**东南大学**

**《协作通信与网络》**

**实验报告**

**论文题目： AF-DF仿真及性能分析**

**姓 名 ： 孙雅伦**

**学 号 ： 61516309**

**专业班级： 615163**

**学院名称： 吴健雄学院**

**2018年12月**

**一、实验目的**

实验工具：Matlab2015b

实验目的：学习AF和DF两种协作方式的原理；

通过仿真来比较AF、DF和不使用协作通信的性能差异；

在不同的功率分配情况下，研究三种方式的性能变化。

**二、实验要求**

1.根据对AF、DF的学习和理解，阐述实现的具体原理，并画出实现框图。

2.对AF、DF和不使用协作通信这三种情况建模，使用Matlab进行仿真。

3.对仿真结果进行分析，总结AF、DF和不使用协作通信的差别。

4.改变协作通信时的功率分配，分析结果。

**三、实验内容**

实验原理：

**1.AF**

放大转发模式（Amplify-and-Forward，AF），有文献中也称为前向放大、非再生中继。在AF协议下，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。图1.1所示为单中继AF协议的原理及其基本过程。

放大

中继R

Hr\_d

Hs\_r

Hs\_d

目的D

源S

图1.1 单中继AF协作通信原理

AF模式是最简单的一种中继方式，相比于直传方式，AF协作通信总能带来一定的信噪比增益，但是由于中继节点除了转发源节点的有用信息外，引入的噪声也同样被放大转发，因此，当源节点与中继节点之间的链路信道条件较差时，信号几乎淹没在噪声之中，因而转发的大部分为噪声，从而会降低目的节点的接收性能。

1.1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，为源发送的信号的功率，为源节点与中继节点间信道噪声。

1.2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。

1.3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。

**2.DF**

解码重传模式（Decode-and-Forward，DF），有文献中也称为前向译码、再生中继。在DF协议下，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。图1.2所示为DF协议的原理及其基本过程。

解码

中继R

Hs\_r

Hr\_d

Hs\_d

目的D

源S

图1.2 单中继DF协作通信原理

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的是错误信息，这样对协作传输反而是不利因素。

2.1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。

2.2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）。在编码过程中，可以选择与源节点一样的编码方案，或者采取不同的编码方式。

有两种简单办法可以降低错误解码所带来的不利影响影响：

第一种方法是信号在源节点发射之前先进行循环冗余校验(CRC)码处理。这样，中继节点接收到源节点的信息后先进行译码处理，之后通过CRC来判别接收到的信息比特里是否存在错误。如果检测出错误，则不进行信息转发；反之则转发信号。但是，CRC的引入将降低了信息的传输速率。

第二种方法不需要对源信号进行CRC编码操作，只需在每个中继节点处设定一个门限值。在对接收信号译码处理之前，先比较它的等效信噪比与门限值的大小。如果大于门限值，中继节点将进行译码处理，并进行信息转发；反之不对信号处理。基于门限的方法虽然简便，但是门限值的选择至关重要。如果太小，中继节点译出的信息很可能存在错误；如果太大，每个中继节点可能都不会进行信息转发，这样协作将失去意义。另外，即便是等效信噪比大于门限值，也并不能保证中继节点译出信息的一定正确。

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)

2.3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)

**3. 理论误码率性能**

假定所传输的信号功率归一化，采用MRC合并方式，则此单中继模型下的AF、DF协作通信系统的理论信噪比为：

式(1-3-1)

其中，对于AF模式：

式(1-3-2)

和

式(1-3-3)

对于DF模式：

式(1-3-4)

和

式(1-3-5)

对于BPSK系统，当发送“1”符号和发送“0”符号概率相等时，最佳判决门限b\*=0，则系统的总误码率为：

式(1-3-6)

其中，为符号“0”被判为符号“1”的概率。

在大信噪比（）条件下，式(3-3-4)可近似表示为：

式(1-3-7)

可见，无论是AF还是DF协作通信系统通过中继产生分集，其信噪比大于非协作通信系统信噪比（），因而其误码率得以降低。

**建模及仿真：**

**1.整体的流程图**

在生成信号并进行BPSK调制后，生成信道参数，之后对直传、AF和DF方式采用并行顺序仿真并获得理论与实际BER，这样做的目的是为了能在相同的信道下比较三者的性能，从而使结果更具可比性。流程图见图2.1

中继放大转发

不使用中继

AF

DF

结束

BER分析

BER分析

BER分析

信号直发

BPSK解调

目的MRC

BPSK解调

BPSK解调

目的MRC

中继解码重传

中继选择

瑞利信道

BPSK调制

生成信号

图2.1 程序流程图

**2.AF实现方式**

在中继节点，对接收到的来自源节点的信号s\_r放大，并向目的节点转发

如图2.2

源发送x\_s

BER分析

BPSK解调

瑞利衰落H\_sd和白噪声

MRC合并

瑞利衰落H\_rd和白噪声

中继放大转发系数β

瑞利衰落H\_sr和白噪声

图2.2 AF流程图

**3.DF实现方式**

在中继节点，对接收到的来自源节点的信号s\_r进行解调、译码、校验、编码、调制，之后向目的节点发送。为简化程序，程序中省略了信道编码过程，中继时采用与源节点相同的调制方式，并假定中继节点能正确解码，强制校验标志tx\_coop为1。所以，实际上，在仿真中，DF中继节点所需做的事情同源节点一样，仅仅是将原始信息BPSK调制后发送。

流程图见图2.3

瑞利衰落H\_sd和白噪声

BER分析

BPSK解调

MRC合并

瑞利衰落H\_rd和白噪声

中继编码

源节点满功率直发

译码对错

中继解码

瑞利衰落H\_sr和白噪声

源发送x\_s

图2.3 DF流程图

**4.代码分析**

**AF：**

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 )); % amplification factor保证中继节点功率受限

signal\_AF = beta \* signal\_sr; % Relay transmit the AF signal 'signal\_A

原本的代码为：

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 )); % amplification factor保证中继节点功率受限

signal\_AF = beta \* signal\_sr; % Relay transmit the AF signal 'signal\_A

原本的代码β的公式代入错误， ，sqrt应该将后面的式子都放在括号里。

**初始化：**

M = 2; % number of symbols

N = 10000; % number of bits

x = randi([0,M-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)]

原代码为：

M = 2; % number of symbols

N = 10000; % number of bits

x = randi(1,N,M); % 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)]

原代码使用randi不正确，randi(1,N,M)只能生成N×M的全为1的矩阵，而randi([0,M-1],1,N)可以生成1×N的由[0,M-1]随机组成的矩阵。

**功率分配：**

将POW\_DIV设为1/3，在使用中继时，则分配给中继的功率为1/3，而分配给目的节点的功率为2/3。

POW\_DIV = 1/3; % Power division factor,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

在蒙特卡罗循环中也要对AF、DF的MRC合并进行修改，以及计算理论BER值也要修改功率。

for tries=0:Monte\_MAX

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

POW\_S = POW\_DIV;% Signal power

POW\_S1=1-POW\_S;

POW\_N = POW\_S / sig; % Noise power

POW\_N1=POW\_S1/sig;

[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'); % Destination received the signal 'y\_rd' from Relay

y\_combine\_AF = Mrc( H\_sd,H\_sr,H\_rd,beta,POW\_S1,POW\_N1,POW\_S,POW\_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

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

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

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

y\_combine\_DF = Mrc( H\_sd,H\_rd,POW\_S1,POW\_N1,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); % wrong number of bits with DF

end;% for tries=0:Monte\_MAX

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

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

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

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

theo\_ber\_SD(snrcount) = Theo\_ber(SNR\_dB);

theo\_ber\_AF(snrcount) = Theo\_ber(H\_sd,H\_sr,H\_rd,POW\_S1,POW\_N1,POW\_S,POW\_N);

theo\_ber\_DF(snrcount) = Theo\_ber(H\_sd,H\_rd,POW\_S1,POW\_N1,POW\_S,POW\_N);

**四、实验结果**

图3.1和图3.2分别是源与中继间信道状态较好、假定DF解码正确的情况下的非协作系统与AF、DF的实际误码率曲线和理论误码率曲线图，图3.3是此时的AF、DF的实际与理论误码率比较图。这三张图都是建立在POW\_DIV=1/2的基础上，也就说中继分配的功率和分配给目标的功率是相同的。

从图3.1和图3.2可以看到，采用AF中继和DF中继的协作通信系统，其误码率明显低于非协作系统。并且，在一定范围内，随着信噪比的提升，协作中继对系统性能的提升越明显，但是，当信噪比达到一定程度后，这种优势将不复存在。

同时可以看出，DF中继能要优于AF中继。这是由于AF将噪声也同样放大转发，而DF中继通过译码、校验等数字处理，避免了噪声对下一链路的影响。并且若源节点与中继节点之间的信道条件较差，转发的信号的大部分为噪声，从而会降低系统性能。

然而，这并非意味着在系统中采用DF中继一定会比AF中继更佳。这主要由于：一方面，DF是数字处理，设备复杂度较高，不适合应用于低成本设备；另一方面，若要获得良好的性能，DF需要相应的校验信息，这在一定程度上降低了数据的传输速率；此外，当信噪比较低时，中继节点译码错误的可能性增加，从而协作失败，或是“错上加错”的编码重传，导致系统的性能急剧恶化，误码率甚至可能会高于非协作系统。仿真中，假设DF中继完全解码正确，从而获得较好的性能。



图3.1 非协作系统与AF、DF（解码正确）的实际误码率曲线



图3.2 非协作系统与AF、DF的理论误码率曲线



图3.3 AF、DF的实际与理论误码率比较

调节功率分配POW\_DIV，当POW\_DIV大于1/2时，如2/3、3/4、4/5等等，使用AD、DF和不使用中继这三种情况的仿真图仍然符合上述的规律，但是当POW\_DIV小于1/2时，AF、DF的性能就会有所下降，由于图片较多，这里只截取了表现明显的POW\_DIV=1/5、1/7、1/8时的实际误码率曲线仿真图。如图3.4、图3.5、图3.6。



图3.4 POW\_DIV=1/5时的实际误码率曲线



图3.5 POW\_DIV=1/7时的实际误码率曲线



图3.6 POW\_DIV=1/8时的实际误码率曲线

**总结：**

功率分配问题在协作通信中十分重要，考虑到现实的功率利用效率，以及实际信道会有很多干扰，在现实中是不会给中继和目标节点平均分配功率的，如何实现最优的分配是人们一直在研究的问题。在仿真时，也能够感觉出来不同功率分配下，AF、DF和不使用中继的性能会有变化，但是由于仿真仍然是较为理想的情况，所以不是十分明显。

通过这次实验，让我更加认识到使用协作通信的优势，以及协作通行还存在的不足，在实际应用时，协作通信想要达到优异的性能，还受到许多限制，不过，不管怎样，一个质量良好的信道是十分重要的。