MIMO 技术

zfzdr

2020 年 9 月 15 日起

目录

1 MIMO 信道

2 MIMO 信道估计

3 MIMO CSI-feedback

4 MIMO 检测

4.1 经典检测方法

假设接收端已知 CSI H^{UL}，系统模型表示为:

$$y_{MAC} = H^{UL}x + z \quad y = Hx + z = H_1x_1 + H_2x_2 + \cdots + H_Kx_K + z \tag{1}$$

4.1.1 ZF 检测

$$\hat{x}_{ZF} = G_{ZF}y \tag{2}$$

其中

$$G_{ZF} = (H^H H)^{-1} H^H$$

4.1.2 MMSE 检测

$$\hat{x}_{MMSE} = G_{MMSE}y$$

其中

$$G_{MMSE} = \arg\underset{G}{min} E\{\|Gy - x\|^2\} = (H^H H + \sigma^2 I_{Nt*K})^{-1} H^H$$

- *regularized channel inversion*

$$\underline{H} = \begin{bmatrix} H \\ \sigma^2 I_{N_t} \end{bmatrix} and \;\; \underline{y} = \begin{bmatrix} y \\ 0_{N_t \times 1} \end{bmatrix}$$

Then

$$G_{MMSE} = (\underline{H}^H \underline{H})^{-1} \underline{H}^H$$

4.2 ZF-SIC 检测

4.3 MMSE-SIC 检测

5 MIMO 预编码

5.1 点对点 MIMO 系统模型

$$y = Hx + z \tag{3}$$

其中，$y \in C^{N_r \times 1}$，$x \in C^{N_t \times 1}$，$z \in C^{N_r \times 1}$，发送端总能量为 $E\{x^H x\} = P$，噪声功率谱密度为 N_0，即 $E\{zz^H\} = N_0 I_{N_r}$，且

$$\begin{aligned} R_{yy} &= E\{yy^H\} \\ &= HR_{xx}H^H + N_0 I_{N_r} \end{aligned} \tag{4}$$

5.2 系统容量

$$\begin{aligned} I(x;y) &= H(x) - H(x|y) \\ &= H(y) - H(y|x) \\ &= H(y) - H(Hx + z|x) \\ &= H(y) - H(z|x) \\ &= H(y) - H(z) \end{aligned} \tag{5}$$

其中，z 是满足复高斯随机分布的多维向量，因此当且仅当 y 也满足复高斯随机分布时，上式取得最大值，且

$$\begin{aligned} &H(y) = log_2|\pi e R_{yy}| = log_2|\pi e H R_{xx} H^H + \pi e N_0 I_{N_r}| \\ &H(z) = log_2|\pi e N_0 I_{N_r}| \end{aligned} \tag{6}$$

于是，

$$I(x;y) = log_2 \left| I_{N_r} + \frac{HRxxH^H}{N_0} \right| \tag{7}$$

5.3 传输侧无 CSI

假设每根天线上的发送信号能量相等且相互独立，即 $R_{xx} = \frac{P}{N_t} I_{N_t}$，则

$$\begin{aligned} C &= log_2 \left| I_{N_r} + \frac{P}{N_t N_0} H H^H \right| \\ &= \sum_{i=1}^{N_t} log_2(1 + \frac{P}{N_t N_0} \lambda_i) \end{aligned} \tag{8}$$

5.4 传输侧有 CSI

预编码提高信道容量

对信道矩阵 H 使用 SVD 分解，即 $H = U \Sigma V^H$，一般假设 $Nr > Nt$, 则

$$\Sigma = \begin{bmatrix} \sqrt{\lambda_1} & 0 & \cdots & 0 \\ 0 & \sqrt{\lambda_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sqrt{\lambda_{N_t}} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{9}$$

令调制后信号能量表示为 $\tilde{x}$，预编码后的发送信号能量为 $x = V^H \tilde{x}$, 则

$$\begin{aligned} y &= Hx + z \\ &= U \Sigma V^H V \tilde{x} + z \\ &= U \Sigma \tilde{x} + z \end{aligned} \tag{10}$$

$$U^H y = U^H U \Sigma \tilde{x} + U^H z => \tilde{y} = \Sigma \tilde{x} + \tilde{z} \tag{11}$$

上式展开为

$$\begin{bmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \vdots \\ \tilde{y}_{N_t} \\ \vdots \\ \tilde{y}_{N_r} \end{bmatrix} = \begin{bmatrix} \sqrt{\lambda_1} & 0 & \cdots & 0 \\ 0 & \sqrt{\lambda_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sqrt{\lambda_{N_t}} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_{N_t} \end{bmatrix} + \begin{bmatrix} \tilde{z}_1 \\ \tilde{z}_2 \\ \vdots \\ \tilde{z}_{N_t} \\ \vdots \\ \tilde{z}_{N_r} \end{bmatrix} \tag{12}$$

即

$$\tilde{y}_i = \sqrt{\lambda_i} \tilde{x}_i + \tilde{z}_i, i = 1, \cdots, r. \quad r = N_t \tag{13}$$

原始的 MIMO 信道等效为 r 个 SISO 信道，每个 SISO 信道的信道容量可以表示为：

$$C_i(P_i) = log_2(1 + \frac{\lambda_i P_i}{N_0}) \tag{14}$$

其中，P_i 表示第 i 根天线上的信号能量，且

$$E\{x^H x\} = \sum_{i=1}^{N_t} E\{|x_i|^2\} = \sum_{i=1}^{N_t} P_i = P \tag{15}$$

于是，信道总容量为：

$$C = \sum_{i=1}^{N_t} C_i(P_i) = \sum_{i=1}^{N_t} log_2(1 + \frac{\lambda_i P_i}{N_0}) \tag{16}$$

可以通过注水算法优化功率分配，达到更大的信道容量，即

$$\begin{aligned} & C = \arg\max_{\{P_i\}} \sum_{i=1}^{N_t} C_i(P_i) = \sum_{i=1}^{N_t} log_2(1 + \frac{\lambda_i P_i}{N_0}) \\ & s.t \quad \sum_{i=1}^{N_t} P_i = P \end{aligned} \tag{17}$$

最优解为

$$\begin{aligned} & P_i^{opt} = (\mu - \frac{N_0}{\lambda_i})^+, i = 1, \cdots, r \\ & \sum_{i=1}^{N_t} P_i = P \end{aligned} \tag{18}$$

5.5 MIMO 注水算法

Algorithm 1: 注水算法

Step1: 迭代计算 p=1, 计算 $\mu = \frac{N_t}{r-\rho+1}$

Step2: 用 μ 计算 $\gamma_i = \mu - \frac{N_t N_0}{E_x \lambda_i}, i = 1, 2, \cdots, r - p + 1$

Step3: 若分配到最小增益的信道能量为负值，即设 $\gamma_{r-p+1} = 0, p = p + 1$，转至 Step1

若任意 γ_i 非负，即得到最佳注水功率分配策略

6 Latex 语法学习

6.1 伪代码

http://hustsxh.is-programmer.com/posts/38801.html

algorithmic 和 algorithmics algorithmic 和 algorithmicx，这两个包很像，很多命令都是一样的，只是 algorithmic 的命令都是大写，algorithmicx 的命令都是首字母大写，其他小写 (EndFor 两个大写)。下面是 algorithmic 的基本命令还有 algorithm2e,latex 的与 algorithm 相关的包常用的有几个，algorithm、algorithmic、algorithmicx、algorithm2e，可以大致分成三类，或者说三个排版环境。最原始的是使用 algorithm+algorithmic，这个最早出现，也是最难用的，需要自己定义一些指令。第二个排版环境是 algorithm+algorithmicx，algorithmicx 提供了一些宏定义和一些预定义好了的环境 (layout)，指令类似 algorithmic。第三个是 algorithm2e，只需要一个包，使用起来和编程的感觉很像，也是我更倾向使用的包。下面是使用 algorithm2e 的例子。[1]

6.2 超链接

需要的宏包 hyperref,