));

Pr(i)=(2\*asr(i)\*ard(i)\*P-2\*asd(i)-2\*asd(i)\*asr(i)\*P-asd(i)\*ard(i)\*P)/(2\*asr(i)\*ard(i)-2\*asr(i)\*asd(i)+sqrt((ard(i)\*P+2)\*(asd(i)-ard(i))\*(asd(i)-2\*asr(i))\*ard(i)\*asr(i)/(asr(i)\*P+1)));

else

Ps(i)=P/2;

Pr(i)=P/2;

end

else

Ps(i)=P/2;

Pr(i)=P/2;

end

case 'equal'

Ps(i)=P/2;

Pr(i)=P/2;

case 'onlys'

Ps(i)=P;

Pr(i)=0;

otherwise

error('No such power allocate method');

end

end

otherwise

error(['Unknown relay-type:',combining\_type]);

end

case 'DAF'

for(i=1:1:size(asd,2))

Ps(i)=P/2;

Pr(i)=P/2;

end

otherwise

error('No such tran Mode');

end

3.实验仿真结果图：

1.ESNRC模式下：

2.MRC



3.ERC



4.FRC



4.四种方式对比图：



3.实验结论;

可看出，在相同的信噪比下，仅改变模型的合并方式，MRC方式的误码率最低，即此方案在此次实验的对比之下为最优。