

# 东南大学

## 《协作通信与网络》

### 实验报告

论文题目：基于误码率的中继协作通信中  
功率分配的研究

姓 名：张鹏辉

学 号：04016632

专业班级：040166

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 R2015b

实验目的：针对单中继 AF 协作通信系统以及 DF 协作通信系统，分析了不同中继位置和功率分配方式对系统误码率性能的影响。以最小化系统误码率为目标，研究了等功率分配(equal power allocation, EPA)和最优功率分配(optimum power allocation, OPA)两种方案下，中继按照线形拓扑轨迹运动时的最优位置。理论分析和仿真结果表明，中继位置及 OPA 的联合优化算法大大提高了传统的 EPA 方案下的系统性能。

## 二、实验要求

- 1、根据本文所列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讨论。
- 4、简述实验心得体会及其他。

### 三、实验内容

#### 3.1 实验原理

协作中继的基本模型是：一个源节点（S），一个目的节点（D），若干个中继节点 R。假设系统发射的总功率为  $P$ ，且  $P=P_s+P_r$ ，其中  $P_s$  和  $P_r$  分别表示分配给源 S 和中继 r 的发射功率。

##### 3.1.1 AF 协作模式基本原理

在 AF 协议下，中继采用模拟处理，不对接收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。图 1 所示为单中继 AF 模式示意图及基本过程。

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。

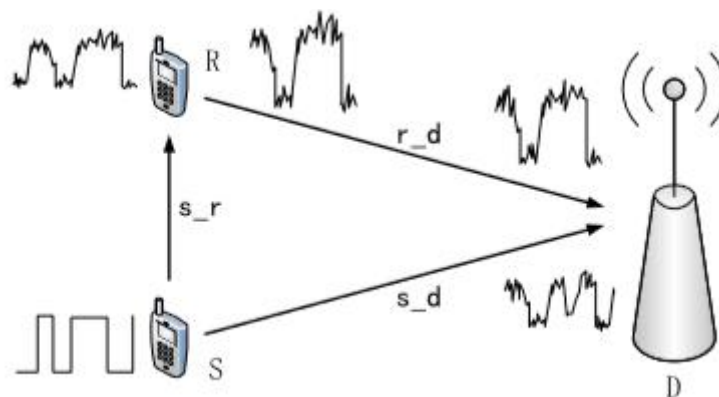


图 1 AF 模式示意图

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  $x_s$ ，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  $y_{s,r}$ ：

$$y_{s,r} = \sqrt{P_s} h_{s,r} x_s + n_{s,r} \quad (3.1)$$

目的节点 D 接收到的信号为  $y_{sd}$ ：

$$y_{s,d} = \sqrt{P_s} h_{s,d} x_s + n_{s,d} \quad (3.2)$$

其中， $P_s$  源发送的信号的功率为， $n_{s,r}$  为源节点与中继节点间信道噪声。

## 2) 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  $y_{sr}$  以系数  $\beta$  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  $\beta$  应满足：

$$\beta = \sqrt{\frac{P_s}{P_s |h_{s,r}|^2 + N_0}} \quad (3.3)$$

可见  $\beta$  取决于信道的衰落系数  $h_{s,r}$ 、源发送的信号的功率  $P_s$  和噪声功率  $N_0$ 。

那么，中继放大后的信号  $y_{AF}$ ：

$$y_{AF} = \beta y_{s,r} \quad (3.4)$$

其中， $P_r$  源发送的信号的功率。

## 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  $y_{sd}$  和来自中继节点的信号  $y_{rd}$ ，按照 MRC 方案进行合并，得到信号  $y$ ：

$$y = a_1 y_{s,d} + a_2 y_{r,d} \quad (3.5)$$

其中  $a_1$ 、 $a_2$  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：

$$a_1 = \frac{\sqrt{P_s} h_{s,d}}{N_0} \quad (3.6)$$

和

$$a_2 = \frac{\beta \sqrt{P_r} h_{s,r}^* h_{r,d}^*}{(\beta^2 |h_{r,d}|^2 + 1) N_0} \quad (3.7)$$

其中， $P_s$  和  $P_r$  分别为协作情况下源节点和中继节点的发送功率， $h_{s,d}^*$ 、 $h_{s,r}^*$ 、 $h_{r,d}^*$  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， $N_0$  为噪声功率。

### 3.1.2 DF 协作模式基本原理

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，即中继节点接收到信号后先解码然后转发。图 2 所示为单中继 DF 模式示意图及基本过程。

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。

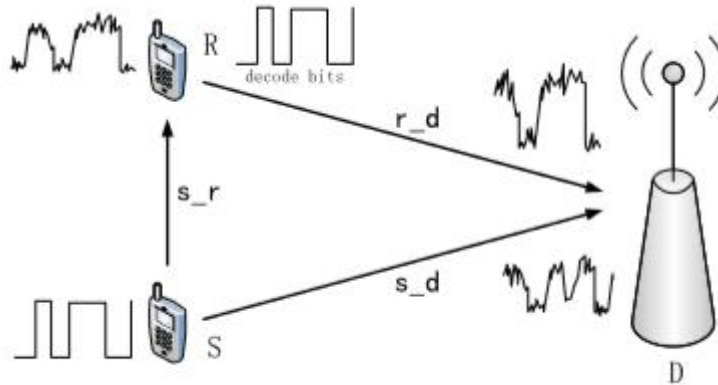


图 2 DF 模式示意图

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  $X_s$ , 其中一路直接 发送到目的节点 D, 一路发送到中继节点 R。

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  $y_{s,r}$  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。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  $y_{DF}$ ，目的节点接收到的来自中继的信号变为

$y_{r,d}$ ：

$$y_{r,d} = \sqrt{P_r} h_{r,d} y_{DF} + n_{r,d} \quad (3.7)$$

### 3) 目的端接收处理

同 AF 模式一样，DF 协议下目的节点按照 MRC 方案得到合并信号  $y$ ：

$$y = a_1 y_{s,d} + a_2 y_{r,d} \quad (3.8)$$

不同的是，加权系数  $a_1$ 、 $a_2$  取决于两路信号的信噪比最优值：

$$a_1 = \frac{\sqrt{P_s} h_{s,d}^*}{N_0} \quad (3.9)$$

和

$$a_2 = \frac{\sqrt{P_r} h_{r,d}^*}{N_0} \quad (3.10)$$

其中， $P_s$  和  $P_r$  分别为协作情况下源节点和中继节点的发送功率， $h_{s,d}^*$ 、 $h_{s,r}^*$ 、 $h_{r,d}^*$  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， $N_0$  为噪声功率。

### 3.1.3 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：

$$\gamma = \gamma_1 + \gamma_2 \quad (3.11)$$

其中，对于 AF 模式：

$$\gamma_1 = \frac{P_s |h_{s,d}|^2}{N_0} \quad (3.12)$$

$$\gamma_2 = \frac{1}{N_0} \frac{P_r^2 |h_{s,r}|^2 |h_{r,d}|^2}{P_r |h_{s,r}|^2 + P_r |h_{s,r}|^2 + N_0} \quad (3.13)$$

对于 DF 模式：

$$\gamma_1 = \frac{P_s |h_{s,d}|^2}{N_0} \quad (3.14)$$

$$\gamma_2 = \frac{P_r |h_{r,d}|^2}{N_0} \quad (3.15)$$

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  $P_e$  为：

$$P_e = P(0)P(0/1) + P(1)P(1/0) = \frac{1}{2} \text{erfc}(\sqrt{\gamma}) \quad \text{式(1-3-6)}$$

其中， $P(0/1)$  为符号“0”被判为符号“1”的概率。

在大信噪比 ( $\gamma \gg 1$ ) 条件下，式(3-3-4)可近似表示为：

$$P_e \approx \frac{1}{2\sqrt{\pi\gamma}} e^{-\gamma} \quad \text{式(1-3-7)}$$

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 ( $\gamma \gg \frac{P_s |h_{s,d}|^2}{N_0}$ )，因而其误码率得以降低。

### 3.2 程序流程、结构及变量说明

图 3 为程序中通信过程的说明及各环节信号、信道参数等的命名说明。一个有统一的一定命名规则的程序，是方便自己调试和他人阅读的。图中，每个节点之前为接收到的信号，节点之后为经过该节点处理后的信号：



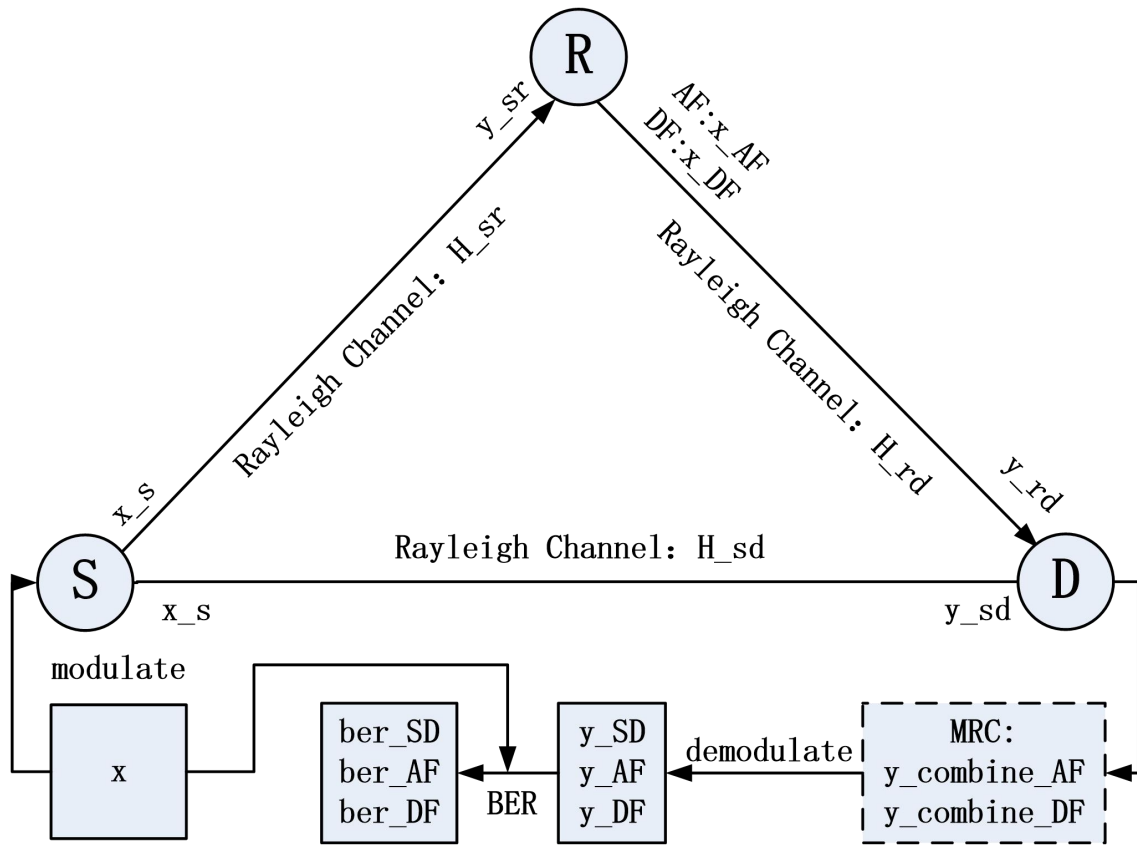


图3 程序通信过程及各环节参量的命名说明

图4为程序的流程图，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。

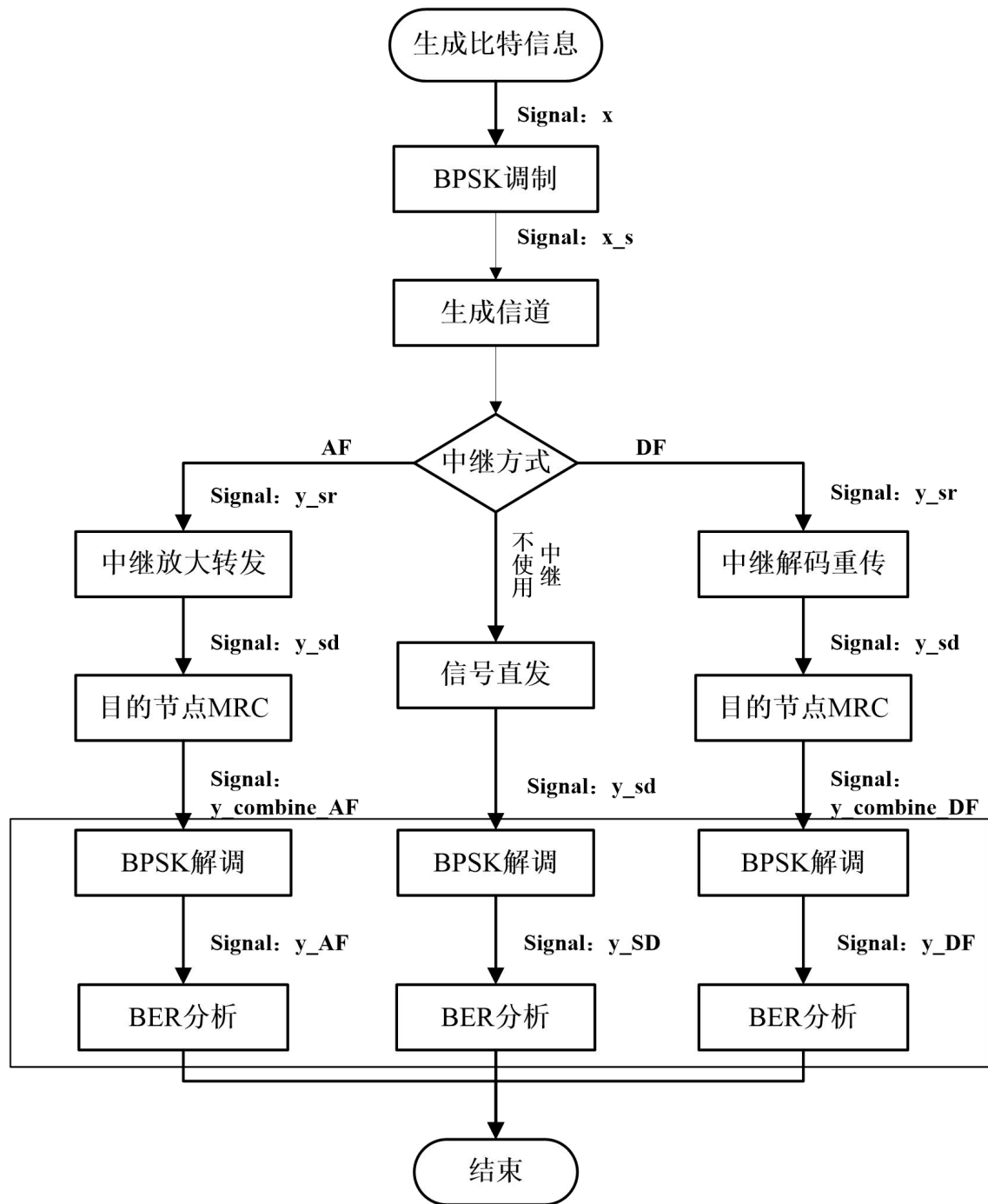


图 4 程序流程图

图 5 为程序的总体结构图，在生成信号、BPSK 调制和生成信道之后，进行不同信噪比下的仿真循环，每个信噪比均进行 Monte\_MAX 次蒙特卡罗循环。

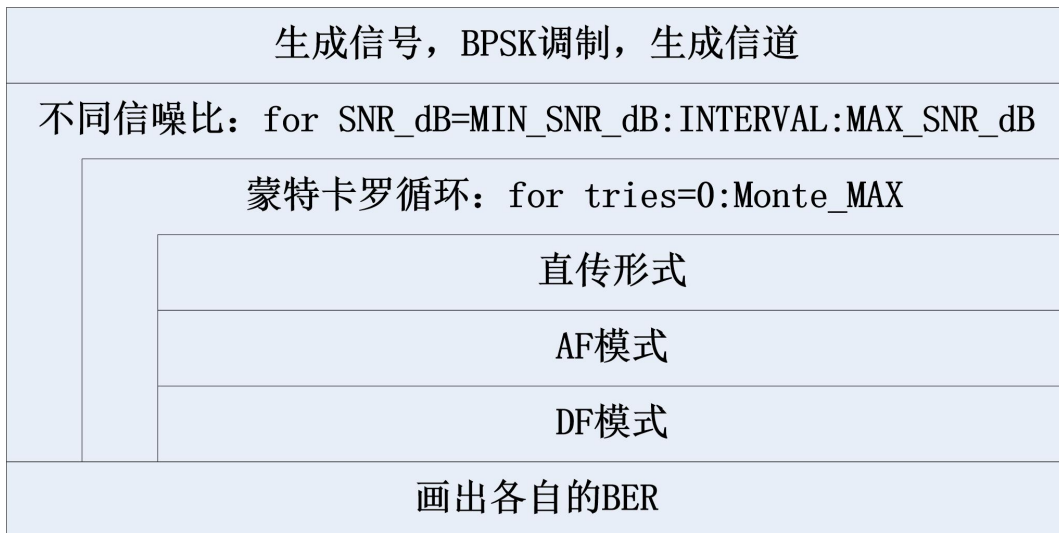


图 5 程序结构图

### 3.3 程序模块实现说明

#### 3.3.1 信道模型

为简化程序，信道采用恒参的瑞利衰落信道，即在一次通信过程中（一次蒙特卡罗过程），衰落系数表现为一恒定复数的形式：

```
function H = RayleighCH(sigma2)
    % average value(0)
    mu = 0; sigma = sqrt(sigma2);
    % Standard deviation( $\sigma$ )
    H = normrnd(mu,sigma)+j*normrnd(mu,sigma);
```

信道的正态分布的均值为方差 $\mu = 0$ ，只需输入方差 $\sigma^2$ ，即可得到信道系数 H。函数  $R = \text{normrnd}(MU, SIGMA)$  的功能是生成均值为 MU、标准差为 SIGMA 的正态分布的随机数据。

高斯白噪声通过函数  $y = \text{awgn}(x, SNR, SIGPOWER)$  实现。如果 SIGPOWER 是数值，则其代表以 dBW 为单位的信号强度；如果 SIGPOWER 为 'measured'，则函数将在加入噪声之前测定信号强度。本程序以 'measured' 参数自动检测信号强度。

#### 3.3.2 MRC 实现

由于 AF 与 DF 模式的最大比合并加权系数的决定因素不同，AF 模式与中继节

点的放大系数 $\beta$ 有关，故自定义函数`y_combine = Mrc( varargin )`通过输入参数的数目不同以区分AF与DF。当然， $\beta$ 也由信道的参数决定，AF与DF模式的MRC加权系数的本质区别在于：AF由于是对接收信号的放大，所以对于R到D的信号加权系数还与S到R信道的参数、S到R信道的信号和噪声功率相关。

MRC的伪代码为：

### 1. 输入参数

信道系数、中继放大系数、信号功率、噪声功率、需要合并的两路信号

### 2. 判断输入参数数目nargin

if nargin = 8

DF模式，执行加权系数为

$a_{sd} = CH_{sd}' * \sqrt{POW\_S\_sd} / POW\_N\_sd;$

$a_{rd} = CH_{rd}' * \sqrt{POW\_S\_rd} / POW\_N\_rd;$

elseif nargin = 10

AF模式，执行加权系数为

$a_{sd} = CH_{sd}' * \sqrt{POW\_S\_sd} / POW\_N\_sd;$

$a_{rd} = (\beta * \sqrt{POW\_S\_rd} * CH_{sr}' * CH_{rd}') /$   
 $((\beta^2 * (\text{abs}(CH_{rd}))^2 + 1) * POW\_N\_rd);$

### 3. 返回合并信号

$y\_combine = a_{sd} * \text{signal\_sd} + a_{rd} * \text{signal\_rd};$

## 3.3.3 BER 实现

对于实际 BER，通过统计蒙特卡罗的误码数，与全部比特数目作比得到。对于理论 BER，通过调用自定义函数得到，其伪代码为：

### 1. 输入参数

非协作的信噪比、信道系数、信号功率、噪声功率

### 2. 判断输入参数数目nargin

if nargin = 1

非协作模式，执行

$\gamma_{sd} = SNR\_SD;$

$\gamma_{rd} = 0;$

elseif nargin = 6

DF模式，执行加权系数为

$\gamma_{sd} = (POW\_S\_sd * (abs(CH\_sd))^2) / POW\_N\_sd;$

$\gamma_{rd} = (POW\_S\_rd * (abs(CH\_rd))^2) / POW\_N\_rd;$

elseif nargin = 7

AF模式，执行

$\gamma_{sd} = (POW\_S\_sd * (abs(CH\_sd))^2) / POW\_N\_sd;$

$numerator = (POW\_S\_rd)^2 * (abs(CH\_sr))^2 * (abs(CH\_rd))^2;$

$denominator = (POW\_S\_rd * (abs(CH\_sr))^2 + POW\_S\_rd * (abs(CH\_rd))^2 + POW\_N\_rd) * POW\_N\_rd;$

$\gamma_{rd} = numerator / denominator;$

### 3. 计算系统信噪比

$\gamma = \gamma_{sd} + \gamma_{rd};$

### 4. 计算系统误码率

$theoretical\_BER = 1 / (2 * \sqrt{\pi * \gamma} * \exp(\gamma));$  % Approximation

%  $theoretical\_BER = \text{erfc}(\sqrt{\gamma}) / 2;$  % Actual value

#### 3.3.4 AF 实现

图 6 是 AF 策略的程序流程图：

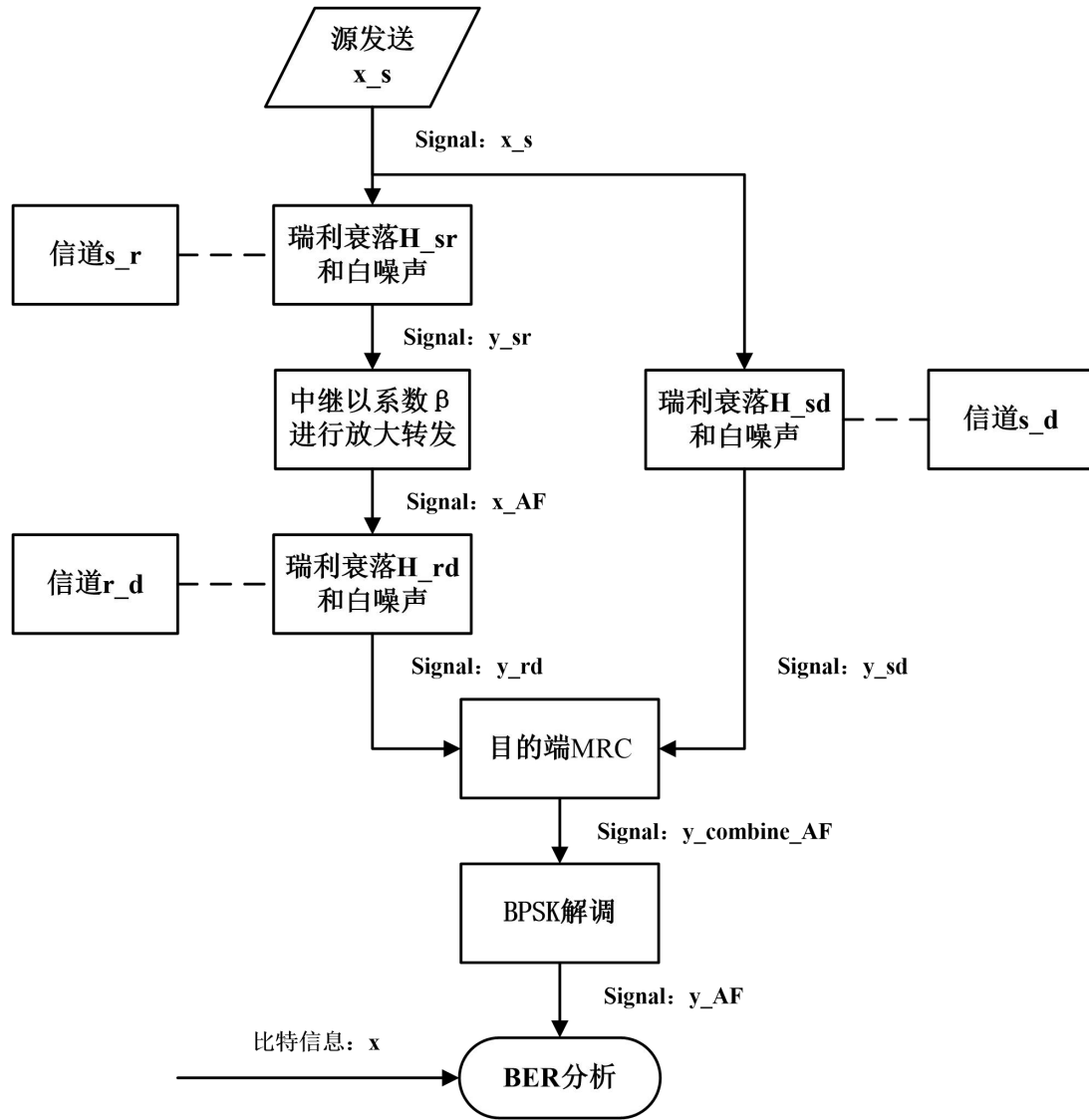


图 6 AF 策略流程图

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，其函数实现：

```
function [beta,r_d_AF] = AF(CH_sr,POW_S,POW_N,signal_sr)
    % amplification factor
    beta = sqrt( POW_S ) / ( (abs(CH_sr))^2 * POW_S + POW_N );
    % Relay will transmit the AF signal ' signal_AF '
    signal_AF = beta * signal_sr;
```

函数通过输入 S 和 R 之间的信道衰落系数、信号与噪声的功率和需要转发的信号  $s_r$ ，实现中继的 AF 过程，返回值为：经放大处理后的信号，它将被发送到目的节点；以及放大系数  $\beta$ ，以供目的节点 MRC 使用。

### 3.3.5 DF 实现

图 7 是 DF 策略的程序流程图：

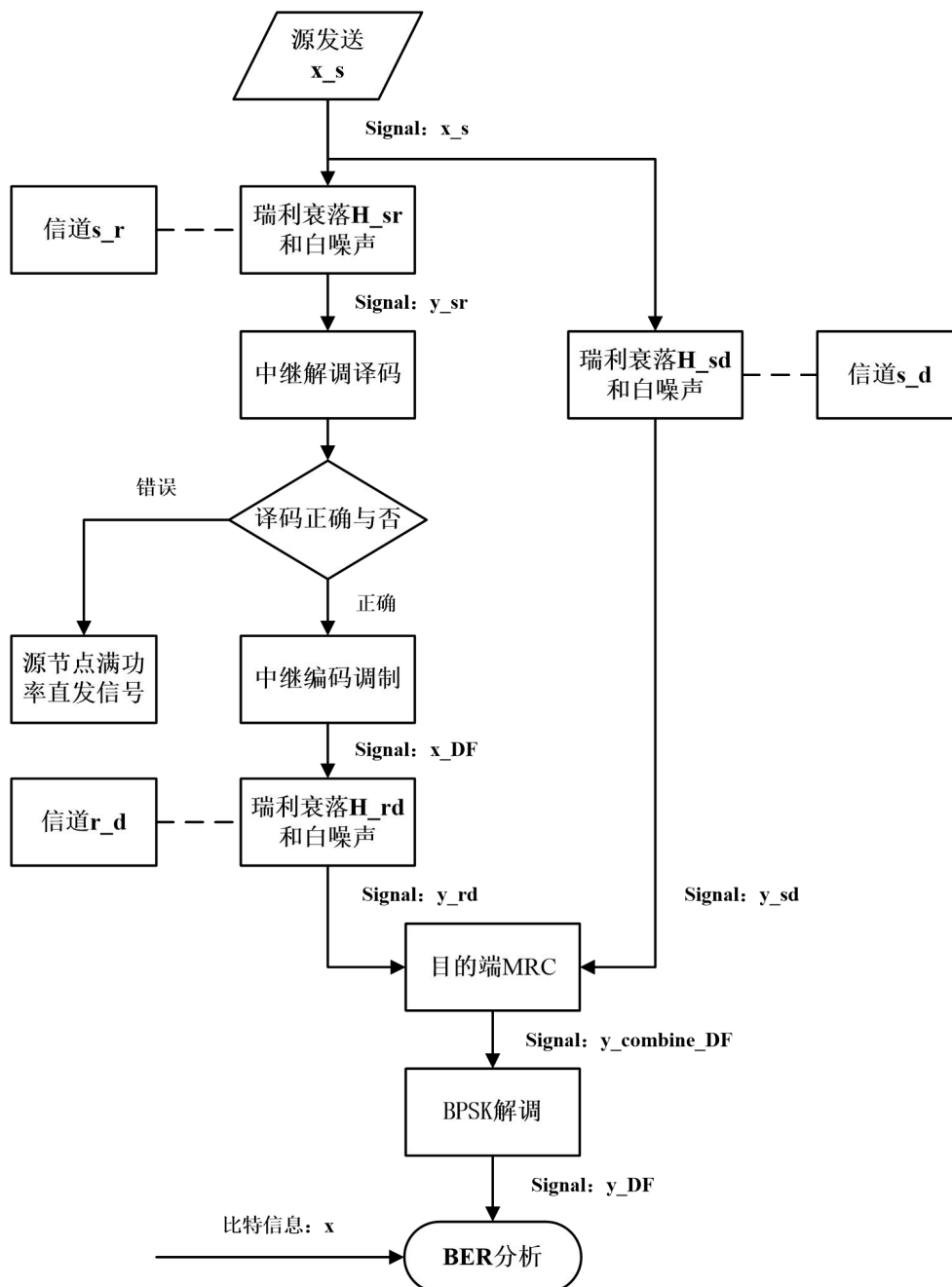


图 7 DF 策略流程图

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。为体现 DF 协议的过程，其函数实现编写为：

```
function signal_DF = DF(M,signal_sr,signal_x)
% first: demodulate the signal 's_r'
    signal_demod = demodulate(modem.pskdemod(M),signal_sr);
% second: decode r_d_l',And determine whether decoding correct
%+++++++If use a fixed DF,forced to set 'tx_coop' '1'+++++++
    tx_coop = 1;    % a sign, indicates whether forwarding
% if (sum(signal_x ~= signal_demod)>0)
%     tx_coop = 0;
% end    % if the relay decoding error, not forward
%+++++++
% third: modulate the signal 'signal_demod'
    if tx_coop == 0
        disp('Fail in DF,Source should transmit the signal to Destination directly
            Without Cooperation');
    elseif tx_coop == 1
        % As the force decoding is correct, so 'signal_demod' is equivalent to
        'signal_x'
        signal_DF = modulate(modem.pskmod(M),signal_x);
    end
```

实际上，在本仿真中可以等效简化为：

```
function signal_DF = DF(M, signal_x)
    signal_DF = modulate(modem.pskmod(M), signal_x);
```

函数通过输入源节点 PSK 调制的进制数 M（以提高程序的通用性）、需要中继的信号 s\_r 和原始比特流 x（虚设置，用于校验是否正确解码，本次仿真中假定解码正确），实现中继的 DF 过程，返回值为重新调制之后的信号，它将被发送到目的节点。



#### 四、实验结果

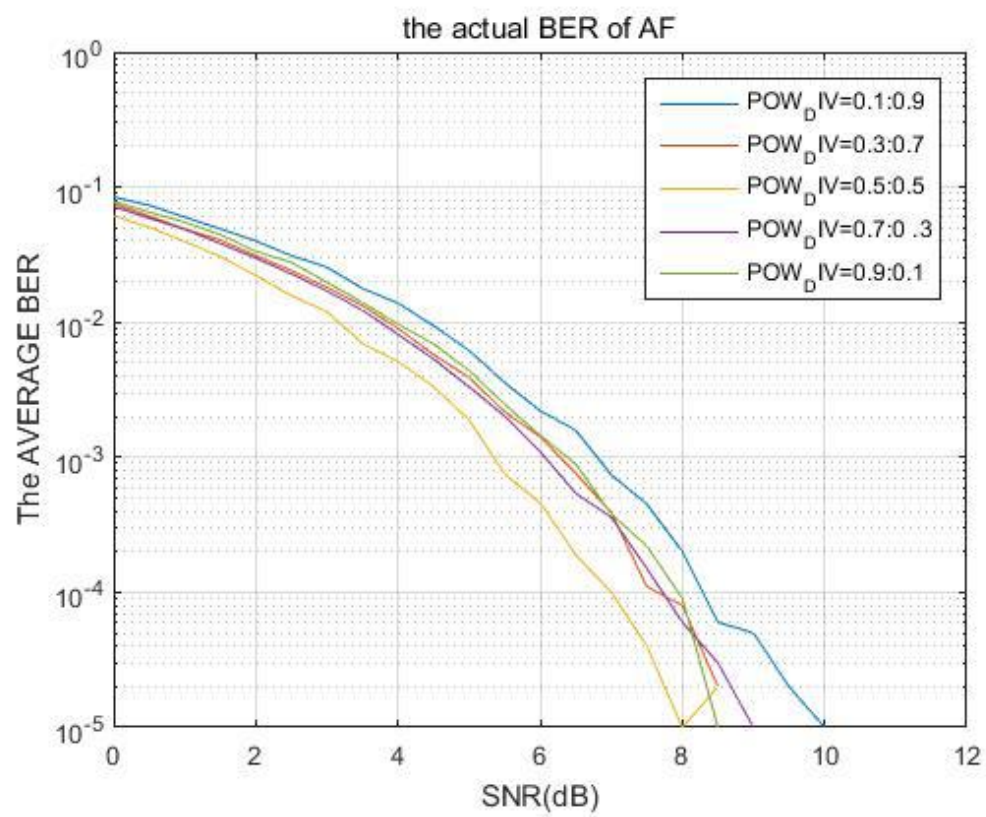


图 8 AF 协议下的实际误码率

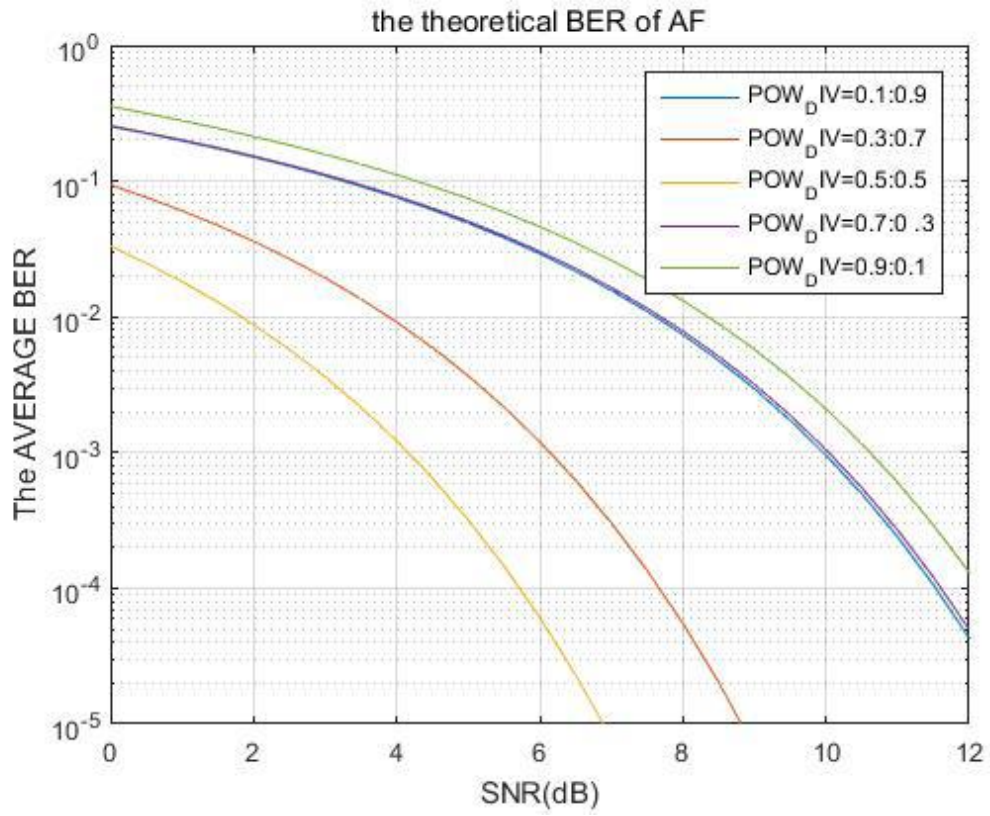


图 9 AF 协议下的理论误码率

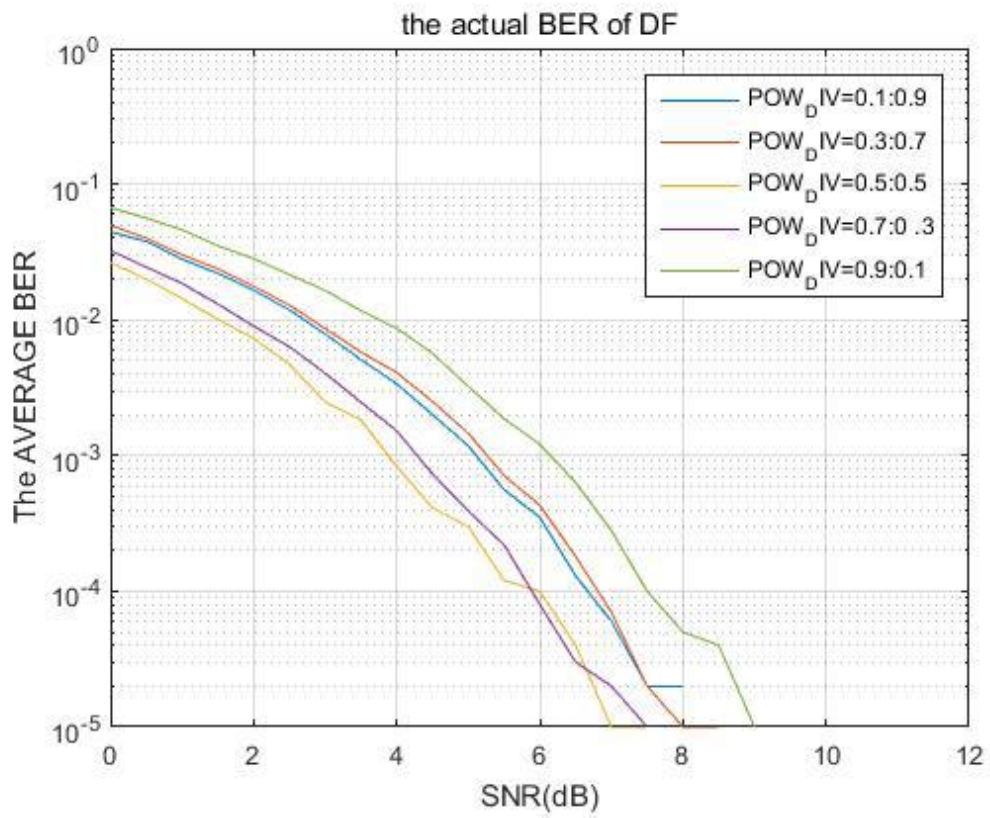


图 10 DF 协议下的实际误码率

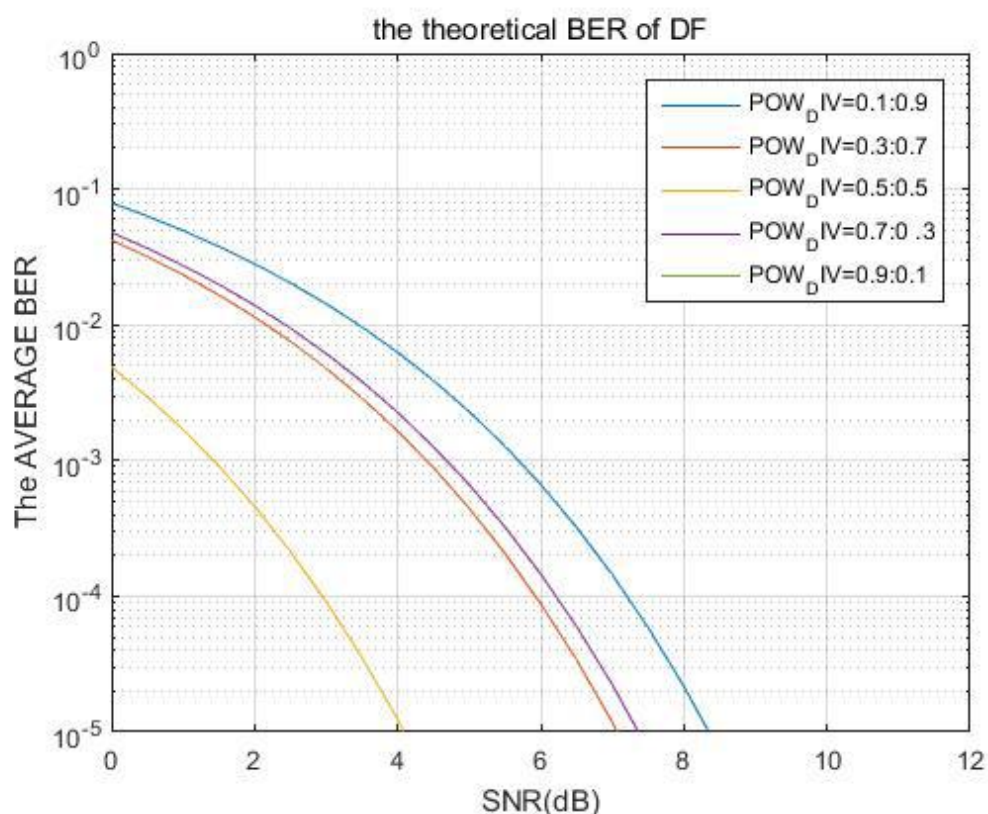


图 11 DF 协议下的理论误码率

## 五、实验总结

图 8 显示了在 AF 方式下的实际误码率，显然，当中继位于源和目的连线的中点处时，系统性能最优，而中继远离中点向源端或目的端方向移动时，系统性能越来越差，在靠近源端或目的端时，系统性能最差。即功率分配比例无论是上升还是下降都将会增大误码率。

图 9 显示了在 AF 方式下的理论误码率，同样在中继位于源和目的连线的中点处时，系统性能最优，而中继远离中点向源端或目的端方向移动时，系统性能越来越差，在靠近源端或目的端时，系统性能最差。

实验结果与实际误码率相符合。

图 10 显示了在 DF 方式下的实际误码率，可以看出当中继位于源和目的连线的中点处时，系统性能最优，而中继远离中点向源端或目的端方向移动时，系

统性能越来越差，在靠近源端或目的端时，系统性能最差。即功率分配比例无论是上升还是下降都将会增大误码率。

图 11 显示了在 DF 方式下的理论误码率，同样在中继位于源和目的连线的中点处时，系统性能最优，而中继远离中点向源端或目的端方向移动时，系统性能越来越差，在靠近源端或目的端时，系统性能最差。

实验结果与实际误码率相符合。

通过对 AF 与 DF 的两组实验对比，可以得知，在 AF 方式或者 DF 方式下，中继位于源和目的连线的中点处时，系统性能最优；中继节点距离两端越远性能越差。

此次实验，加深了对于一般路径损耗模型与距离变化关系的理解，能够独立运用 matlab 语言实现其对于路径损耗与距离的仿真，实验难度不大，但是对于自身意义不小。

参考文献：

【1】隋海虹, 宋高俊, 蒋丽霞. 基于误码率的中继位置及功率分配联合优化[J]. 计算机系统应用, 2012, 21(3):96-100.

【2】孙琳, 马社祥. 基于误码率的快速中继选择算法 [J]. 计算机应用, 2011, 31(3):613-616.

【3】商敏虹, 武贵路, 徐平平. 基于误码率的 DF 协作通信系统功率分配方法, 现代电子技术, 2014, 37 (21) : 9-11.

## 2. 实验程序

Matlab 调试程序:

```
%Author: Penghui Zhang
%Date:2018/12/13
%*****Main_Start_AF*****
clear all;%%清除了所有的变量，包括全局变量 global
datestr(now)%生成指定格式的日期和时间，now 代表当前日期
n=1;
for i=0.1:0.2:0.9
[SNR_dB,ber_AF,theo_ber_AF]=main_AF(i);
a(n,:)=SNR_dB;
b(n,:)=ber_AF;
c(n,:)=theo_ber_AF;
n=n+1;
end

figure(1)%the actual BER of AF
semilogy(a(1,:),b(1,:),a(2,:),b(2,:),a(3,:),b(3,:),a(4,:),b(4,:),a(5,:),b(5,:),));
legend('POW_DIV=0.1:0.9','POW_DIV=0.3:0.7','POW_DIV=0.5:0.5','POW_DIV=0.7:0.3','POW_DIV=0.9:0.1');
grid on; %增加网格
ylabel('The AVERAGE BER');
xlabel('SNR(dB)');
title('the actual BER of AF');
axis([0,12,10^(-5),1]);

figure(2) % the theoretical BER of AF
semilogy(a(1,:),c(1,:),a(2,:),c(2,:),a(3,:),c(3,:),a(4,:),c(4,:),a(5,:),c(5,:),));
legend('POW_DIV=0.1:0.9','POW_DIV=0.3:0.7','POW_DIV=0.5:0.5','POW_DIV=0.7:0.3','POW_DIV=0.9:0.1');
grid on;
ylabel('The AVERAGE BER');
xlabel('SNR(dB)');
title('the theoretical BER of AF');
axis([0,12,10^(-5),1]);

%*****Main_Start_DF*****

clear all;%%清除了所有的变量，包括全局变量 global
datestr(now)%生成指定格式的日期和时间，now 代表当前日期
```

```

n=1;
for i=0.1:0.2:0.9
[SNR_dB,ber_DF,theo_ber_DF]=main_DF(i);
a(n,:)=SNR_dB;
b(n,:)=ber_DF;
c(n,:)=theo_ber_DF;
n=n+1;
end

figure(1)%the actual BER of DF
semilogy(a(1,:),b(1,:),a(2,:),b(2,:),a(3,:),b(3,:),a(4,:),b(4,:),a(5,:),b(5,,:));
legend('POW_DIV=0.1:0.9','POW_DIV=0.3:0.7','POW_DIV=0.5:0.5','POW_DIV=0.7:0.3','POW_DIV=0.9:0.1');
grid on; %增加网格
ylabel('The AVERAGE BER');
xlabel('SNR(dB)');
title('the actual BER of DF');
axis([0,12,10^(-5),1]);

figure(2) % the theoretical BER of DF
semilogy(a(1,:),c(1,:),a(2,:),c(2,:),a(3,:),c(3,:),a(4,:),c(4,:),a(5,:),c(5,,:));
legend('POW_DIV=0.1:0.9','POW_DIV=0.3:0.7','POW_DIV=0.5:0.5','POW_DIV=0.7:0.3','POW_DIV=0.9:0.1');
grid on;
ylabel('The AVERAGE BER');
xlabel('SNR(dB)');
title('the theoretical BER of DF');
axis([0,12,10^(-5),1]);

%*****main_AF*****
function [SNR_dB,ber_AF,theo_ber_AF] = main_AF( POW_DIV1 )
%% original definition
    MIN_SNR_dB = 0;
    MAX_SNR_dB = 14;
    INTERVAL = 0.5; % SNR interval
    % POW_DIV = 1/2; % 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_DIV2 = 1-POW_DIV1; % send singal to Destinatio
    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
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(modem.pskmod(M),x); % The signal 'x_s' after bpsk
modulation

%% Rayleigh Fading / Assumed to cross reference channel %采用恒参的瑞利
衰落信道, 即一次通信过程中, 衰落系数表现为一恒定复数形式
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

%% 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;

    for tries=0:Monte_MAX

        sig = 10^(SNR_dB/10); % SNR, said non-dB
        POW_S1 = POW_DIV1;    % Signal power sr
        POW_S2 = POW_DIV2;    % Signal power rd
        POW_N1 = POW_S1 / sig; % Noise power sr
        POW_N2 = POW_S2 / sig; % Noise power rd

        % 'x_s' is transmitted from Source to Relay and Destination
        % AWGN:在某一信号中加入高斯噪声
        y_sd = awgn( sqrt(POW)*H_sd * x_s, SNR_dB, 'measured'); %
Destination received the signal 'y_sd' from Source %'measured'表示测定
信号强度

```

```

        y_sr = awgn( sqrt(POW_DIV1)*H_sr * x_s, SNR_dB, 'measured');    %
Relay received the signal 'y_sr' from Source

        %y = awgn(x,SNR,SIGPOWER) 如果 SIGPOWER 是数值, 则其代表以 dBW 为单位的
信号强度; 如果 SIGPOWER 为'measured', 则函数将在加入噪声之前测定信号强度。

        %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_S2,POW_N2,y_sr);
        y_rd = awgn( sqrt(POW_S2)*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_S2,POW_N2,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 %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);
    % 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_S2,POW_N2);

end;    % for SNR_dB=MIN_SNR_dB:INTERVAL:MAX_SNR_dB

%% draw BER curves
SNR_dB = MIN_SNR_dB:INTERVAL:MAX_SNR_dB;
End

%*****main_DF*****
function [SNR_dB,ber_DF,theo_ber_DF] = main_DF( POW_DIV1 )

```



```

%% original definition
MIN_SNR_dB = 0;
MAX_SNR_dB = 14;
INTERVAL = 0.5; % SNR interval
% POW_DIV = 1/2; % 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_DIV2 = 1-POW_DIV1; % send signal to Destination
POW = 1; % without cooperation, Source send signals directly to
the Destination 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
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(modem.pskmod(M),x); % The signal 'x_s' after bpsk
modulation

%% Rayleigh Fading / Assumed to cross reference channel %采用恒参的瑞利
衰落信道，即一次通信过程中，衰落系数表现为一恒定复数形式
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

%% 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_DF = 0;

    for tries=0:Monte_MAX

```

```

sig = 10^(SNR_dB/10); % SNR, said non-dB
POW_S1 = POW_DIV1;      % Signal power sr
POW_S2 = POW_DIV2;      % Signal power rd
POW_N1 = POW_S1 / sig;  % Noise power sr
POW_N2 = POW_S2 / sig;  % Noise power rd

% 'x_s' is transmitted from Source to Relay and Destination
% AWGN:在某一信号中加入高斯噪声
y_sd = awgn( sqrt(POW)*H_sd * x_s, SNR_dB, 'measured'); %
Destination received the signal 'y_sd' from Source %'measured'表示测定
信号强度
y_sr = awgn( sqrt(POW_DIV1)*H_sr * x_s, SNR_dB, 'measured'); %
Relay received the signal 'y_sr' from Source
%y = awgn(x,SNR,SIGPOWER) 如果 SIGPOWER 是数值, 则其代表以 dBW 为单位的
信号强度; 如果 SIGPOWER 为'measured', 则函数将在加入噪声之前测定信号强度。

%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 固定解码转发中继
协议
x_DF = DF(M,y_sr,x);
y_rd = awgn( sqrt(POW_DIV2)*H_rd * x_DF, SNR_dB, 'measured');
y_combine_DF =
Mrc( H_sd,H_rd,POW_S1,POW_N1,POW_S2,POW_N2,y_sd,y_rd);
y_DF = demodulate(modem.pskdemod(M),y_combine_DF);
err_num_DF = err_num_DF + Act_ber(x,y_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_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_DF(snrcount) =
Theo_ber(H_sd,H_sr,H_rd,POW_S1,POW_N1,POW_S2,POW_N2);

end; % for SNR_dB=MIN_SNR_dB:INTERVAL:MAX_SNR_dB

%% draw BER curves

```

```
SNR_dB = MIN_SNR_dB:INTERVAL:MAX_SNR_dB;  
end
```