# MIMO 技术

zfzdr

2020 年 9 月 15 日起

## 目录

# 第一部分 MIMO 检测

## 1 点对点 MIMO 检测

### 1.1 系统模型

$$y = Hx + z$$

### 1.2 经典检测方法

#### 1.2.1 MF 检测

$$\hat{x}_{MF} = H^H y \tag{1}$$

匹配滤波器，又称最大比合并，目标是最大化接收信噪比。

- 优点：复杂度低，不需要矩阵求逆运算

- 缺点：对于病态矩阵（ill-conditioned），检测性能严重劣化

### 1.2.2 ZF 检测

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

其中 $G_{ZF} = (H^H H)^{-1} H^H$，目标是最大化接收信干噪比（received signal-to-interference ration）

- 优点：性能优于 MF（复杂度较 MF 高）
- 缺点：对忽略噪声的影响，与 MF 类似，会有噪声加强的副作用

### 1.2.3 MMSE 检测

MMSE 检测是优化问题的解：

$$\begin{aligned} G_{MMSE} &= \underset{G \in C^{N_t \times N_r}}{\arg min} \|x - Gy\|^2 \\ &= \underset{G}{\arg min} E\{\|Gy - x\|^2\} = (H^H H + \sigma^2 I_{Nt})^{-1} H^H \end{aligned} \tag{3}$$

于是，MMSE 检测表达式为：

$$\hat{x}_{MMSE} = G_{MMSE} y \tag{4}$$

*regularized channel inversion*

$$\underline{H} = \begin{bmatrix} H \\ \sigma^2 I_{N_t} \end{bmatrix} and \ \underline{y} = \begin{bmatrix} y \\ 0_{N_t \times 1} \end{bmatrix} \tag{5}$$

Then

$$G_{MMSE} = (\underline{H}^H \underline{H})^{-1} \underline{H}^H \tag{6}$$

- 优点：有效解决了 ZF 和 MF 导致的噪声加强问题，能够在噪声功率较大（信噪比较低时）时获得较大的性能增益
- 缺点：需要对噪声功率谱密度的先验信息，且需要矩阵求逆

### 1.2.4 ZF-SIC 检测

### 1.2.5 MMSE-SIC 检测

### 1.2.6 总结

最早的大规模 MIMO 检测器是在 2008 年由 Vardhan et al. [**?**] 提出的，是基于最大似然搜索的。紧接着，研究者提出了使用局部搜索和置信传播提出了近似 ML 的检测器。但是随着天线数目的增加，矩阵求逆操作的复杂度呈指数增长，计算复杂度急剧上升，为了解决这个问题，2013 年，Wu et al. [**?**] 提出了基于近似求逆方法的上行检测器。

# 第二部分 MIMO 预编码

# 2 点对点 MIMO 系统

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

$$\begin{aligned} R_{yy} &= E\{yy^H\} \\ &= HR_{xx}H^H + N_0 I_{N_r} \end{aligned} \tag{8}$$

## 2.2 系统容量

$$\begin{aligned} I(x;y) &= H(x) - H(x|y) \\ &= H(y) - H(y|x) \\ &= H(y) - H(Hx + z|x) \\ &= H(y) - H(z|x) \\ &\cdot = H(y) - H(z) \end{aligned} \tag{9}$$

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

$$\begin{aligned} H(y) &= log_2|\pi e R_{yy}| = log_2|\pi e HR_{xx}H^H + \pi e N_0 I_{N_r}| \\ H(z) &= log_2|\pi e N_0 I_{N_r}| \end{aligned} \tag{10}$$

于是，

$$I(x;y) = log_2 \left| I_{N_r} + \frac{HRxxH^H}{N_0} \right| \tag{11}$$

## 2.3 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

$$\begin{aligned} C &= log_2 \left| I_{N_r} + \frac{P}{N_t N_0} HH^H \right| \\ &= \sum_{i=1}^{N_t} log_2(1 + \frac{P}{N_t N_0} \lambda_i) \end{aligned} \tag{12}$$

## 2.4 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

$$\Sigma = \begin{bmatrix} \sqrt{\lambda_1} & 0 & \cdots & 0 \\ 0 & \sqrt{\lambda_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sqrt{\lambda_{N_t}} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{13}$$

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

$$\begin{aligned} y &= Hx + z \\ &= U\Sigma V^H V\tilde{x} + z \\ &= U\Sigma\tilde{x} + z \end{aligned} \tag{14}$$

$$U^H y = U^H U\Sigma\tilde{x} + U^H z => \tilde{y} = \Sigma\tilde{x} + \tilde{z} \tag{15}$$

上式展开为

$$\begin{bmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \vdots \\ \tilde{y}_{N_t} \\ \vdots \\ \tilde{y}_{N_r} \end{bmatrix} = \begin{bmatrix} \sqrt{\lambda_1} & 0 & \cdots & 0 \\ 0 & \sqrt{\lambda_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sqrt{\lambda_{N_t}} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \vdots \\ \tilde{x}_{N_t} \end{bmatrix} + \begin{bmatrix} \tilde{z}_1 \\ \tilde{z}_2 \\ \vdots \\ \tilde{z}_{N_t} \\ \vdots \\ \tilde{z}_{N_r} \end{bmatrix} \tag{16}$$

即

$$\tilde{y}_i = \sqrt{\lambda_i}\tilde{x}_i + \tilde{z}_i, i = 1, \cdots, r.\square\square\ r = N_t \tag{17}$$

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：

$$C_i(P_i) = log_2(1 + \frac{\lambda_i P_i}{N_0}) \tag{18}$$

其中，$P_i$ 表示第 $i$ 根天线上的信号能量，且

$$E\{x^H x\} = \sum_{i=1}^{N_t} E\{|x_i|^2\} = \sum_{i=1}^{N_t} P_i = P \tag{19}$$

于是，信道总容量为：

$$C = \sum_{i=1}^{N_t} C_i(P_i) = \sum_{i=1}^{N_t} log_2(1 + \frac{\lambda_i P_i}{N_0}) \tag{20}$$

可以通过注水算法优化功率分配，达到更大的信道容量，即

$$\begin{aligned} C &= \arg\max_{\{P_i\}} \sum_{i=1}^{N_t} C_i(P_i) = \sum_{i=1}^{N_t} log_2(1 + \frac{\lambda_i P_i}{N_0}) \\ s.t &\sum_{i=1}^{N_t} P_i = P \end{aligned} \tag{21}$$

最优解为

$$\begin{aligned} P_i^{opt} &= (\mu - \frac{N_0}{\lambda_i})^+, i = 1, \cdots, r \\ \sum_{i=1}^{N_t} P_i &= P \end{aligned} \tag{22}$$

## 2.5 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 $\mu$ 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 $\gamma_i$ 非负，即得到最佳注水功率分配策略

## 2.6 经典预编码算法

参考文献 [**?**]

**起源**　MIMO 无线通信技术最早广泛应用于 WiFi 技术上（IEEE 802.11ac/n),WiMax 技术（IEEE 802.16e）,4G(LTE/LTE-A) 等系统。最早的 MIMO 思想来源于点对点的 MIMO 系统，接着发展出了 MU-MIMO, 每个用户配备一个单天线。

### 2.6.1　线性预编码

**Matched Filter(MF)**

$$W_{MF} = \sqrt{\alpha} H^H \tag{23}$$

**ZF**

$$W_{ZF} = \sqrt{\alpha} H (H^H H)^{-1} \tag{24}$$

**RZF(或 MMSE)**

$$W_{MMSE} = \sqrt{\alpha} H (H^H H + X + \lambda I_K)^{-1} \tag{25}$$

**TPE**

$$W_{TPE} = \sum_{j=0}^{J-1} w_j (H^H H)^j H^H \tag{26}$$

**优缺点对比**　优缺点如图1所示。

TABLE II
ADVANTAGES AND DISADVANTAGES OF THE MF, ZF, RZF, TPE, AND PZF PRECODING TECHNIQUES

| SC precoding techniques | Advantages | Disadvantages |
|---|---|---|
| MF | -Near optimal performance if there are more BS antennas than users<br>-Low computational complexity<br>-Better performance at lower SNRs | -Unable to achieve full diversity at high spectral efficiency<br>-Suffering from error floors for all positive multiplexing gains<br>-Having lower achievable rate in the case of less BS antennas<br>-Not robust against interuser interference |
| ZF | -Low computational complexity<br>-Higher power efficient -Better performance at high SNRs<br>-Decoupling a multi-user channel into independent single-user channels<br>-Achieving a large portion of dirty paper coding capacity | -Noise amplification and power penalty if the channel is highly correlated<br>-Unable to support too many users<br>-Medium complexity |
| RZF | -Guaranteed optimality if the ratio between the SINR requirement and the average channel attenuation is the same for all users | -Requiring matrix inverse calculation, leading to high complexity<br>-Suffering from error floors for all positive multiplexing gains |
| TPE | -Scalable computational complexity, thus computationally efficient<br>-Balancing precoding complexity and system throughput<br>-More suitable for real-time hardware implementation | -Lower order techniques having limited performance<br>-Higher performance requiring considerably more hardware |
| PZF | -Highly desirable closed-form performanc<br>-Low complexity -Facilitating multistream processing<br>-Large power gain | -Spectral efficiency being tightly upper bounded by the ZF precoder |

图 1: 优缺点对比

## 2.7 基于 AI 的预编码算法

1. *Kyeongbo Kong, Woo-Jin Song, Moonsik Min, Deep-learning-based precoding in multiuser MIMO downlink channels with limited feedback, 2008.0417*

   基于深度学习的均衡，反馈和预编码，下行 MU-MIMO 系统。基站侧发送预编码使用 transmitter DNN 实现，每个 receiver DNN 引入二值化层来模仿量化过程，从而使能端到端训练。由于二值化层的引入导致网络的梯度失真从而容易陷入局部极小值，网络训练过程中引入了辅助训练网络作为教导网络，有效避免陷入局部极小值。与传统的线性预编码相比，该网络通过端到端训练达到了更高的和速率。

2. *Novel MMSE Precoder and Decoder Designs Subject to Per-antenna Power Constraint for Uplink Multiuser MIMO Systems, I-Tai Lu, Jialing Li and Enoch Lu, 2009*

   关于多用户 MIMO 上行预编码的文章，理论比较多。文中提到，最经典的点对点 MIMO 系统在发送总功率一定时，最优预编码矩阵有最优解，但是在每根天线的功率有限或扩展到多用户 MIMO, 闭式解不存在，对于 MU-MIMO，每个用户总功率有限时，可以通过两种方法寻找数值解，TCIO 和 GIA。

3. *Evolution of Uplink MIMO for LTE-Advanced*

   LTE 中上行 MIMO 的技术介绍

4. *Deep Learning based Hybrid Precoding for mmWave Massive MIMO system using ComcepNet, 2020.7*

   提出了新的预编码网络结构”ComcepNet“，该网络结合了复数卷积块和 Inception 网络的特征，展现了比 autoprecoder 更好的性能。

5. *Deep Learning Based Precoder Design in MIMO Systems with Finite-Alphabet Inputs,2020*

   有限元预编码

6. *Deep Learning for Distributed Channel Feedback and Multiuser Precoding in FDD Massive MIMO, 20.07*

多用户信道估计和反馈问题可以视为一个分布式信源编码问题。与传统的每个用户独立估计信道信息不同，文章提出联合设计导频并且接收端使用 DNN 直接将接收导频映射为反馈 bit，然后基站将所有用户的反馈 bit 直接映射为预编码矩阵，可以显著低提高整机性能。文章进一步提出鲁棒化设计策略，并且泛化 DNN 结构使之适应可变用户和可变的反馈 bit。Numerical results show that the DNN-based approach with short pilot sequences and very limited feedback overhead can already approach the performance of conventional linear precoding schemes with full CSI.

7. *Deep Unfolding for Communications Systems: A Survey and Some New Directions,2019*

   本文综述了深度展开的原理，并讨论了其在通信系统中的最新应用，重点讨论了多天线 (MIMO) 无线系统中的检测和预编码以及纠错码的信念传播译码。

8. *Deep reinforcement learning approach to MIMO precoding problem: Optimality and Robustness, 2020*

   强化学习

9. *Model-Driven Deep Learning for Massive Multiuser MIMO Constant Envelope Precoding*

   常包络预编码设计可以显著降低硬件损耗和功率损耗，但是已有的常包络预编码算法计算复杂度高。文章提出使用模型驱动的方式将深度学习与 conjugate gradient algorithm 结合，原始迭代算法展开并添加可训练参数，通过数据学习有效的参数。仿真结果表明该网络能有效抑制多用户干扰和计算复杂度。

10.

# 3 多用户 MIMO 下行系统

## 3.1 系统模型

考虑一个由基站和 $K$ 个相互独立的用户组成的单小区下行多用户 MIMO 系统，其中基站配置 $M$ 根天线，用户 $k$ 配置 $N_k$ 根天线（$k = 1, 2, \cdots, K$）。设基站向用户 $k$ 发送的数据向量为 $\boldsymbol{s}_k = [s_{k1}, s_{k2}, \cdots, s_{kN_k}]^T$，且每个用户的数据向量相互独立，发给每个用户的数据向量都经过一个预编码矩阵 $\boldsymbol{V}_k \in \mathbb{C}^{M \times L_k}$，其中 $L_k$ 为用户 $k$ 所支持的数据流数目，满足 $L_k \leq N_k$，则基站发送的信号为：

$$\begin{aligned} \boldsymbol{x} &= \sum_{i=1}^{K} \boldsymbol{V}_i \boldsymbol{s}_i \\ &= [\boldsymbol{V}_1, \boldsymbol{V}_2, \cdots, \boldsymbol{V}_K] \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \\ \vdots \\ \boldsymbol{s}_K \end{bmatrix} \\ &= \boldsymbol{V}\boldsymbol{s} \end{aligned} \tag{27}$$

假设基站端发射功率限制为 P，那么发送信号满足条件:

$$\begin{aligned} \mathbb{E}\{tr(\boldsymbol{x}\boldsymbol{x}^H)\} &= \mathbb{E}\{tr(\boldsymbol{V}\boldsymbol{s}(\boldsymbol{V}\boldsymbol{s})^H)\} \\ &= \mathbb{E}\{tr(\boldsymbol{V}\boldsymbol{V}^H)\} \\ &= \sum_{i=1}^{K} tr(\boldsymbol{V}_i \boldsymbol{V}_i^H) \leq P \end{aligned} \tag{28}$$

用户 $k$ 的接收信号可以表示为

$$\begin{aligned} \boldsymbol{y}_k &= \boldsymbol{H}_k \boldsymbol{x} + \boldsymbol{n}_k \\ &= \boldsymbol{H}_k \sum_{i=1}^{K} \boldsymbol{V}_i \boldsymbol{s}_i + \boldsymbol{n}_k \end{aligned} \tag{29}$$

其中，$\boldsymbol{H}_k \in \mathbb{C}^{N_k \times M}$ 为基站到用户的信道矩阵,$\boldsymbol{n}_k \sim \mathcal{CN}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_{N_k})$ 表示相应的加性独立同分布（i.i.d）高斯白噪声向量。将所有用户的接收信号统一表

示为

$$\boldsymbol{y}=\boldsymbol{H}\boldsymbol{x}+\boldsymbol{n}=\begin{bmatrix}\boldsymbol{H}_1\\ \boldsymbol{H}_2\\ \vdots\\ \boldsymbol{H}_K\end{bmatrix}\boldsymbol{x}+\boldsymbol{n} \tag{30}$$

式中，$\boldsymbol{H}=\begin{bmatrix}\boldsymbol{H}_1\\ \boldsymbol{H}_2\\ \vdots\\ \boldsymbol{H}_K\end{bmatrix}=\begin{bmatrix}\boldsymbol{h}_{11} & \boldsymbol{h}_{12} & \cdots & \boldsymbol{h}_{1M}\\ \boldsymbol{h}_{21} & \boldsymbol{h}_{22} & \cdots & \boldsymbol{h}_{2M}\\ \vdots & \vdots & \ddots & \vdots\\ \boldsymbol{h}_{K1} & \boldsymbol{h}_{K2} & \cdots & \boldsymbol{h}KM\end{bmatrix}\in\mathbb{C}^{\sum_{k=1}^{K}N_k\times M}$ 表示基站到用户的总信道矩阵，其中元素 $\boldsymbol{h}_{ij}$ 也是一个矩阵，表示第 $j$ 根天线到第 $i$ 个用户的信道矩阵。

假设用户信道是平坦衰落的，基站端可以通过信道估计获得准确的信道状态信息，并且假设移动端知道自己的信道状态信息。下行链路 CSI 的获取：TDD$\Longrightarrow$ 利用上下行信道互译性原理；FDD$\Longrightarrow$CSI 反馈。

将式 30 写为如下形式：

$$\begin{aligned}\boldsymbol{y}_k&=\boldsymbol{H}_k\sum_{i=1}^{K}\boldsymbol{V}_i\boldsymbol{s}_i+\boldsymbol{n}_k\\&=\boldsymbol{H}_k\boldsymbol{V}_k\boldsymbol{s}_k+\boldsymbol{H}_k\sum_{i=1,i\neq k}^{K}\boldsymbol{V}_i\boldsymbol{s}_i+\boldsymbol{n}_k\end{aligned} \tag{31}$$

其中，**第一项表示期望接收信号，第二项表示共信道干扰（CCI）[也成为多用户干扰，MUI]，第三项式噪声项**，对于用户 $k$ 来说第二项和第三项是毫无用处的，会影响用户信号的正常接收，因此线性预编码的主要目标就是在一定的功率限制条件下，使得共信道干扰和噪声项的影响尽量小，最大化期望接收信号。

## 3.2 MRT 预编码

由于可以平衡系统性能和计算复杂度，最大比传输（MRT）预编码【又称作匹配滤波器（MF）预编码】是最简单易实现的预编码算法，通过最大化接收信噪比（SNR）实现。在大规模 MIMO 系统中，当基站天线数 $M$ 足够大时，最简单的 MRT 线性预编码方案便可以得到最优的系统性能。

在发射端已知完美信道状态信息的前提下，MRT 线性预编码矩阵为：

$$\boldsymbol{V} = \beta_{MRT}\boldsymbol{H}^{H} \tag{32}$$

(注：以上编码矩阵后可以加上功率分配矩阵组成整个预编码矩阵)

式中，$\beta_{MRT} = \sqrt{\frac{1}{tr(\boldsymbol{H}\boldsymbol{H}^{H})}}$ 为约束基站发送功率的约束因子。大规模 MIMO 系统中**基站天线数的不断增加使得信道矩阵列向量之间逐渐呈现正交性，即不同终端间的干扰逐渐降低甚至被完全消除**，因此最简单的 MRT 预编码下便可以获得最优的频谱效率和最好的信号传输质量，且复杂度最低。

传统 MIMO 系统中，匹配滤波预编码方案的侧重点在于接收端用户的信号增益最大化，但在多用户系统的场景下，随着传输信道相关性的提升，此方案由于没有考虑如何对用户间的干扰进行处理，将会导致整个系统性能快速下降。

## 3.3 基于广义信道求逆的预编码

### 3.3.1 ZF 预编码

迫零预编码的主要思想是**利用信道的伪逆矩阵形式对发送数据进行编码处理，以达到完全消除其他用户干扰（共信道干扰）的目的**由32可知，为了消除 CCI(即让第二项为 0)，预编码矩阵需满足：

$$\boldsymbol{H}_k\boldsymbol{V}_i = \boldsymbol{0}, i = 1, 2, \cdots, K, i \neq k \tag{33}$$

即，迫零预编码矩阵 $\boldsymbol{V}_k$ 必须满足除目标用户 $k$ 以外的其他用户联合信道矩阵 $\tilde{\boldsymbol{H}}_k = \begin{bmatrix} \boldsymbol{H}_1 \\ \vdots \\ \boldsymbol{H}_{k-1} \\ \boldsymbol{H}_{k+1} \\ \vdots \\ \boldsymbol{H}_K \end{bmatrix}$ 的零空间内，信道矩阵的**伪逆**可以实现以上要求，$H$ 的伪逆矩阵为

$$\boldsymbol{H}^{+} = \boldsymbol{H}^{H}(\boldsymbol{H}\boldsymbol{H}^{H})^{-1} = [\boldsymbol{H}_1^{+}, \boldsymbol{H}_1^{+}, \cdots, \boldsymbol{H}_K^{+}] \tag{34}$$

假设移动端都只配备单根天线，即 $N_k = 1, k = 1, 2, \cdots, K$, 则 $\boldsymbol{H}_k^+ \in \mathbb{C}^{M\times 1}, k = 1, 2, \cdots, K$，为了使得预编码矩阵满足功率约束条件29, 令

$$\boldsymbol{V}_k = \xi \boldsymbol{H}_k^+ \boldsymbol{P}_k^{1/2} \tag{35}$$

式中，功率控制参数满足 $\xi^2 = \frac{P}{tr(\boldsymbol{P}(\boldsymbol{H}\boldsymbol{H}^H)^{-1})}$, $\boldsymbol{P} = diag\{\boldsymbol{P}_1\ \boldsymbol{P}_2\ \cdots\ \boldsymbol{P}_K\}$ 是功率分配矩阵。

### 3.3.2 RZF 预编码

当移动端所处环境中缺少散射体，或者用户之间的距离很近，会导致信道矩阵会因为高度的相关性而成为病态矩阵（欠秩），因而产生噪声放大问题。在低信噪比的情况下，噪声放大问题会严重影响系统性能。正则化迫零预编码则不仅考虑了用户之间的信号干扰，也考虑了噪声问题，所以在性能上优于迫零预编码。

正则化迫零预编码主要是在信道求逆之前加上一个单位阵（此单位阵的作用即为减弱信道矩阵的相关性，避免求逆时产生噪声放大问题），以替代对信道的直接求逆，其编码形式为

$$\boldsymbol{V} = \begin{bmatrix} \boldsymbol{V}_1 & \boldsymbol{V}_2 & \cdots & \boldsymbol{V}_K \end{bmatrix} = \xi \boldsymbol{H}^H (\boldsymbol{H}\boldsymbol{H}^H + \aleph \boldsymbol{I})^{-1} \boldsymbol{P}^{1/2} \tag{36}$$

式中，$\xi^2 = \frac{P}{tr(\boldsymbol{P}\boldsymbol{H}(\boldsymbol{H}^H + M\alpha\boldsymbol{I})^{-1}\boldsymbol{H}^H}$ 为功率归一化参数，$\boldsymbol{V}_k$ 为用户 $k$ 的预编码矩阵，$\boldsymbol{P} = diag\{\boldsymbol{P}_1\ \boldsymbol{P}_2\ \cdots\ \boldsymbol{P}_K\}$ 是功率分配矩阵，$\alpha$ 为正则化因子，用来平衡用户间的干扰与信道噪声。根据最大化信号干扰噪声比（SINR）准则，最优化正则因子为

$$\alpha = K\sigma^2 \tag{37}$$

此时，正则化迫零预编码的性能是最优的，其既考虑了多用户接收信号之间干扰，又考虑了噪声因素的影响，不管是在高信噪比下还是低信噪比下，都有不错的性能表现。

## 3.4 系统模型

# 第三部分 LAtex 语法学习

# 4 Latex 语法学习

## 4.1 伪代码

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

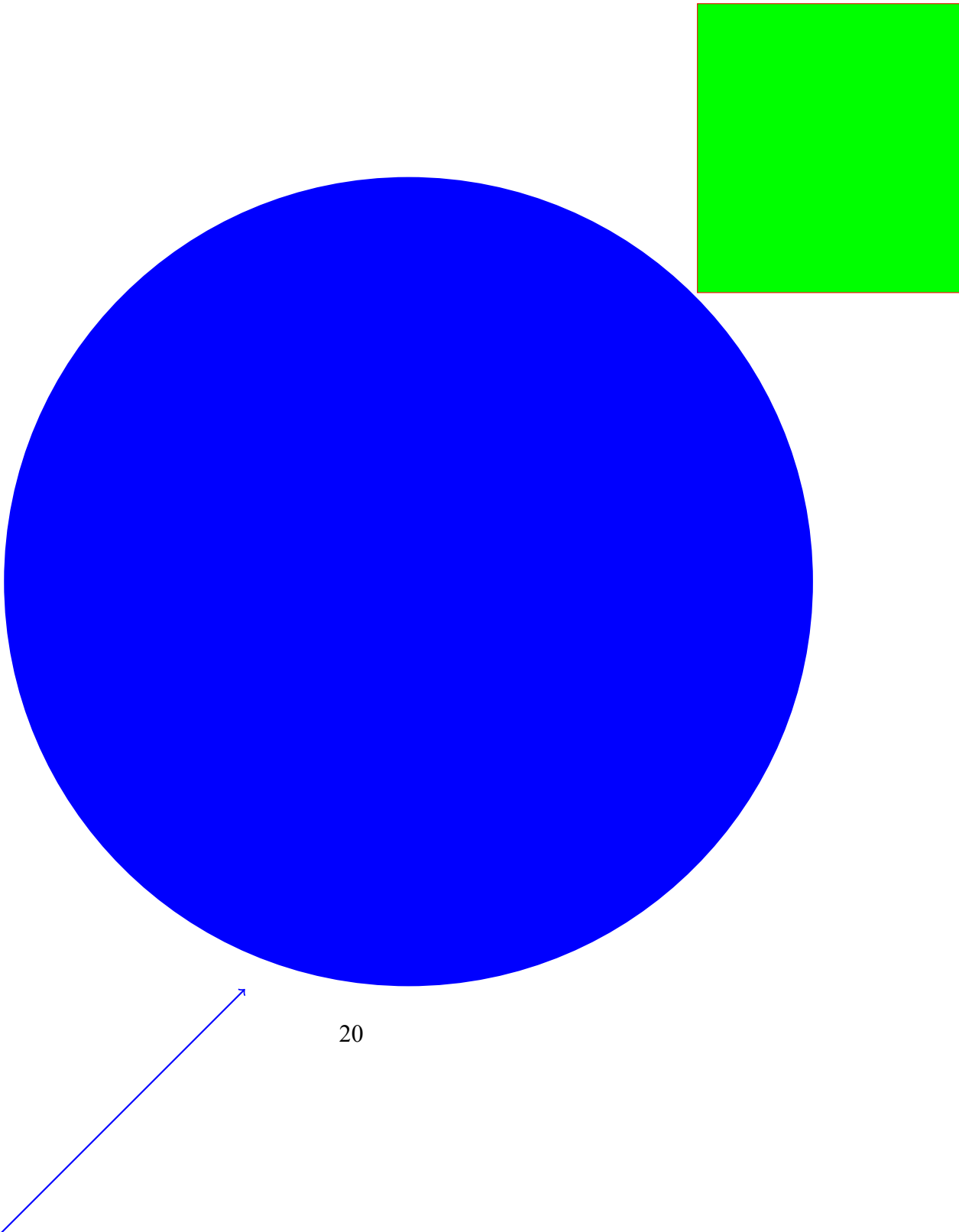
**algorithmic 和 algorithmics** algorithmic 和 algorithmicx，这两个包很像，很多命令都是一样的，只是 algorithmic 的命令都是大写，algorithmicx 的命令都是首字母大写，其他小写 (EndFor 两个大写)。下面是 algorithmic 的基本命令还有 algorithm2e,latex 的与 algorithm 相关的包常用的有几个，algorithm、algorithmic、algorithmicx、algorithm2e，可以大致分成三类，或者说三个排版环境。最原始的是使用 algorithm+algorithmic，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]

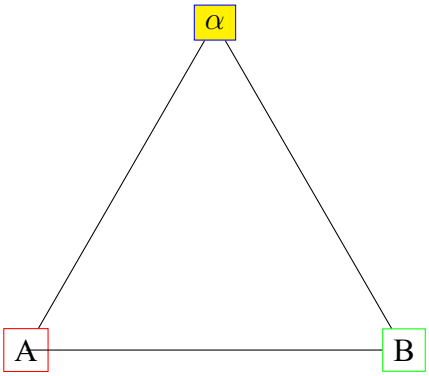
## 4.2 超链接

需要的宏包 hyperref,

### 4.3 插入图片

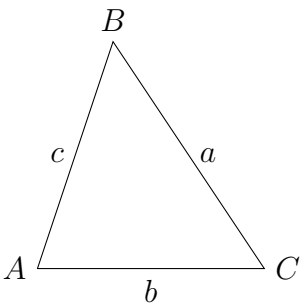
### 4.4 tikz 绘图






三种标准文类的章节命令与层次深度

| 层次深度 | 层次名 | book | report | article |
|---|---|---|---|---|
| -1 | part | \part | \part | |
| 0 | chapter | \chapter | \chapter | \part |
| 1 | section | \section | \section | \section |
| 2 | subsection | \subsection | \subsection | \subsection |
| 3 | subsubsection | \subsubsection | \subsubsection | \subsubsection |
| 4 | paragraph | \paragraph | \paragraph | \paragraph |
| 5 | subparagraph | \subparagraph | \subparagraph | \subparagraph |




```
1 public class Main {
2     public static void main(String[] args)
3     {
4         System.out.println("Hello,World");
5     }
6 }
```

# 5 Latex 字体

## 5.1 Latex 字母的三种类体

正体 R

```
$\rm{R}$
```

花体 1 $\mathcal{R}$

```
$\mathcal{R}$
```

花体 2 $\mathbb{R}$

```
$\mathbb{R}$
```

花体 3 $\mathscr{R}$

```
$\mathscr{R}$
```