**东南大学**

**《协作通信与网络》**

**实验报告**

**论文题目： 三种合并方式（SC,EGC,MRC）**

**对AF，DF协作性能的影响**

**姓 名 ： 唐全意**

**学 号 ： 04016327**

**专业班级： 信息工程（3）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 R2017b

实验目的：

1. 了解掌握AF与DF协作的基本原理以及程序实现。
2. 掌握三种三种合并方式（SC,EGC,MRC）的原理及实现公式。
3. 通过AF与DF代码实现检验三种不同的合并方式对协作性能产生的影响。

## 二、实验原理

**1.AF简介：**

在AF协作下，中继采用模拟处理，不对接收的信号进行解调和解码，而是直接对接收到的带有噪声的信号进行放大，然后发送给接收端。

**2.AF理论公式介绍及核心代码实现：**

1．源端广播过程

源节点S以广播的方式向周围发送信号（在程序中表示为x\_s），其中一路直接发送到目的节点D，一路发送到中继节点R。经过信道后，则

中继节点R接收到的信号为（程序中表示为y\_sr）：

式(1-1-1)

目的节点D接收到的信号为（程序中表示为y\_sd）：

式(1-1-2)

其中，源发送的信号的功率为，为源节点与中继节点间信道噪声。

代码实现：

y\_sd =awgn( sqrt(POW\_DIV)\*H\_sd\*x\_s,SNR\_dB,'measured');

y\_sr =awgn( sqrt(POW\_DIV)\*H\_sr\*x\_s,SNR\_dB,'measured');

2．中继端放大转发过程

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（基站）。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：

式(1-1-3)

可见*β*取决于信道的衰落系数、源发送的信号的功率和噪声功率。

那么，中继放大后的信号为（程序中表示为x\_AF）：

式(1-1-4)

目的节点接收的来自中继的信号为（程序中表示为y\_rd）：

式(1-1-5)

其中为中继节点发送的信号的功率。

代码实现：

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

y\_rd = awgn( sqrt(POW\_S)\*H\_rd \* x\_AF,SNR\_dB,'measured');

function [beta,signal\_AF] =AF(CH\_sr,POW\_S,POW\_N,signal\_sr)

beta = sqrt( POW\_S) / ( (abs(CH\_sr))^2 \* POW\_S + POW\_N );

signal\_AF = beta \* signal\_sr;

3．目的端接收处理

目的节点将接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（程序中表示为y\_combine\_AF）：

式(1-1-6)

其中、分别为目的节点接收到的来自源、来自中继的信号的加权系数，为：

式(1-1-7)

式(1-1-8)

式中，和分别为协作情况下源节点和中继节点的发送功率，、和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。

代码实现：

a\_sd = CH\_sd' \* sqrt(POW\_S\_sd) / POW\_N\_sd;

a\_rd = (beta \* sqrt(POW\_S\_rd) \* CH\_sr' \* CH\_rd') / ( (beta^2\*(abs(CH\_rd))^2+1) \* POW\_N\_rd );

**3.DF简介：**

在DF协作下，中继先要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。

**4.DF理论公式介绍及核心代码实现：**

1．源端广播过程

公式及代码实现均与AF模式相同，在此不重复介绍。

2．中继端解码重传过程

中继节点R直接将收到的来自源节点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号，如果正确，则将该信号重新编码调制，然后转发给目的节点D（基站）。

经过中继重新编码调制信号为（程序中表示为x\_DF），目的节点接收的来自中继的信号变为（程序中表示为y\_rd）：

式(1-2-1)

代码实现：

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

3．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方案得到合并信号*y*（程序中表示为y\_combine\_DF）：

式(1-2-2)

不同的是，加权系数、取决于两路信号的信噪比最优值：

式(1-2-3)

式(1-2-4)

代码实现：

a\_sd = CH\_sd' \* sqrt(POW\_S\_sd) / POW\_N\_sd;

a\_rd = CH\_rd' \* sqrt(POW\_S\_rd) / POW\_N\_rd;

**5.SC简介及其核心代码实现**

选择式合并(SC)：

选择式合并是所有合并方式中最简单的一种，其原理是检测所有接收机输出

信号的信噪比，选择其中信噪比最大的那一路作为合并器的输出

代码实现：

y\_AF\_sd = demodulate(modem.pskdemod(M),y\_sd);

y\_AF\_rd = demodulate(modem.pskdemod(M),y\_rd);

if Act\_ber(x,y\_AF\_sd)>Act\_ber(x,y\_AF\_rd)

err\_num\_AF = err\_num\_AF + Act\_ber(x,y\_AF\_rd); else

err\_num\_AF = err\_num\_AF + Act\_ber(x,y\_AF\_sd);

end

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

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

if Act\_ber(x,y\_DF\_sd)>Act\_ber(x,y\_DF\_rd)

err\_num\_DF = err\_num\_DF + Act\_ber(x,y\_DF\_rd); else

err\_num\_DF = err\_num\_DF + Act\_ber(x,y\_DF\_sd);

End

1. **EGC简介及其核心代码实现**

等增益合并(EGC) :

当接收机收到的各支路信号的加权系数 k1= k2 =…= kN时，即为等增益合并。

代码实现：

y\_combine\_AF=y\_sd+y\_rd;

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

err\_num\_AF = err\_num\_AF + Act\_ber(x,y\_AF);

y\_combine\_DF=y\_sd+y\_rd;

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

err\_num\_DF = err\_num\_DF + Act\_ber(x,y\_DF);

1. **MRC简介及其核心代码实现**

最大比合并(MRC)：

最大比合并原理是各支路加权系数与该支路信噪比成正比。信噪比越大，加权系数越大，对合并后信号贡献越大。若每条支路的平均噪声功率是相等的，可以证明，当各支路加权系数为 时，分集合并后的平均输出信噪比最大。式中，Ak为第 k 条支路信号幅度， 为每条支路噪声平均功率。

代码实现： y\_combine\_AF=Mrc( H\_sd,H\_sr,H\_rd,beta,POW\_S,POW\_N,POW\_S,POW\_N,y\_sd,y\_rd);;

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

err\_num\_AF = err\_num\_AF + Act\_ber(x,y\_AF);

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

function signal\_combine = Mrc( varargin )

if nargin == 8 % DF

CH\_sd = varargin{1};

CH\_rd = varargin{2};

POW\_S\_sd = varargin{3};

POW\_N\_sd = varargin{4};

POW\_S\_rd = varargin{5};

POW\_N\_rd = varargin{6};

signal\_sd = varargin{7};

signal\_rd = varargin{8};

% a\_sd and a\_rd are determined such that the SNR of the MRC output is maximized

a\_sd = CH\_sd' \* sqrt(POW\_S\_sd) / POW\_N\_sd;

a\_rd = CH\_rd' \* sqrt(POW\_S\_rd) / POW\_N\_rd;

elseif nargin == 10 % AF

CH\_sd = varargin{1};

CH\_sr = varargin{2};

CH\_rd = varargin{3};

beta = varargin{4};

POW\_S\_sd = varargin{5};

POW\_N\_sd = varargin{6};

POW\_S\_rd = varargin{7};

POW\_N\_rd = varargin{8};

signal\_sd = varargin{9};

signal\_rd = varargin{10};

% a\_sd and a\_rd are determined such that the SNR of the MRC output is maximized

a\_sd = CH\_sd' \* sqrt(POW\_S\_sd) / POW\_N\_sd;

a\_rd = (beta \* sqrt(POW\_S\_rd) \* CH\_sr' \* CH\_rd') / ( (beta^2\*(abs(CH\_rd))^2+1) \* POW\_N\_rd );

end

%% output after MRC

signal\_combine = a\_sd\*signal\_sd + a\_rd\*signal\_rd;

## 实验结果

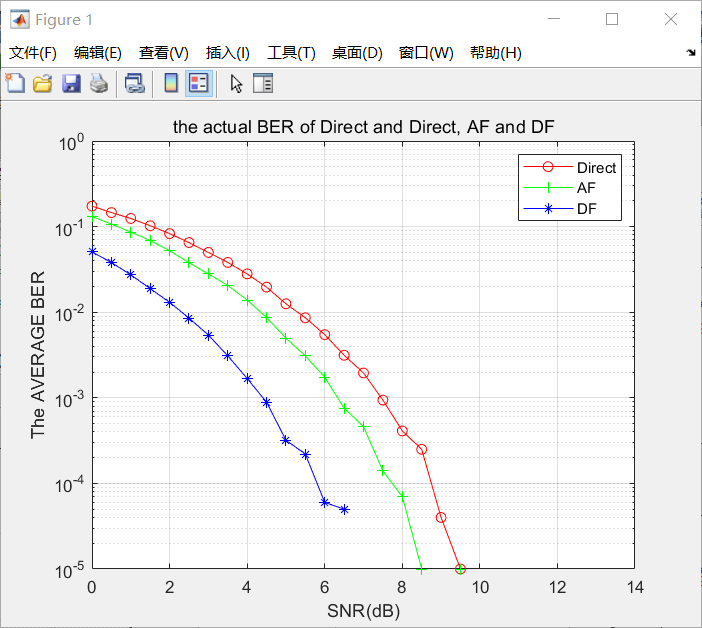


图1：最大比合并情况下非协作系统与AF、DF的实际误码率曲线

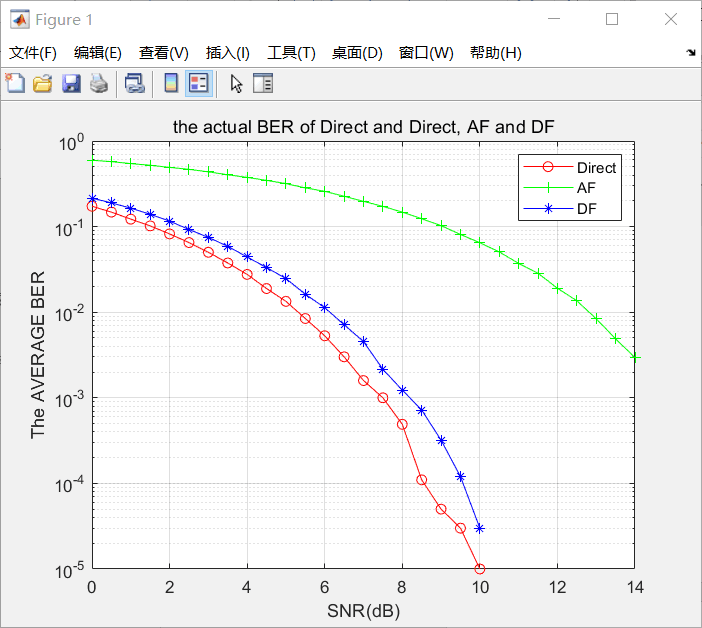


图2：等增益合并情况下非协作系统与AF、DF的实际误码率曲线

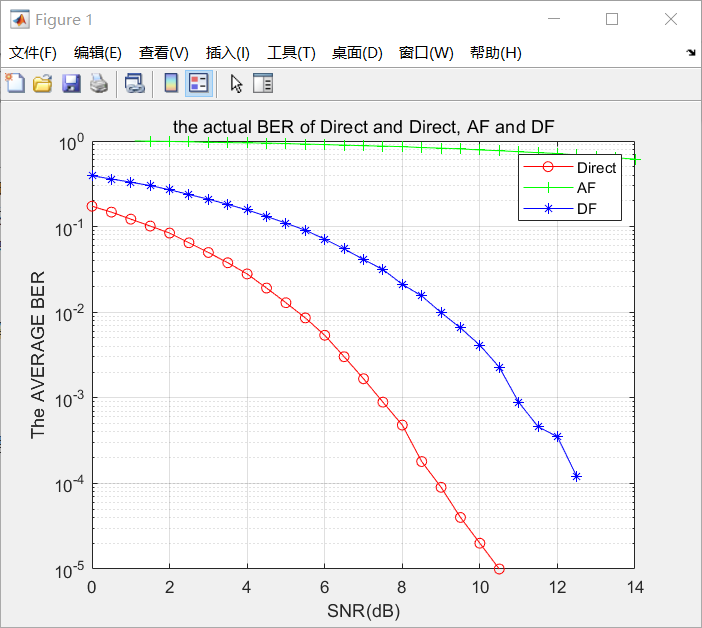


图3：选择式合并情况下非协作系统与AF、DF的实际误码率曲线

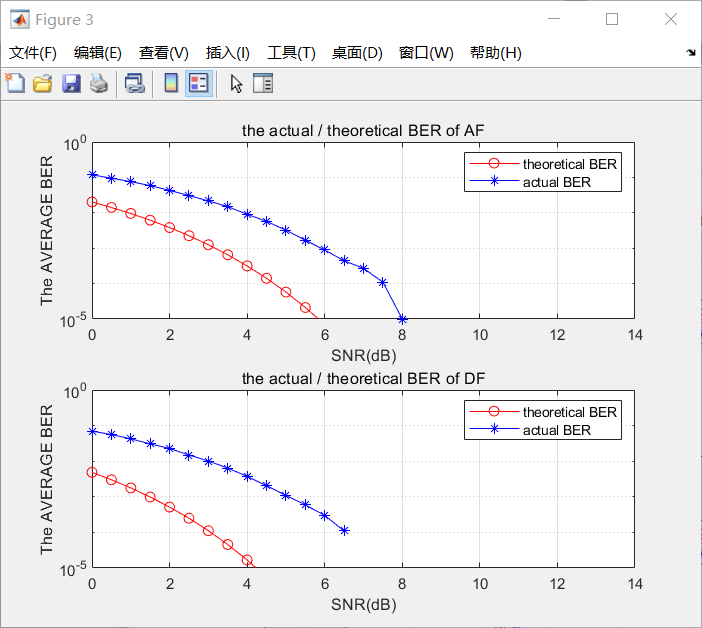


图4：最大比合并情况下AF、DF的实际与理论误码率比较

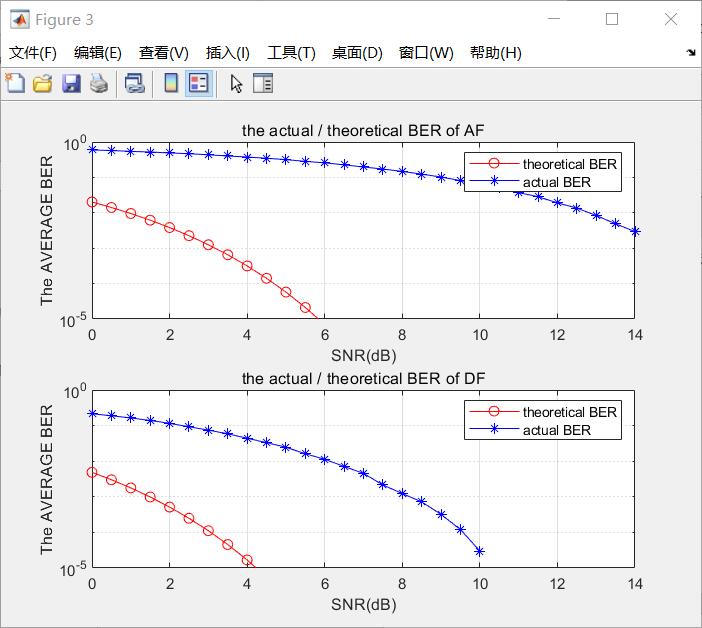


图5：等增益合并情况下AF、DF的实际与理论误码率比较

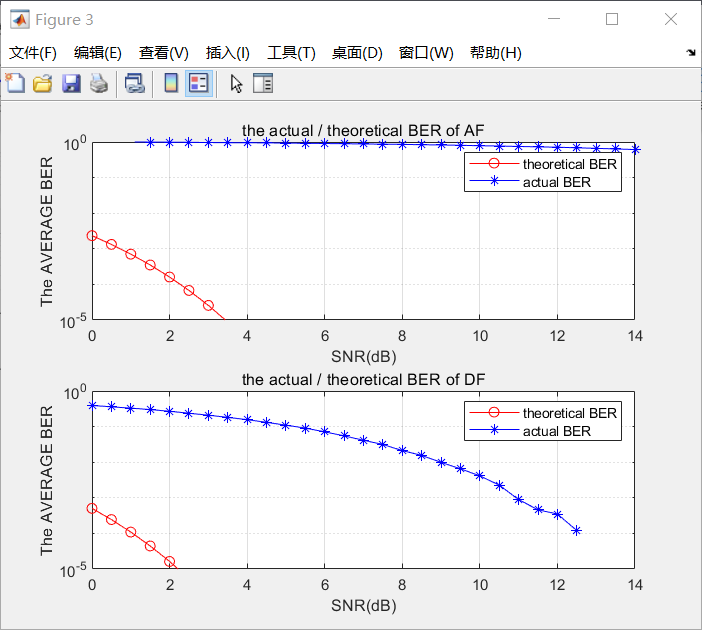


图6：选择式合并情况下AF、DF的实际与理论误码率比较

## 四、实验总结

对比图一，图二，图三可以明显的得到结论，即在对协作通信的性能影响方面，最大比合并最优，等增益合并次之，而选择式合并差。

对比图四，图五，图六可以得出类似的结论，也就是选择式合并造成的实际误码率最高，等增益合并造成的实际误码率次之，而最大比合并造成的实际误码率最小。

单独看图一，图二，图三可以看出，并不是所有情况下协作的性能都优于非协作。在仿真实验中，性能最优的最大比合并模式下，协作的性能都是优于非协作的，但是在性能稍次的等增益合并与选择式合并的情况下，由于协作的复杂通信过程，给仿真通信效果带来的负面影响，已经大于了其所能带来的误码率降低的优势，因此，便出现了等增益合并以及选择式合并的情况下，协作的性能低于非协作性能的情况。

通过这次实验，加深了对AF，DF协作通信，以及协作通信三种合并方式的理解。当然也有一些值得改进的不足之处，比如代码的稳定性等方面，还有进一步研究优化的空间。