


通信原理与计算机网络

滤波器算法


- 高斯滤波
- 巴特沃斯滤波
- IIR滤波器（无限冲激响应滤波器）
 - 直接1型。基本结构。先实现系统函数的零点，再实现极点，需要2N个乘法器和2N个延迟器

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^M b_k \cdot z^{-k}}{1 - \sum_{k=1}^N a_k \cdot z^{-k}} = H_1(z) \cdot H_2(z)$$

 image-20200727145013334

- 直接2型。先实现系统函数的极点，再实现零点。需要N个延迟器和2N个乘法器

$$H(z) = \frac{Y(z)}{X(z)} = H_2(z) \cdot H_1(z)$$

 image-20200727145535362

- 级联型，并联型
- FIR滤波器
 - FIR系统仅在 处有N-1阶极点，在其它地方没有极点，有(N-1)个零点分布在有限Z平面内的任何位置上
 - 直接型

$$\text{差分方程: } y(n) = \sum_{k=0}^{N-1} h(k)x(n-k)$$

$$\text{系统函数: } H(z) = \sum_{n=0}^{N-1} h(n)z^{-n}$$

- 级联型
 - 快速卷积型
 - 线性相位型
 - 频率取样型
- FIR与IIR滤波器的区别

IIR滤波器的系统函数是有理分式，其分母多项式对应于反馈电路。因而他是递归结构的；FIR滤波器的系统函数是有限多项式，属于非递归结构

IIR幅频特性精度很高，不是线性相位的，可以应用于对相位信息不敏感的音频信号上；

FIR幅频特性精度较之于IIR低，但是线性相位，就是不同频率分量的信号经过FIR滤波器后他们的时间差不变。这是很好的性质

802.3 (Ethernet)


- 星型

802.11 (wireless LAN)

- 802.11数据包类型
 - 数据，用来携带更高层次的数据，负责在工作站之间传输数据
 - 管理，管理数据包控制网络的管理功能

- 控制，控制对物理媒介的访问，如光缆等。
- 802.11协议管理帧
 - 数据链路层是一个很靠近底层的通信协议，它使用Bit来表示信息(也使用Bit来标识数据包的开始和结束)，所以数据链路层的协议格式并没有强制要求一个固定的长度，即802.11协议长度是可变的。不同功能的数据帧长度会不一样。这一特性说明mac802.11数据帧显得更加灵活，然而，也会更加复杂。
 - 管理帧包括：beacon帧，探测请求帧，探测响应帧等。控制帧包括：RTS帧，CTS帧，ACK帧，ps-poll帧。
- 四种主要物理组件
 - 工作站（Station），指配备无线网络接口的计算设备，即支持802.11的终端设备
 - 接入点（Access Point），具备无线至有线桥接功能的设备称为接入点，提供接入服务并将802.11MAC帧转换为以太网帧。
 - 无线媒介（Wireless Medium），802.11最初标准化了两种射频物理层和一种红外线物理层
 - 分布式系统（Distribution System），是802.11的逻辑组件，负责将帧传送至目的地，将各个AP连接起来的骨干网络
- 802.11层次和功能
 - MAC
 - 访问机制，分段，加密
 - MAC管理
 - 联盟，鉴权，同步，漫游，MIB管理信息库，电源管理
 - PLCP物理层汇聚协议
 - 空闲信道评估信道（载波监听）
 - PMD物理媒介相关
 - 调制，编码
 - PHY管理
 - 信道管理，MIB管理信息库
 - 站点管理
 - 协调所有管理功能

 image-20200711162941712

 image-20200711161655729

OFDM(正交频分复用技术)

- 采用多个载波，将信道分为多个子信道，将基带码元均匀分散到每个载波上，降低码元速率，减小码间串扰。
- 每个子载波满足公式：

$$f_k = \frac{k}{2T_B}$$

其中 k 为整数， T_B 为码元持续时间最小子载频间隔为：

$$\Delta f = f_k - f_i = \frac{1}{T_B}$$

MIMO技术

- 典型MIMO系统包含 m 个发送天线和 n 个接收天线，如果定义发送天线 i 和接收天线 j 之前的信道为 h_{ij} ，则可以得到 $n \times m$ 的信道矩阵。

$$\begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{bmatrix}$$

- 当收发天线之间的信道为窄带时不变系统时，可以得到MIMO系统接收信号的表现形式为,其中接收信号 Y ，发送信号 X ，噪声为 n ：

$$Y = HX + n$$

- MIMO系统的信道容量由以下公式决定：

$$C = m \log_2 \left(\frac{n}{m} \times \frac{S}{N} \right)$$

其中普通信道的香农公式为：

$$C = B \log_2 \left(1 + \frac{S}{N} \right)$$

以上参数的意义为：C：信道容量（bps），B：信道的频带宽度，S/N：信噪比，m：发送天线的个数，n：接收天线的个数。

- SU-MIMO和MU-MIMO

如果MIMO系统用于增加一个用户的速率，称之为单用户MIMO（SU-MIMO）。多用户MIMO（MU-MIMO）：如果每个独立的数据流分配给不同的用户，称之为多用户。这种模式主要对上行链路有用。从UE的复杂程度和体积来看，每个UE只能有一个发射天线，因此称之为“协同MIMO”

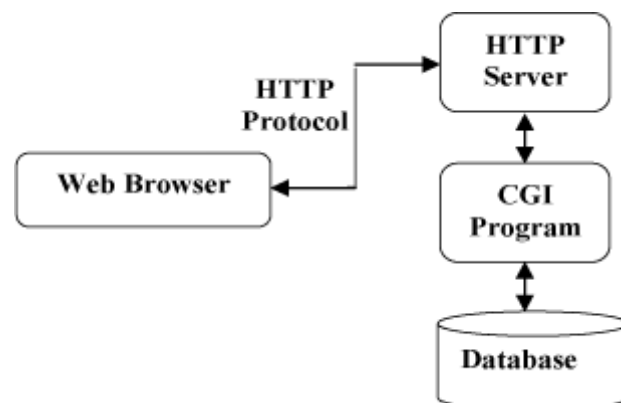
- Massive MIMO（大规模天线技术）

传统的MIMO我们称之为2D-MIMO，以8天线为例，实际信号在做覆盖时，只能在水平方向移动，垂直方向是不动的，信号类似一个平面发射出去，而Massive MIMO，是信号水平维度空间基础上引入垂直维度的空域进行利用，信号的辐射状是个电磁波束。所以Massive MIMO也称为3D-MIMO。

HTTP与HTTPS

- HTTP简介

- 无连接，无连接的含义是限制每次连接只处理一个请求
- 媒体独立，任何类型的数据都可以通过HTTP发送
- 无状态，对于事务没有记忆能力



- 常见http状态码

- 200，请求成功
- 301，资源被永久转移到其他URL

- 404, 请求的资源不存在
- 500, 内部服务器错误
- HTTP请求方法
 - HTTP1.0, GET, POST, HEAD,
 - GET, 请求指定页面信息, 并返回实体
 - HEAD, 请求指定页面信息的报头
 - POST, 向指定资源提交数据并处理请求,
 - HTTP1.1新增, OPTIONS、PUT、PATCH、DELETE、TRACE 和 CONNECT 方法
- HTTPS
 - 就是在HTTP的基础上增加了数据加密, 是密文传输
 - HTTPS请求的过程需要CA证书要验证身份以保证客户端请求到服务器端之后, 传回的响应是来自于服务器端, 而HTTP则不需要CA证书。
 - 对称加密: 解密和加密采用相同的加密算法
 - 非对称加密: 需要两个密钥, 公有密钥和私有密钥。加密密钥和解密密钥成对出现。
 - 摘要算法: 数字摘要采用单项Hash函数将需要加密的明文“摘要”成一串固定长度(128位)的密文, 这一串密文又称为数字指纹, 它有固定的长度, 而且不同的明文摘要成密文, 其结果总是不同的, 而同样的明文其摘要必定一致。“数字摘要”是https能确保数据完整性和防篡改的根本原因

TCP/UDP

- TCP三次握手四次挥手
 - 三次握手的过程
 - 1) 主机A向主机B发送TCP连接请求数据包, 其中包含主机A的初始序列号 $\text{seq}(A)=x$ 。(其中报文中同步标志位 $\text{SYN}=1$, $\text{ACK}=0$, 表示这是一个TCP连接请求数据报文; 序号 $\text{seq}=x$, 表明传输数据时的第一个数据字节的序号是 x) ;
 - 2) 主机B收到请求后, 会发回连接确认数据包。(其中确认报文段中, 标识位 $\text{SYN}=1$, $\text{ACK}=1$, 表示这是一个TCP连接响应数据报文, 并含主机B的初始序列号 $\text{seq}(B)=y$, 以及主机B对主机A初始序列号的确认号 $\text{ack}(B)=\text{seq}(A)+1=x+1$)
 - 3) 第三次, 主机A收到主机B的确认报文后, 还需作出确认, 即发送一个序列号 $\text{seq}(A)=x+1$; 确认号为 $\text{ack}(A)=y+1$ 的报文;
 - 四次挥手过程

假设主机A为客户端, 主机B为服务器, 其释放TCP连接的过程如下:

 - 1) 关闭客户端到服务器的连接: 首先客户端A发送一个FIN, 用来关闭客户到服务器的数据传送, 然后等待服务器的确认。其中终止标志位 $\text{FIN}=1$, 序列号 $\text{seq}=u$ 。
 - 2) 服务器收到这个FIN, 它发回一个ACK, 确认号 ack 为收到的序号加1。
 - 3) 关闭服务器到客户端的连接: 也是发送一个FIN给客户端。
 - 4) 客户端收到FIN后, 并发回一个ACK报文确认, 并将确认序号 seq 设置为收到序号加1。首先进行关闭的一方将执行主动关闭, 而另一方执行被动关闭。
- TCP与UDP
 - TCP面向连接, UDP是无连接的
 - TCP 提供可靠的服务, 也就是说, 通过 TCP 连接传送的数据, 无差错, 不丢失, 不重复, 且按序到达; UDP 尽最大努力交付, 即不保证可靠交付
 - TCP 的逻辑通信信道是全双工的可靠信道; UDP 则是不可靠信道
 - 每一条 TCP 连接只能是点到点的; UDP 支持一对一, 一对多, 多对一和多对多的交互通信

- TCP 面向字节流（可能出现黏包问题），实际上是 TCP 把数据看成一连串无结构的字节流；UDP 是面向报文的（不会出现黏包问题）
 - UDP 没有拥塞控制，因此网络出现拥塞不会使源主机的发送速率降低（对实时应用很有用，如 IP 电话，实时视频会议等）
 - TCP 首部开销20字节；UDP 的首部开销小，只有 8 个字节
- TCP可靠的原因：
 - 三次握手，超时重传，滑动窗口，拥塞控制。
- TCP黏包问题
 - TCP是一个面向字节流的传输服务，数据没有边界，可能出现两个数据包黏在一起的情况。
 - 解决：1，发送定长包，2，包头加上包体长度，3，设置边界符号，4，使用更加复杂的应用层协议
- 流量控制
 - 接收方根据接收窗口的空闲缓存来通知（advertisedWindowSize）发送方控制发送窗口，使得发送方发送的数据量不大于接收方的接收窗口，已达到流量控制的目的。如果接收方read和receive的速度一样快，那么接收窗口发送给advertisedWindowSize的大小为MaxRecvBuffer，发送方获得advertisedWindowSize之后会计算出发送方可以发送数据的EffectiveWindowSize,这个值表征了发送方目前可以发送多少数据。

如果接收方read的速度比recevie的速度慢，就会造成接收窗口的逐渐减小，因此发送方收到的advertisedWindowSize的大小也在减小，当advertisedWindowSize减小到0，说明接受方不能收取任何数据，意味着TCP将发送进程阻塞。
- 拥塞控制
 - 拥塞标志：TCP协议认为超时是拥塞的标志
 - 慢启动和快速恢复
 - 慢启动：是在网络发生拥塞时，将慢启动阈值置为当前拥塞窗口的一半，拥塞窗口置为1，采用乘性增加性减的方式进行快速启动的方式
 - 快重传：丢包并不意味着拥塞发生，丢包之后等待超时并慢启动这一过程会降低网络性能，如果在丢包之后可以立即解决而不用慢启动则可以避免网络性能的降低。发送方一旦收到三个连续的重复确认，就对相应的报文段立即重传，而不是等待该报文段的超时重传计时器超时再重传。
 - 快恢复：发生丢包之后，快速重传和快速恢复同时发生，快速恢复是指，将慢启动阈值置为当前拥塞窗口的一半，将拥塞窗口置为慢启动阈值+3，并使用加性增的手段。

 image-20200716153632698

DNS

- 在浏览器输入域名时，操作系统检查本地host是否具备网址映射关系，如果有则调用该IP地址映射，完成解析
- 若本地host没有，则查找本地DNS解析器缓存，
- 若以上两个都没有，则查找TCP/IP参数中设置的首选DNS服务器，称为本地DNS服务器。若所需查找的域名包含在本地配置资源中，则返回解析结果，此解析具有权威性。若不包含在本地资源配置中，只是在缓存中，则该解析结果不具有权威性
- 若以上都失效
 - 若没用转发模式，则将请求发送到13台根DNS，若根DNS没有查到则转发到下一个根DNS，直到查到为止
 - 若使用了转发模式，此DNS服务器就会把请求转发至上一级DNS服务器，由上一级服务器进行解析，上一级服务器如果不能解析，或找根DNS或把转请求转至上上级，以此循环。不管是本地DNS服务器用的是转发，还是根提示，最后都是把结果返回给本地DNS服务器，由此DNS服务器再返回给客户机

CGI

- CGI（Common Gateway Interface）公共网关接口，是外部扩展应用程序与 Web 服务器交互的一个标准接口

ARP

- 地址解析协议，即ARP（Address Resolution Protocol），是根据IP地址获取物理地址的一个TCP/IP协议。主机发送信息时将包含目标IP地址的ARP请求广播到局域网络上的所有主机，并接收返回消息，以此确定目标的物理地址

ICMP

- ICMP（Internet Control Message Protocol）Internet控制报文协议。它是TCP/IP协议簇的一个子协议，用于在IP主机、路由器之间传递控制消息。控制消息是指网络通不通、主机是否可达、路由是否可用等网络本身的消息。这些控制消息虽然并不传输用户数据，但是对于用户数据的传递起着重要的作用。

TCP/IP协议族

- IP
- ARP
- RARP
- IGMP
- ICMP

OSI网络模型

- MAC层与物理层
 - 物理层为数据链路层提供比特流传输服务
 - 数据链路层传输数据帧
- 应用层协议：
 - HTTP FTP TFTP SMTP SNMP DNS TELNET HTTPS POP3 DHCP
- 表示层，会话层
- 传输层：TCP，UDP
- 网络层：ICMP，IGMP

编码

- NRZ（不归零），1->高信号，0->低信号
- NRZI（不归零反转），1->信号跳变，0->信号保持
- Manchester，1->高到低跳变,0->低到高跳变

可靠传输

- 循环冗余校验（CRC）
- 停止等待协议
- 连续ARQ
- 滑动窗口协议

p2p与e2e

- 端到端与点到点是针对网络中传输的两端设备间的关系而言的。
 - 端到端传输指的是在数据传输前，经过各种各样的交换设备，在两端设备间建立一条链路，就像它们是直接相连的一样，链路建立后，发送端就可以发送数据，直至数据发送完毕，接收端确认接收成功。

- 点到点系统指的是发送端把数据传给与它直接相连的设备，这台设备在合适的时候又把数据传给与之直接相连的下一台设备，通过一台一台直接相连的设备，把数据传到接收端。

MAC层调度算法

• ZFS调度算法

- ZFS的中心思想是每次增加一个使得总容量最大的用户，直至用户总容量下降或者选满M个用户
- 注释：U，全体用户集合。 h_u 是用户u的信道增益， s_i 是第i步选中的用户。 S_i 是第i步迭代以后选择的用户集合，i是已选择用户总数，假设基站的总数为M，总用户数为K（K>M）
- 算法流程：

■ 1，初始化

$$\begin{aligned} i &= 1 \\ s_1 &= \arg \max_{x \in U} h_x h_x^H \\ S_1 &= \{s_1\} \end{aligned}$$

定义基站和集合 S_1 的用户使用ZFBF编码进行MU-MIMO传输的总容量为 $R_{zf}(S_1)_{max}$ 。

2，当i<M时新增一个用户

$$\begin{aligned} i &\leftarrow i + 1 \\ s_i &= \arg \max_{u \in U/S_{i-1}} R_{zf}(S_{i-1} \cup \{u\}) \end{aligned}$$

令 $S_i = S_{i-1} \cup s_i$ ，集合 S_i 的总容量为 $R_{zf}(S_i)_{max}$ 。如果 $R_{zf}(S_i)_{max} > R_{zf}(S_{i-1})_{max}$ ，则继续步骤2新增用户，如果 $R_{zf}(S_i)_{max} \leq R_{zf}(S_{i-1})_{max}$ 则 $i \leftarrow i - 1$ ，进入步骤3用户选择结束阶段。

3，用户选择结束阶段

4，计算ZFBF预编码矩阵和功率分配

$$W = H_{S_i}^H (H_{S_i} H_{S_i}^H)^{-1}$$

• SUS

- SUS算法是基于信道垂直偏量的F范数的低复杂度用户选择呢算法。
- 定义： S_0 为SUS算法选择的用户集合， \mathbf{g}_k ，是用户k的信道增益 \mathbf{h}_k 在已选用户集合的信道空间的垂直补空间上的投影， $\pi(i)$ 是第i轮递推中选中的用户。 $T(i)$ 是第i轮地推中的备选用户集合，假设基站天线数为M，用户总数为K，（K>M），SUS算法流程如下

■ 步骤1，初始化

$$\begin{aligned} i &= 1 \\ T_1 &= 1, 2, \dots, K \\ S_0 &= \emptyset \end{aligned}$$

- 步骤2，对备选用户集合 T_i 中的每个用户k计算 \mathbf{g}_k ，即 \mathbf{h}_k 垂直于子空间 $span\{\mathbf{g}_{(1)}, \mathbf{g}_{(2)}, \dots, \mathbf{g}_{(i-1)}, \}$ 。

$$\begin{aligned} \mathbf{g}_k &= \mathbf{h}_k - \sum_{j=1}^{i-1} \frac{\mathbf{h}_k \mathbf{g}_{(j)}^H}{\|\mathbf{g}_{(j)}\|^2} \mathbf{g}_{(j)} \\ &= \mathbf{h}_k (1 - \sum_{j=1}^{i-1} \frac{\mathbf{g}_{(j)}^H}{\|\mathbf{g}_{(j)}\|^2} \mathbf{g}_{(j)}) \end{aligned}$$

备注：当i = 1时，以上公式为 $\mathbf{g}_k = \mathbf{h}_k$ 。

- 步骤3，选中第i个用户

$$\pi(i) = \underset{k \in T_i}{\operatorname{argmax}} \|\mathbf{g}_k\|$$

$$S_0 \leftarrow S_0 \bigcup \{\pi(i)\}$$

$$\mathbf{h}_{(i)} = \mathbf{h}_{\pi(i)}$$

$$\mathbf{g}_{(i)} = \mathbf{g}_{\pi(i)}$$

- 步骤4, 如果 $S_0 < M$, 选择与 $\mathbf{g}_{(i)}$ 准正交的用户作为下一轮迭代的备选用户

$$T_{i+1} = \{k \in T_i, k \neq \pi(i) \mid \frac{\mathbf{h}_k \mathbf{g}_{(i)}^H}{\|\mathbf{h}_k\| \|\mathbf{g}_{(i)}\|} < \alpha\}$$

$$i \leftarrow i + 1$$

α 是一个收信道强度，信道相关性，基站天线数M，总用户数K以及总发送功率等多方面因素影响的值。如果 T_{i+1} 非空，而且已选用户集合的势满足 $S_0 < M$ 则进入步骤2；否则选择过程结束，输出 S_0 为选择服务的用户集合