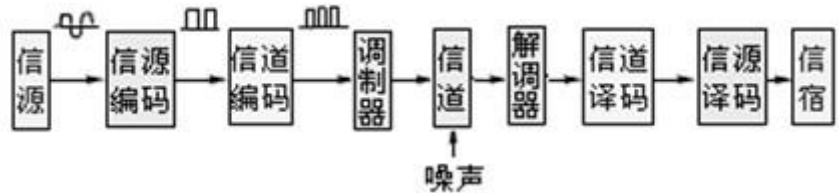


信号处理过程浅析

大家看这个信号处理的流程图，没有基础的同学可能会一头雾水，本文将一点点把这个图拆开，尽量让大家有一个直观的理解。



1. 信号处理方法

研究信号，研究信号传输，首先要清楚信号的分析方法，一个原始信号是一个波形，这个波形是随着时间变化的，是时域的表达，在时域对这个信号进行分析和处理有时候非常困难，那我们伟大的数学家或者说物理学家，则发现了另一个角度来观察分析这个时域的信号，就叫做频域。他们还发现了换算两种域的数学方法，那就是傅里叶分析。

那么我们就可以知道了：

通信的基础理论是信号分析的两种方法：**1 是将信号描述成时间的函数，2 是将信号描述成频率的函数**。也有用时域和频率联合起来表示信号的方法。时域、频域两种分析方法提供了不同的角度，它们提供的信息都是一样，只是在不同的时候分析起来哪个方便就用哪个。

2. 时域频域

那怎么理解时域与频域的概念呢？

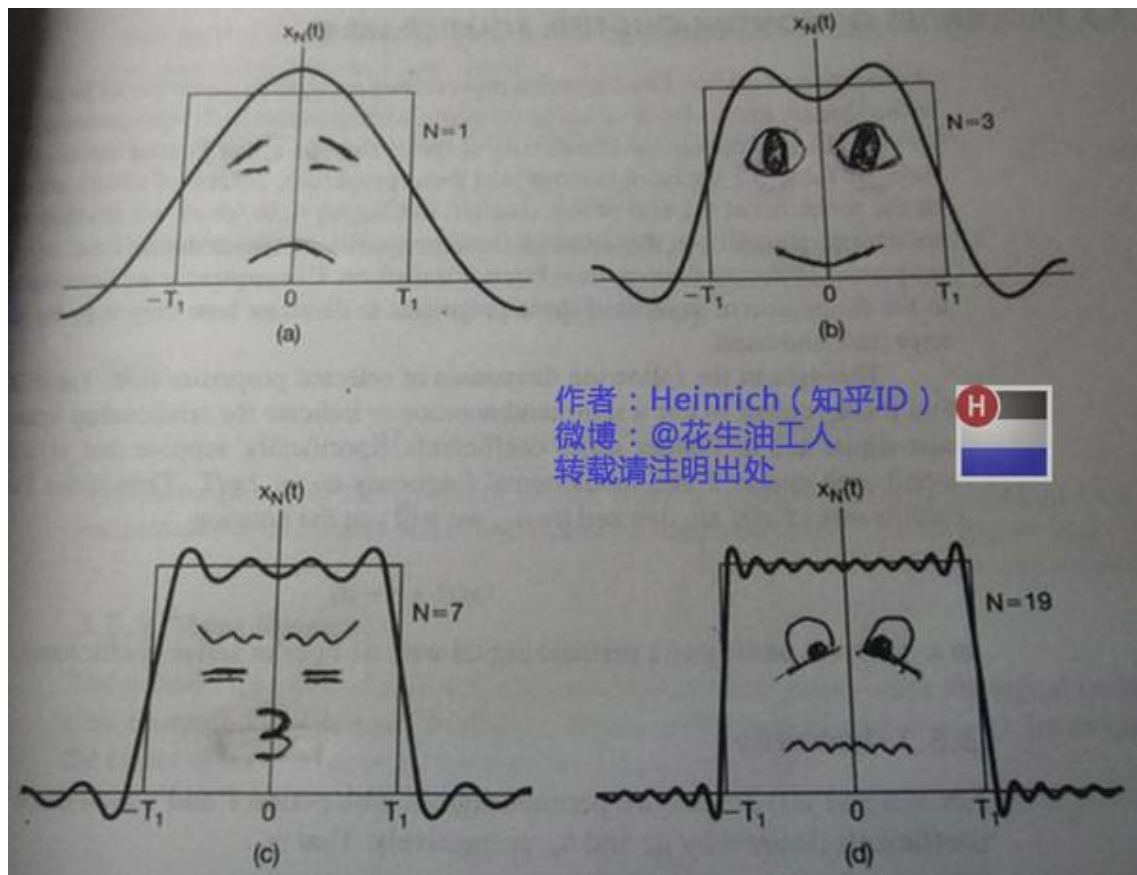
从我们出生，我们看到的世界都以时间贯穿，股票的走势、人的身高、汽车的轨迹都会随着时间发生改变。这种以时间作为参照来观察动态世界的方法我们称其为时域分析。而我们也想当然的认为，世间万物都在随着时间不停的改变，并且永远不会静止下来。但如果我告诉你，用另一种方法来观察世界的话，你会发现世界是永恒不变的，这个静止的世界就叫做频域。**频域最重要的性质是：它不是真实的，而是一个数学构造。时域是惟一客观存在的域，而频域是一个遵循特定规则的数学范畴。**

傅里叶分析就是一种时域和频域之间转换的数学方法。傅里叶分析可分为傅里叶级数（Fourier Serie）和傅里叶变换(Fourier Transformation)。

既然我们接受了频域的存在，那么频域是什么样子的呢，这时就必须提到神奇的正弦波**正弦波是频域中唯一存在的波形，这是频域中最重要的规则，即正弦波是对频域的描述，因为时域中的任何波形都可用正弦波合成**。这是正弦波的一个非常重要的性质。

比如，一个时域矩形的波形，是如何被正弦波合成的呢，如下图：

（知乎的一位大神分析的很形象，转载借鉴给大家~）



第一幅图是一个郁闷的正弦波 $\cos(x)$

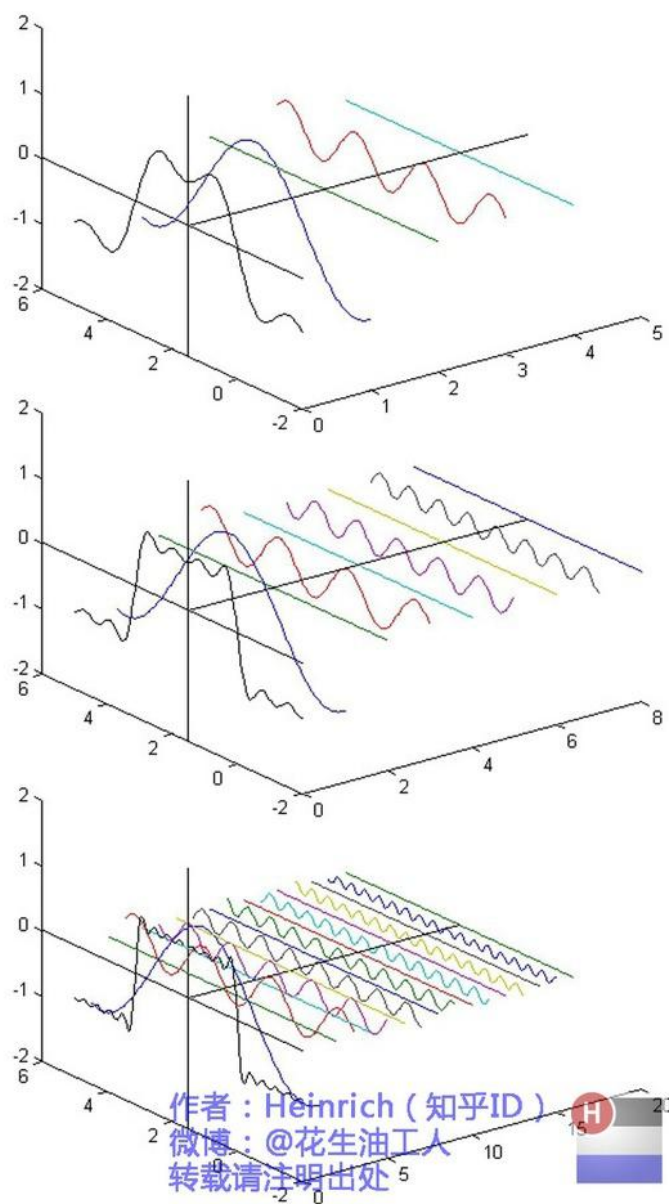
第二幅图是 2 个卖萌的正弦波的叠加 $\cos(x) + a \cdot \cos(3x)$

第三幅图是 4 个发春的正弦波的叠加

第四幅图是 10 个难受的正弦波的叠加

随着正弦波数量逐渐的增长，他们最终会叠加成一个标准的矩形

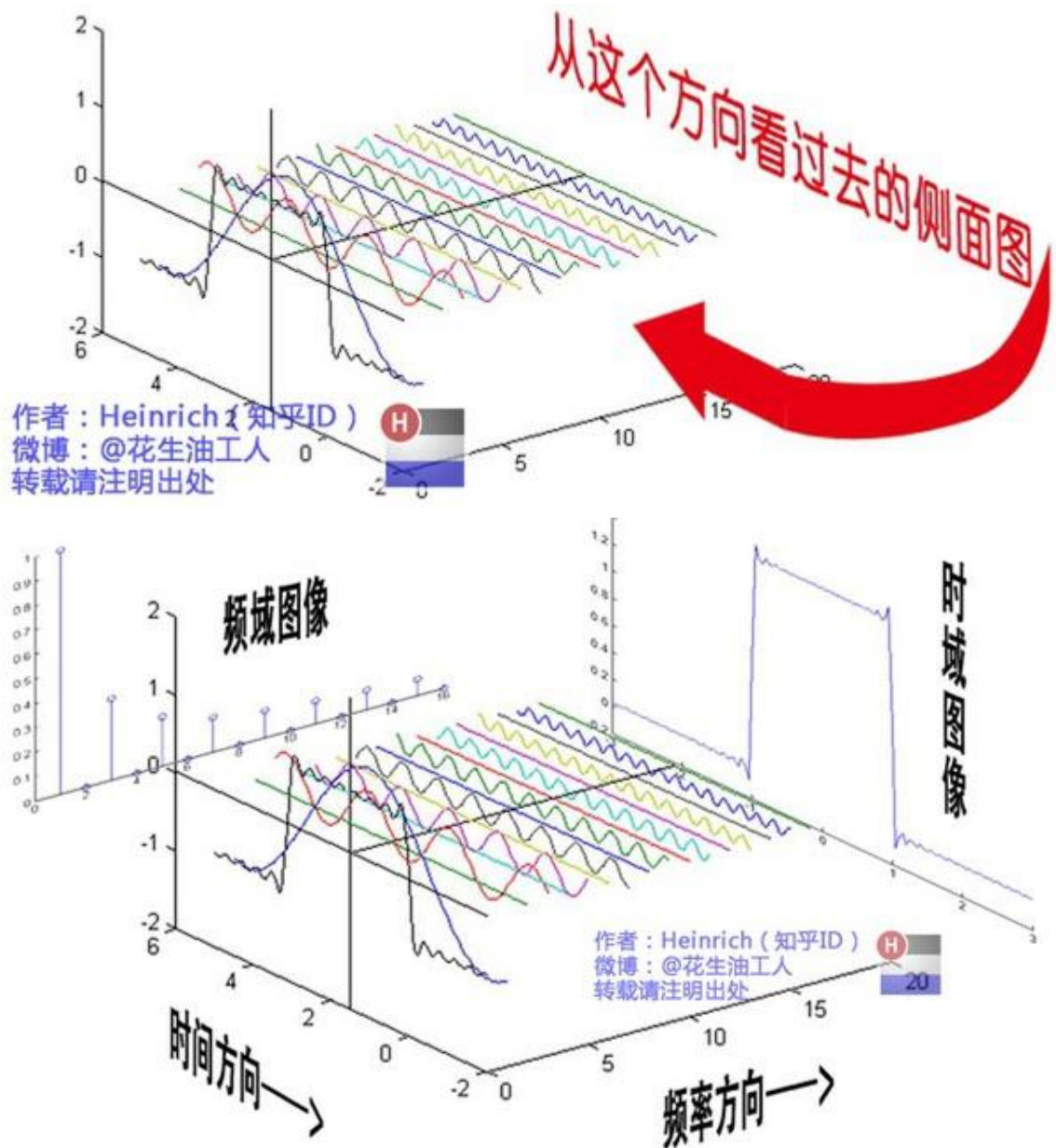
不仅仅是矩形，你能想到的任何波形都是可以如此方法用正弦波叠加起来的。
换个角度更清晰，下图将这个矩形波拆成了正弦波，从侧面看更清晰。



在这几幅图中，最前面黑色的线就是所有正弦波叠加而成的总和，也就是越来越接近矩形波的那个图形。而后面依不同颜色排列而成的正弦波就是组合为矩形波的各个分量。这些正弦波按照频率从低到高从前向后排列开来，而每一个波的振幅都是不同的。

那么矩形波在频域里是什么样子呢？

居然就是这些正弦波形在另一个坐标轴看过去的投影。

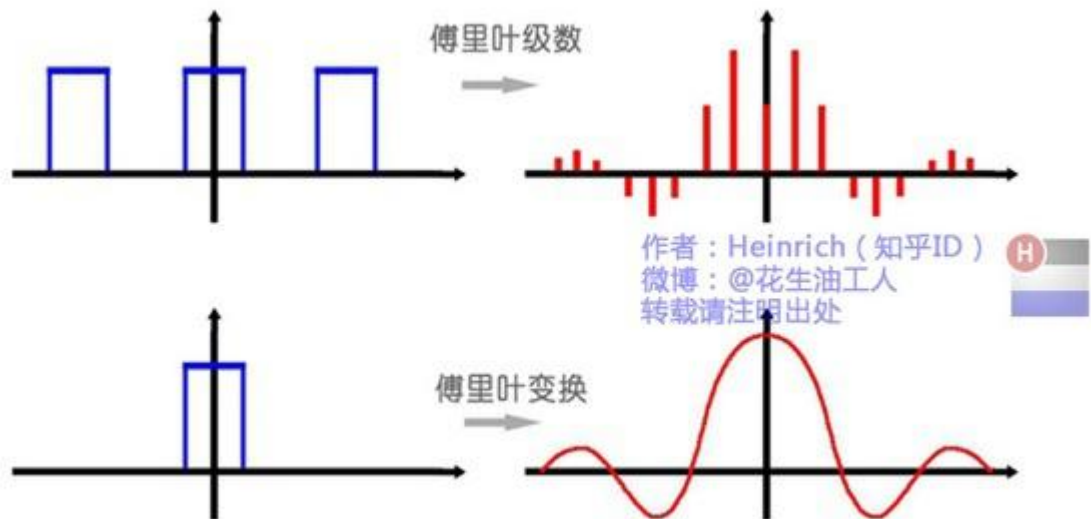


傅里叶分析分为两种，一种是傅里叶级数，一种是傅里叶变换。

傅里叶级数的本质是将一个周期的信号分解成无限多分开的（离散的）正弦波。

傅里叶级数，在时域是一个周期且连续的函数，而在频域是一个非周期离散的函数。例如上图所说的矩形波。

傅里叶变换，则是将一个时域非周期的连续信号，转换为一个在频域非周期的连续信号。



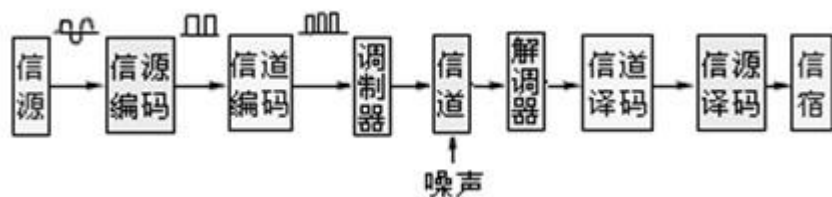
换一个角度理解：傅里叶变换实际上是对一个周期无限大的函数进行傅里叶变换。如果在时域这个周期信号的周期越大，在频域的竖线的间距就越小，到了极限的话，时域的非周期的波形可以看做周期是无限大，那么在频域的竖线的间距就无限小，就会变成连续的曲线了。

基本傅里叶变换和神奇的正弦波先说这么多，还有一个相位的分析先不介绍了，大家有兴趣可以点链接，这个博客里介绍了更多更详细的内容：

<https://blog.csdn.net/l494926429/article/details/51818012>

3. 载波

了解了信号的处理方法，那么接下来看下信号的整个处理流程



最基本的，看框图的首尾两端，信号要从发射端发射，然后接收端接收，这时就出现了一个问题，原始信号一般频率都很低，自己发不出去，那么就只能借助周期性的高频信号帮助他，“骑乘”在高频信号上一起发出去，这个过程就叫调制，这个周期性高频信号就叫载波。就像卫星自己不能去太空，火箭要搭载卫星，货物自己不能到达目的地，要用货车来搭载一样。

而货车搭载货物的方式，就是不同的调制方式，例如，可能是大货车小货车在路上不断交替，也可能是多个货车同时出发，可能是不同型号，不同颜色的货车走不同路线，最终到达目的地，这些运输过程上的区别就可以理解为不同的调制方式。

4.调制

调制所用的载波应该是什么样最好呢？没错，就是正弦波。之前讲过正弦波的神奇性质，而且还要分析信号，经常要在频域进行，而频域只有正弦波，因此载波都是正弦波。

正弦波的三要素高中也学过，就是振幅，相位，频率。这三点不同的千变万化的正弦波的各种组合就能构成多种多样的调制方式。

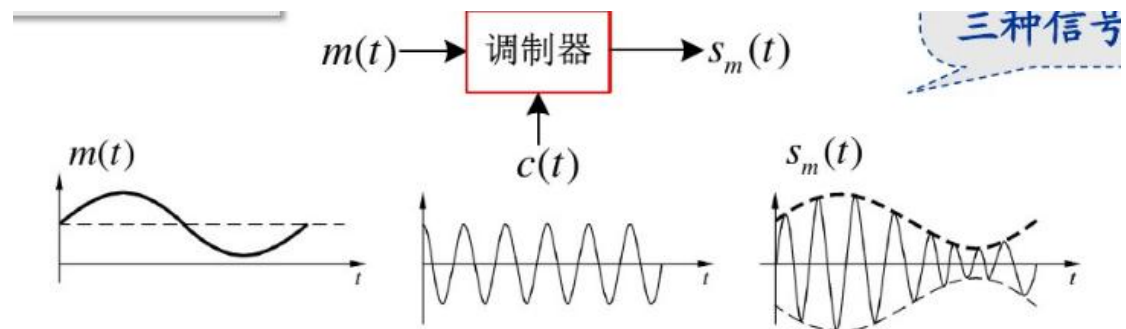
解调是什么呢，同理，就是卸载货物的过程了，将信号和载波分离开的过程。

调制分为两种，用模拟信号进行调制和数字信号进行调制。

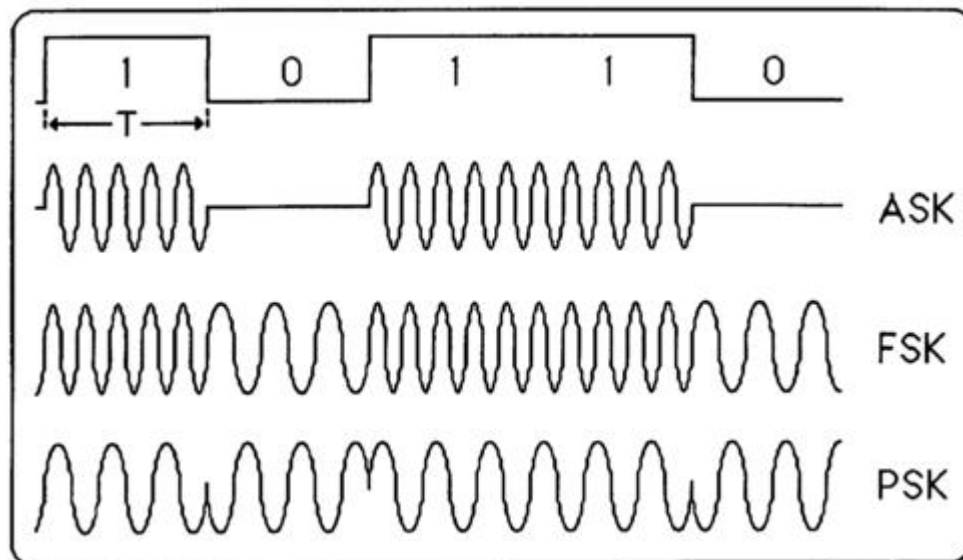
（简单理解为模拟是连续的波形，数字是被处理成 1 和 0 的数字的序列组合）

模拟调制分为：幅度调制（AM），频率调制（FM），相位调制（PM）

下图为幅度调制举例：



数字调制分为：振幅键控（2ASK），频移键控（2PSK，QPSK，8PSK），相移键控（2FSK），正交幅度调制（QAM）

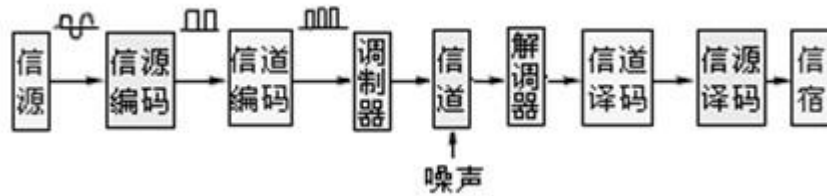


多种数字调制的原理大家可以参考这篇博客：

<https://blog.csdn.net/gzydominating40/article/details/53456352>

5. 编码

现在我们再看一下信号的传输系统模型，发现在调制解调的两侧还有一个编解码过程，分为信源编码和信道编码。



信源编码：主要是利用信源的统计特性，解决信源的相关性，去掉信源冗余信息，从而达到压缩信源输出的信息率，提高系统有效性的目的。第三代移动通信中的信源编码包括语音压缩编码、各类图像压缩编码及多媒体数据压缩编码。

信道编码：为了保证通信系统的传输可靠性，克服信道中的噪声和干扰的。它根据一定的（监督）规律在待发送的信息码元中（人为的）加入一些必要的（监督）码元，在接受端利用这些监督码元与信息码元之间的监督规律，发现和纠正差错，以提高信息码元传输的可靠性。信道编码的目的是试图以最少的监督码元为代价，以换取最大程度的可靠性的提高。我们主要研究的是信道编码。

简单地说就是信号被信源编码处理成 1 和 0 的数字组合进行发射接收后，由于经过复杂的信道，经常会不可避免的出现传输错误，信道编码就是为了发现甚至纠正这种错误，提高信息传输的可靠性。

随着信道编码的研究，目前正在使用的编码方式有 BCH 码，RS 码，卷积码，turbo 码，以及最近几年很火的 LDPC 码，每一种码都有自己的优缺点。通信原理的课本教材会简单介绍这些编码方式的实现原理。

6. 交织

有很多通信系统在编码与调制中间还有一个交织和解交织过程，那么交织又是什么呢？

编码过程会发现信号传输过程中的错误，但是有时错误的产生是有记忆的，就是说前一个错误跟后一个错误之间是有关联的，可能会导致错误一连串出现。那么**交织是为了将这种突发的有记忆的错误的造成独立的随机无记忆的错误。**

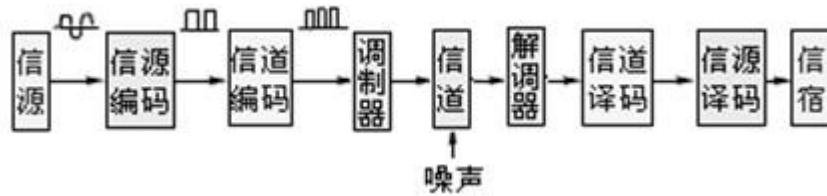
目前最常用的方法就是将信息送入交织器，将交织器设计成按列写入，但是却按行读取的矩阵存储器。这样输出的信息的错误就会被随机打乱，去掉了记忆性。

7. 信道

到现在我们在看传输系统的框图，最后一个点就是信道，如果我们不做专门的信道研究，都是默认信道是**高斯白噪声的信道，是最基本的噪声与干扰模型。它的幅度分布服从高斯分布，而功率谱密度是均匀分布的**，处理起来是最简单的。实际上信息传输过程信道类型很多，非常复杂，很多研究者也都在信道方面进行研究。

8.总结

最后我们再看一眼框图



信号传输

- (1) 先用信源编码将信号化成数字形式，
 - (2) 为了防止在传输过程出现错误，通过信道编码进行处理，给原来的码加一些冗余的监督码
 - (3) 再进入调制器，与载波进行结合，一起发射出去
- 补充：调制还有一个重要作用可以抗干扰，因为干扰的信号一般频率都比我们要传输的信号频率高很多，通过载波调制，可以在频域上将信号与噪声拆开，再用频率滤波器把噪声所在的频率过滤，只留下我们需要的信号，就可以达到一定程度上抗干扰的目的。
- (4) 走过含有各种噪声的信道
 - (5) 到达接收部分，进入解调器，将信号与载波分离开
 - (6) 通过译码还原信号，这个过程必不可少还有错误，不可能百分之百都正确，那么研究编解码的目的最主要也就是提高这种传输的可靠性，想尽办法降低误码率。
 - (7) 最后信源译码，恢复成开始的连续波形，完成整个信号传输的过程。

到现在我们已经把这整个传输过程框图都拆开了，希望对大家的理解有所帮助。

后记：文章写的很仓促，也比较浅显，很多概念内容网上写的解释都很书面化，本文加了很多个人的理解，肯定也有很多不严谨不完善的地方，希望没有相关基础的同学能一起交流，有学过相关知识的同学随时补充纠正~同时也希望本文能对其他小伙伴的相关分享有所帮助，我们可以随时交流(*^▽^*)