

通信系统仿真

第1章 仿真的作用

何晨光

哈尔滨工业大学

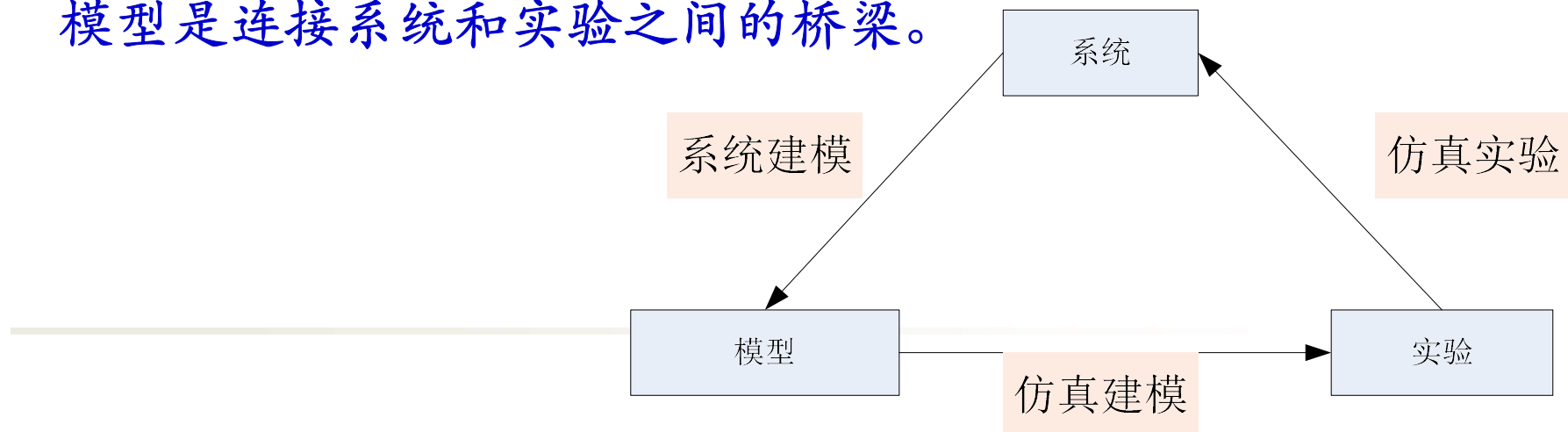
电子与信息工程学院

Communication Research Center

通信系统仿真基本概念

以通信系统理论、随机过程与统计学理论和优化理论为基础，以计算机和仿真软件为工具，对现实通信系统或未来通信系统进行实验研究的理论和方法。

- 系统仿真由三个要素构成：系统、模型、实验
- 系统是问题的本源；
- 实验是解决问题达到目的的手段；
- 模型是连接系统和实验之间的桥梁。





应用系统仿真的情况

“最后的方法”

1. 不存在完整的数学公式，或者还没有一套解答数学模型公式的方法。
2. 需要对系统或过程进行长期运行的比较。
3. 系统还处于设计阶段，并没有真正建立起来，进行多个方案的比较择优
4. 在真实系统上做实验会破坏系统的运行。
5. 需做多次试验时，很难保证每次的操作条件都相同，因而无法对试验结果做出正确的判断。
6. 试验费用太大或有危险性。



Why 通信系统计算机仿真

- 现代通信系统的复杂性促进了仿真的广泛使用
 - 通信系统的结构、系统运行时所处的环境
 - 系统特性：功率和带宽有限
 - 系统要求：高速数据
 - 复杂的调制和脉冲成形技术，差错控制、接收端的高级信号处理技术。
 - 高速数据→同步要求变得更严格，接收机也更复杂
 - 更恶劣的环境中，多跳非线性放大器
 - 干扰
 - 多径和阴影→衰落。
 - 不能够解析处理

设计和分析问题使用传统的（不基于仿真的）方法不再是易于解析处理的。



Why 通信系统计算机仿真

- 计算机不仅功能强大，而且价格便宜
- 计算机方法的发展
- 另一个动机：深入理解系统特性的有价值的工具。
- 好的仿真跟在实验室实现一个系统很类似，多点测量，参数研究，可以任意改动滤波器带宽和信噪比（SNR）等参数，而且还能很快地观测到这些改变对系统性能的影响。
- 时域波形、信号谱图、眼图、信号星座图、直方图。仿真能更容易也更经济地对各种假设情况进行研究。

R.W. Hamming: 仿真的主要作用不在于获得数值而在于获得深入的理解。

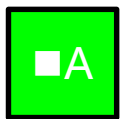


Why 这门课程

- 前面的原因：仿真的好处与重要性
- 已经学习过通信原理、无线通信、移动通信、matlab还有必要吗？
 - 仿真的可信度？
 - 仿真的基本概念
 - 仿真工具是基础，方法论是重点
- 进入科研研究的需要
 - 选题、仿真、硬件等
- 定位物理层、通信信号处理

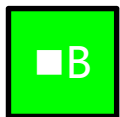
锻炼系统仿真的能力
巩固对专业知识的理解
为毕业设计做准备

测试题02：系统仿真由哪三个要素构成？



A

系统



B

模型



C

实验



D

模块

提交

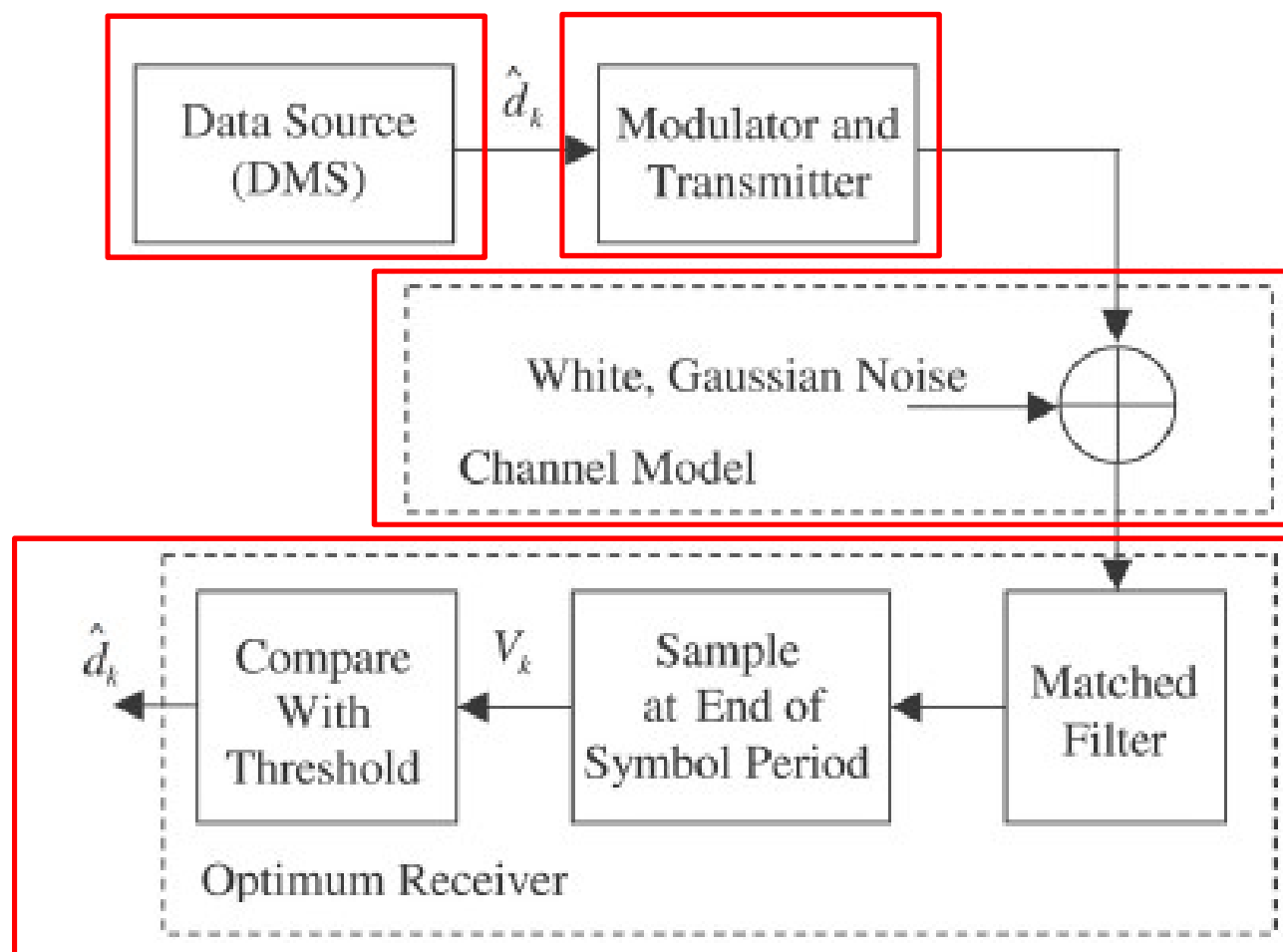


第一章 仿真的作用

- 1.1 复杂性示例
- 1.2 仿真的多学科特点
- 1.3 模型
- 1.4 确定性与随机性仿真
- 1.5 仿真的作用
- 1.6 仿真软件
- 1.7 选择Matlab

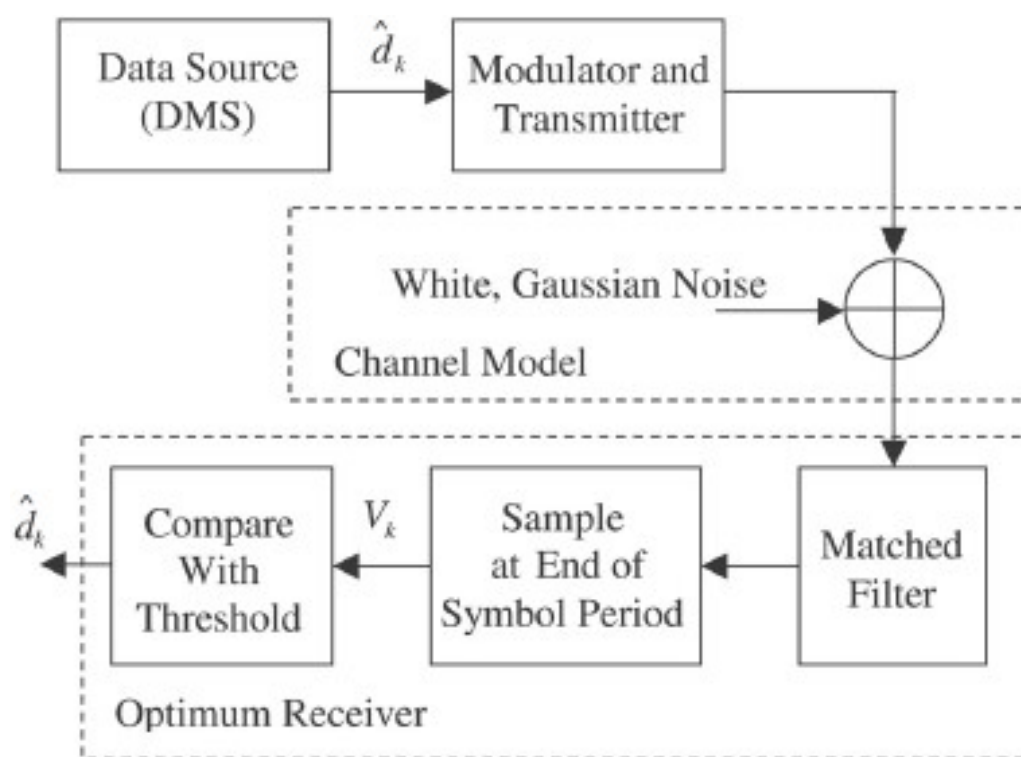
1.1 复杂性示例

- 易于解析处理的系统



1.1 复杂性示例

- 易于解析处理的系统



我们称这个系统为**易于解析处理的**，利用基本通信原理的知识，能很容易的对系统进行分析。

$$P_E = Q\left(\sqrt{k \frac{E_s}{N_0}}\right)$$

E_s : 表示一个符号周期内的平均能量，跟波形集合相关联

N_0 : 为加性信道噪声的单边功率谱密度

k : 取决于波形的相关程度。例如FSK, $k=1$; PSK相差为 π 弧度，使得信号为逆相关， $k=2$



1.1 复杂性示例

- 易于解析处理的系统

原因：

- 信道是AWGN信道，接收机是线性的
- 匹配滤波器是线性系统
- 判决统计量是高斯随机变量
- 假设数据源为无记忆
- 假设理想的符号同步，保证了正确获取判决统计量

通过分析，可以计算出作为接收端滤波器参数函数的误比特率，并可求出对应最小BER的滤波器参数值。所以系统的性能，通过传统的分析方法很容易确定。



1.1 复杂性示例

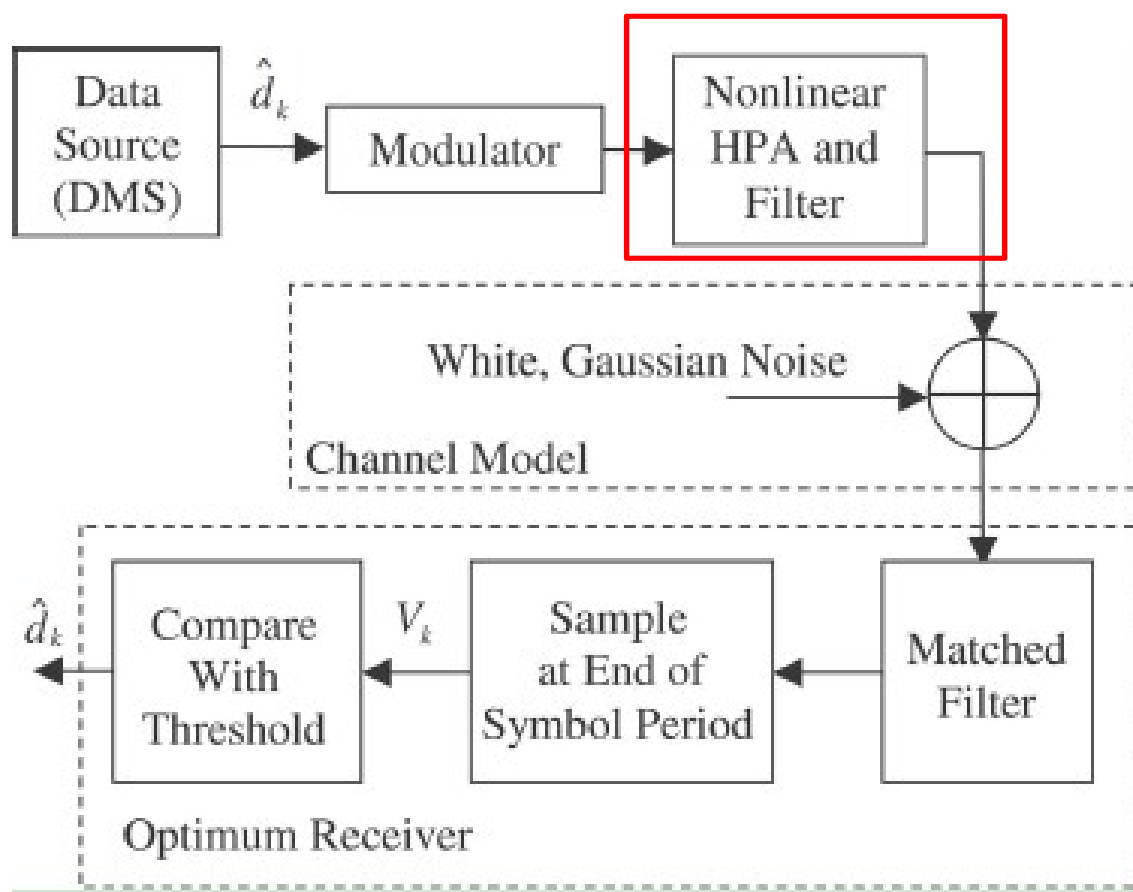
易于解析处理的系统是否也有作用呢？

意义：

- 可以作为更复杂的系统的**基本模块**；
- 开发出的仿真于直接的分析结果进行比较，能容易得到验证。
- 根据待研究系统的建模需求，对数据源、调制器、信道或接收机进行修改。
- 可以将其他子系统添加到仿真模型中，确保系统建立仿真模型的**起点**是正确无误的。

1.1 复杂性示例

- 需繁琐解析处理的系统



非线性功率放大器

频谱拓展

带通滤波器

信号时间扩散

符号间干扰 (ISI)

1.1 复杂性示例

- 需繁琐解析处理的系统

由于ISI，要进行判决的第*i*个符号的差错概率跟它前面的一个或多个符号有关。在解调第*i*个符号时必须考虑前续符号的数目，取决于滤波器输出信号的记忆。如果第*i*个符号的差错概率取决于前*k*个符号，计算

$$\Pr\{E_i | d_{i-1} d_{i-2} \cdots d_{i-k}\}$$

对二进制的情况，有 2^k 个长度为*k*的不同序列。假定每个数据符号等概率地取0或者1，第*i*个符号的差错概率为

$$P_E = \frac{1}{2^k} \sum_{d_{i-1}=0}^1 \sum_{d_{i-2}=0}^1 \cdots \sum_{d_{i-k}=0}^1 \Pr\{E_i | d_{i-1} d_{i-2} \cdots d_{i-k}\}$$

也就是所，必须计算 2^k 个不同的差错概率，并对这个结果做平均。因为假设信道是AWGN信道，这个差错概率都是高斯Q函数，可以直接计算每一个Q函数的值，但是计算过程很繁琐，因而经常会用到仿真的方法。



1.1 复杂性示例

- 需繁琐解析处理的系统

噪声注入点到统计量采集点之间的系统是线性的，即统计量为 V_k 为

$$V_k = S_k + I_k + N_k$$

其中， S_k 和 I_k 分别对应于信号和符号间干扰的 V_k 分量
 N_k 为对应于信道噪声的 V_k 分量。

- 如果信道噪声是高斯随机变量，则 N_k 也是高斯随机变量，因为它是高斯随机变量的线性变化。
- V_k 也是高斯随机变量，其方差和 N_k 的相同，但具有均值 $S_k + I_k$ ，这两个量均为确定性变量，所以 V_k 的均值可以直接计算。由信道噪声的功率谱密度和从信道到 V_k 出现处的系数等效噪声宽带可以确定 V_k 的方差，因而，可以得知 V_k 的概率密度函数并很容易求出差错概率。
- 总而言之，尽管系统中有非线性，但由于噪声并没有通过非线性部分， V_k 的概率密度函数还是很容易确定的



1.1 复杂性示例

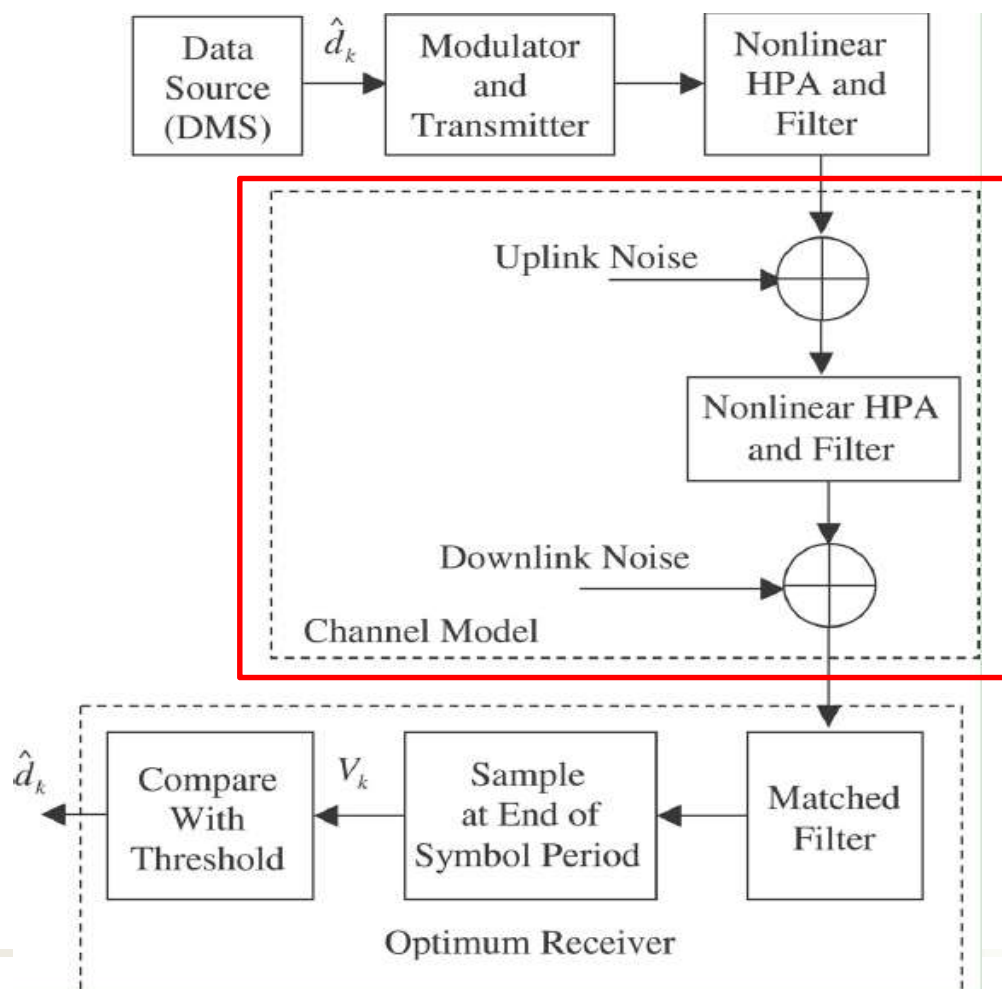
- 需繁琐解析处理的系统

噪声仅通过系统的线性部分这一事实对仿真方法论有很大的影响，因为噪声没有通过非线性部分， V_k 的均值可以使用无噪声仿真来迅速确定，同时可以解析确定 V_k 的方差。因此， V_k 的概率密度函数已知，并可很容易确定差错概率。

这些概念能汇总为一个简单而快速的仿真方法：半解析方法。在这种方法中，解析和仿真以一种快速仿真的方式结合在一起。

1.1 复杂性示例

- 难以解析处理的系统



将卫星转发器建模为
非线性功率放大器

增加系统的复杂度使得仿真变得十分必要。

目前许多系统都属于无法解析求解的这一类情况。比如，高干扰和多径环境下运行的无线蜂窝链路往往要通过仿真来详细分析。

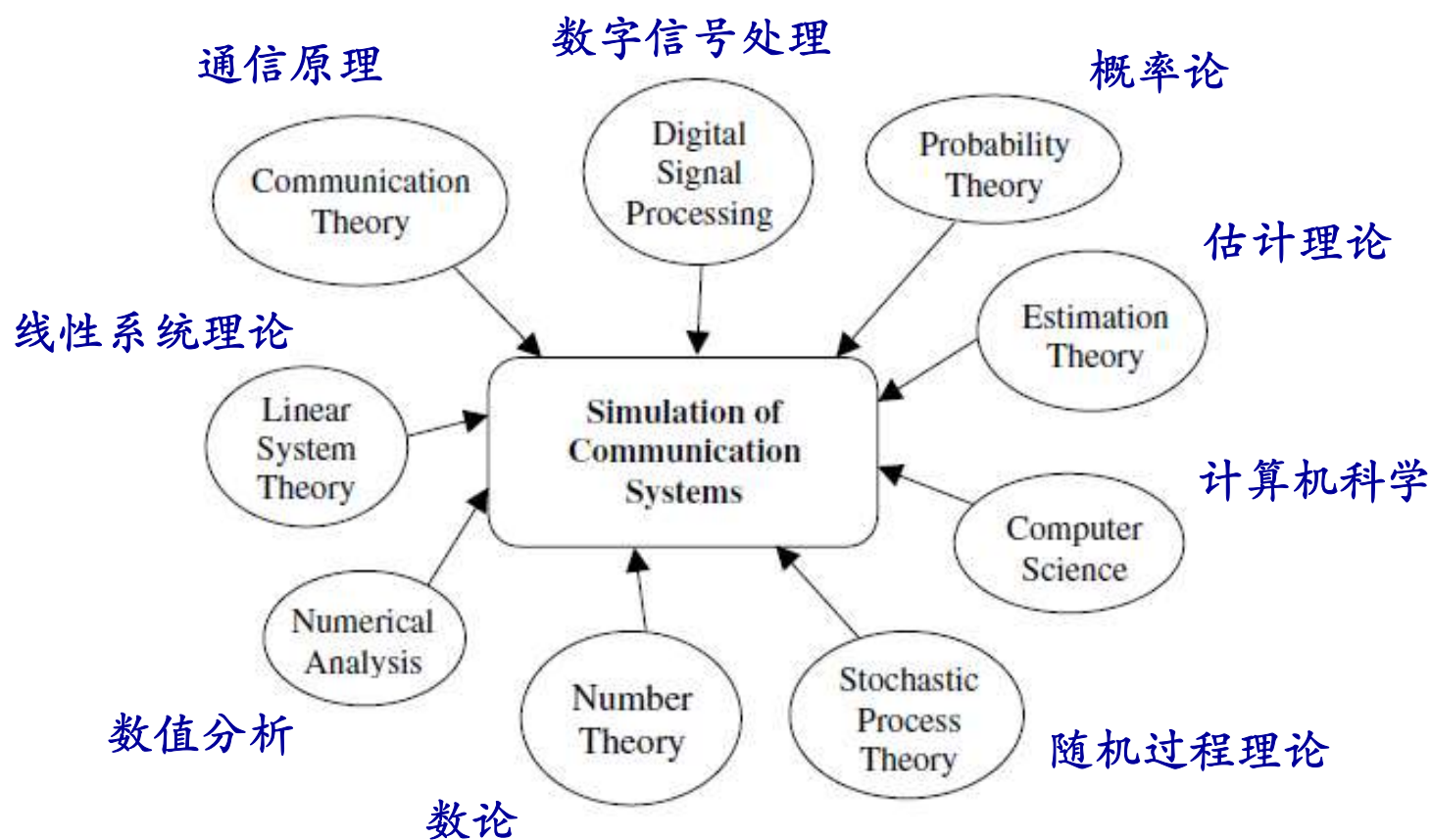
两(单)跳卫星通信系统模型

测试题03 下列哪些系统是必须要使用仿真方法的？

- ☐ A 易于解析处理的系统
- ☐ B 可以半解析处理的系统
- ☒ C 难以解析处理的繁琐系统

1.2 仿真的多学科特点

影响研究通信系统仿真的9个重要学科领域



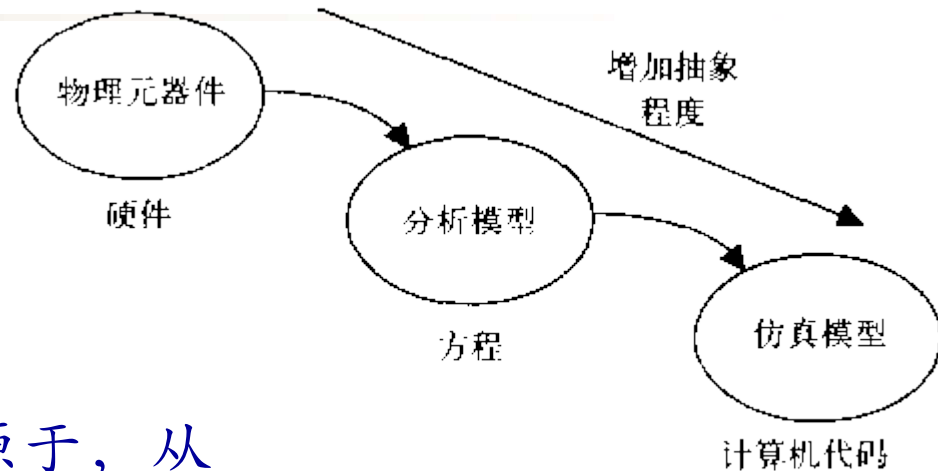


1.3 模型

- 通信系统仿真开发的第一步就是对研究的系统建立仿真模型
- 要点：模型要足够详细，以保持系统的基本特征；但又不能过于复杂，采用适度的计算资源开销。
- 精度，复杂度和计算量要求等方面做折中折衷。
- 解析模型和仿真模型二者都是物理元器件或系统的抽象。可以是单个元器件，也可以是子系统模块，也可以是一个完整的通信系统。
- 建模的首要和最重要的步骤是确定模型要描述的物理元器件的特征和工作特性。

1.3 模型

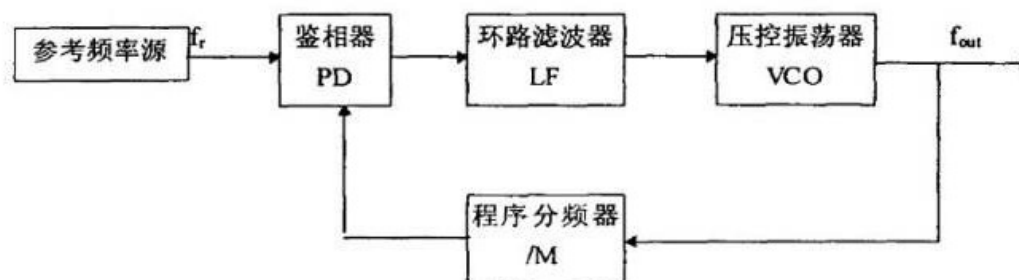
- 从物理元器件到解析模型，最后再到仿真模型，抽象程度依次增加。
- 抽象程度的增加部分来源于，从物理元器件到解析模型这一过程中所作的假设和近似，每做一次假设和近似，我们就离物理元器件的真实特性远一步
- 解析模型以方程或方程组的形式给出，这些方程仅对设备的某些方面建立了模型，对要建模设备进行了部分描述。
- 仿真模型通常是一组算法，这种算法用数值方法来求解定义解析模型的方程。



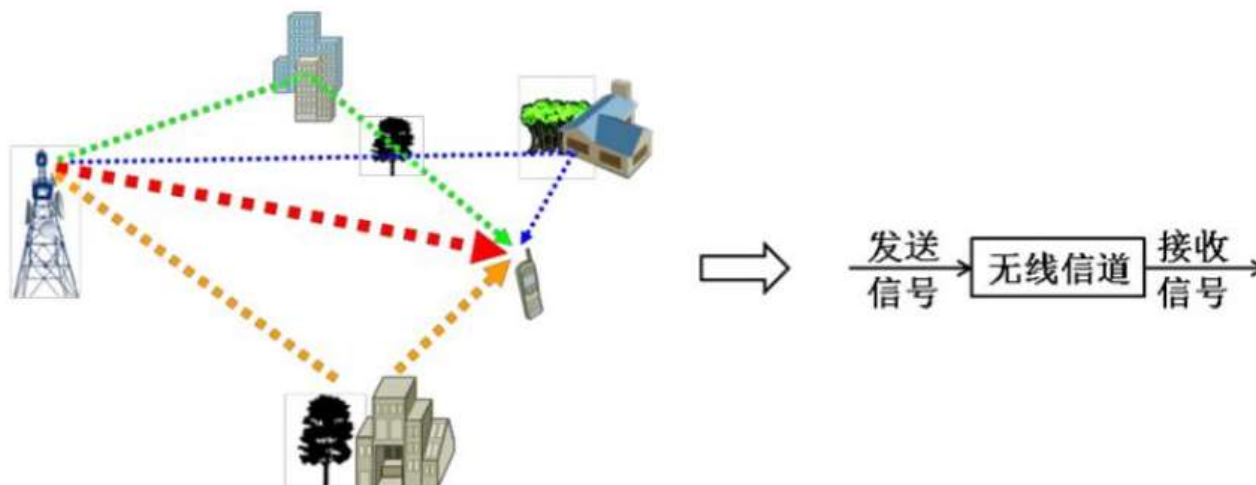
1.3 模型

例：锁相环模型

- 锁相环的解析模型可以有多种不同抽象程度的形式。可以由一组方程构成。
- 其中每一个方程对应一个功能运算（鉴相器，环路滤波器和压控振荡器）。通过观察这些方程，就可以明白在建立解析模型的过程中所作的假设。
- 表示不同信号处理运算的各个方程，又可以组成一个非线性微分方程（可能是时变的），来描述锁相环的输入和输出关系，这又是一个抽象得多的模型。在这个模型中，我们无法在分别识别出锁相环中的各信号处理运算及其对应的波形。



1.3 模型



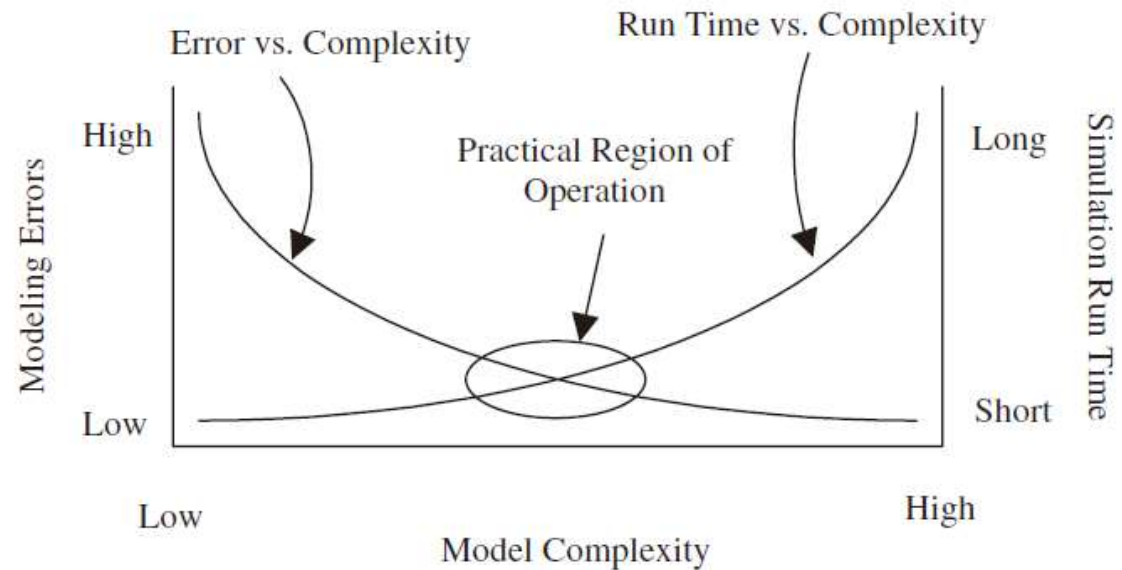
具有不同抽象程度的模型。

例：信道模型

- 采用波形级的方法来建模。在这种方法中，模型对波形采样值进行处理；
- 基于符号而非波形采样值的离散马尔科夫过程，而且，马尔科夫过程通常还将调制器、发射机和接收机纳入信道中，这些模型都是高度抽象的，也难以用精确的参数表示。但是一旦找到这种模型，就可以给出数值上**高效**的**仿真**。

1.3 模型

系统模型的复杂度
仿真精度
运行速度
工程实践
试验成本
.....



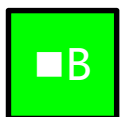
例如：仿真采样频率。提高采样率可以有效减少混叠误差，但是却增加了仿真运行时间。

系统建模考虑的不是物理元器件，而是可以做各种工程解析的模型。解析模型是物理元器件的抽象，涉及到很多假设和近似。仿真模型基于解析模型，涉及到更进一步的假设和近似。在此过程中，必须高度谨慎，以保证模型的有效性，并保证仿真结果能反映实际情况。

测试题04 : 下列哪些说法正确?



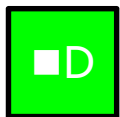
从物理元器件到解析模型，最后再到仿真模型，抽象程度依次增加。



抽象程度的增加部分来源于，从物理元器件到解析模型这一过程中所作的假设和近似



系统建模考虑的不是物理元器件，而是可以做各种工程解析的模型。



解析模型可以有多种不同抽象程度的形式。

提交



1.4 确定性与随机性仿真

- 确定性仿真，分析和确定性仿真都会给出一个数值结果，这一点是很重要的。每次作分析，都会得到同一个数。而每次进行确定性仿真，也会获得相同的结果。节省时间和避免在进行冗长计算时可能出现的计算错误。
- 随机性仿真，仿真的输入是随机变量的一个样本。仿真的结果将不再是一个确定性的波形，而对波形进行采样会得到一组随机变量。

1.4 确定性与随机性仿真

例：调制格式BPSK，信道是AWGN，计算接收机输出端的误符号率：

$$P_E = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \quad Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} \exp\left(-\frac{y^2}{2}\right) dy$$

E_b 为符号能量， N_0 为单边噪声功率谱密度， $Q(x)$ 为高斯Q函数

P_E 是一个数而不是随机变量，尽管接收机输入端出现了随机量（噪声）。 P_E 是无数次试验的平均。

这里的试验包括让数字信号通过系统和观察结果。结果当然是在接收机输出端要么作出正确判决，要么观察到一个差错。



1.4 确定性与随机性仿真

遍历性 (Ergodic) 过程，可以用两种不同的方法确定差错概率

- 观察单个的发送位，并通过总体均值 (Ensemble Average) 来计算 P_E 。包含无穷多个噪声波形的总体，每个波形都具有相同统计特性。
- 发送无穷多个二进制符号，使用单个噪声样本函数，再作时间平均来确定 P_E 。

遍历性：随机过程的各个样本函数都同样经历了随机过程的各种可能状态，因此从随机过程的任何一个样本函数就能得到随机过程的全部统计信息，任何一个样本函数的特性都能充分地代表整个随机过程的特性。

1.4 确定性与随机性仿真

- Goal: BER 蒙特卡罗方法

输入大量符号

计算输出的错误符号

差错概率的蒙特卡罗估计

$$\hat{P}_E = \frac{N_e}{N}$$

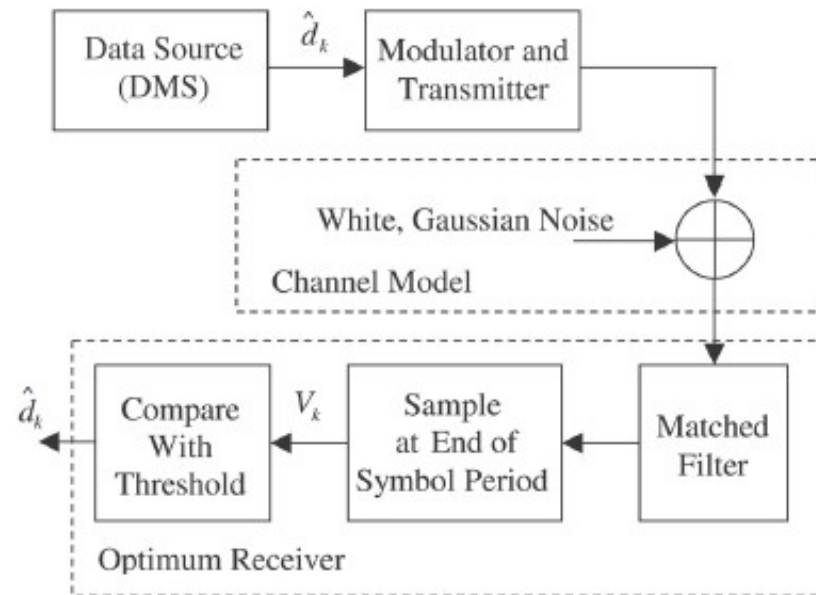
使用概率的相对概率定义，符号

差错概率可以定义为

$$P_E = \lim_{N \rightarrow \infty} \frac{N_e}{N}$$

- 误比特率 (*bit error rate*)
- 比特差错概率 (*probability of bit error*)

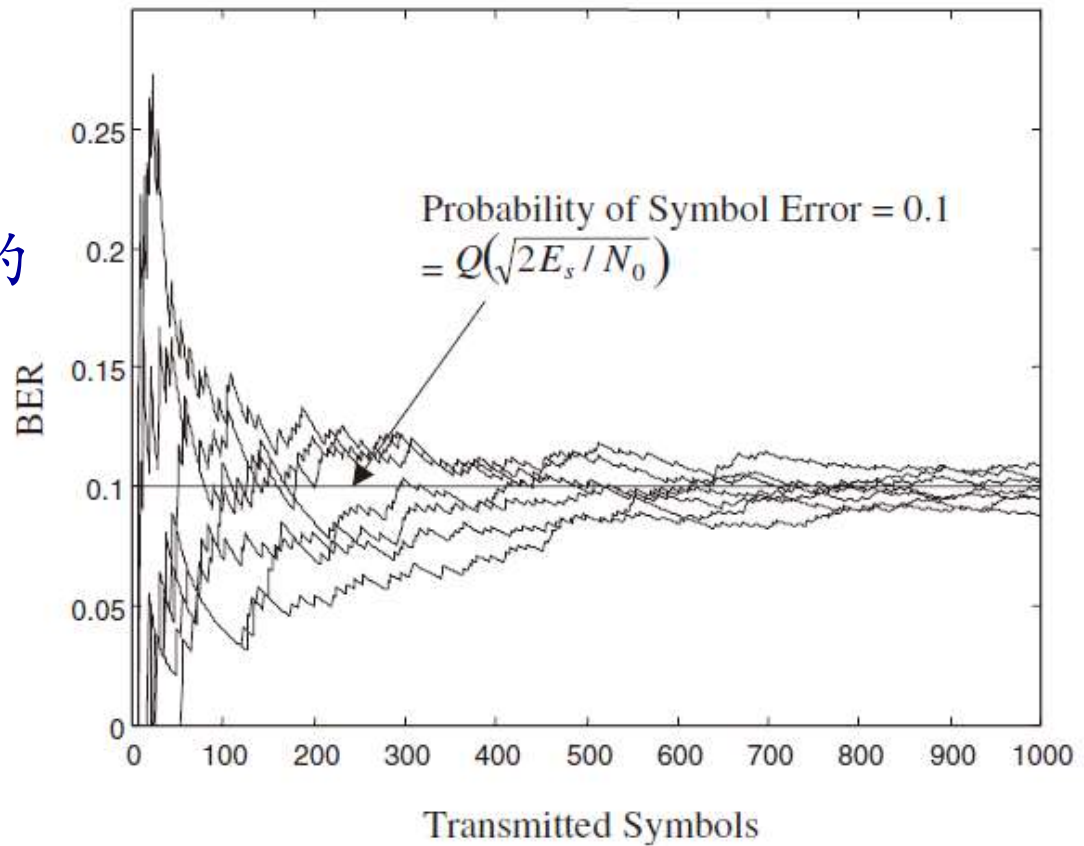
误比特率是比特差错概率的一致性估计



1.4 确定性与随机性仿真

误码率是一种比率，它表示每N个符号出Ne个差错的意思；在带噪声的信道上发送N个符号的随机试验重复多次，会得到不同的结果，是一个随机变量。

误码概率基于无穷多个符号通过系统这个假设，因此误码概率是一个数而不是随机变量。只要 E_s 和 N_0 保持恒定，这个就是固定值。



测试题05：误码概率是一个

- ☐ A 随机过程
- ☐ B 随机变量
- ☒ C 固定值
- ☐ D 时间函数

■ 提交



1.5 仿真的作用

- 通信系统设计的各个阶段
- 主要用在性能评估和设计权衡研究（参数优化）
- 测试过程和基准的设定、生命终结预测、现场系统布置后出现异常状况的调查。
- 复杂通信系统的的设计是以“自顶至底”的方式完成的，但硬件实现通常是“自低至顶”进行。



1.5 仿真的作用

- 复杂通信系统的的设计是以“自顶至底”的方式完成的，但硬件实现通常是“自低至顶”进行。
- 在设计系统时，从系统级（最高抽象级）开始，填入系统级设计的细节，再往下到子系统级，最终下到原件级，然后我们就到了最底层，可以从这里获取原件组装的详情。
- 在构建系统时，最先制造出元件来，然后把元件组装成子系统，最后再用子系统构成整个系统。
- 仿真开发也采用“自顶至底”的方法，从具有最高抽象程度的系统级仿真开始，到越来越详细的子系统及元件的模型和仿真。



系统设计过程的不同阶段

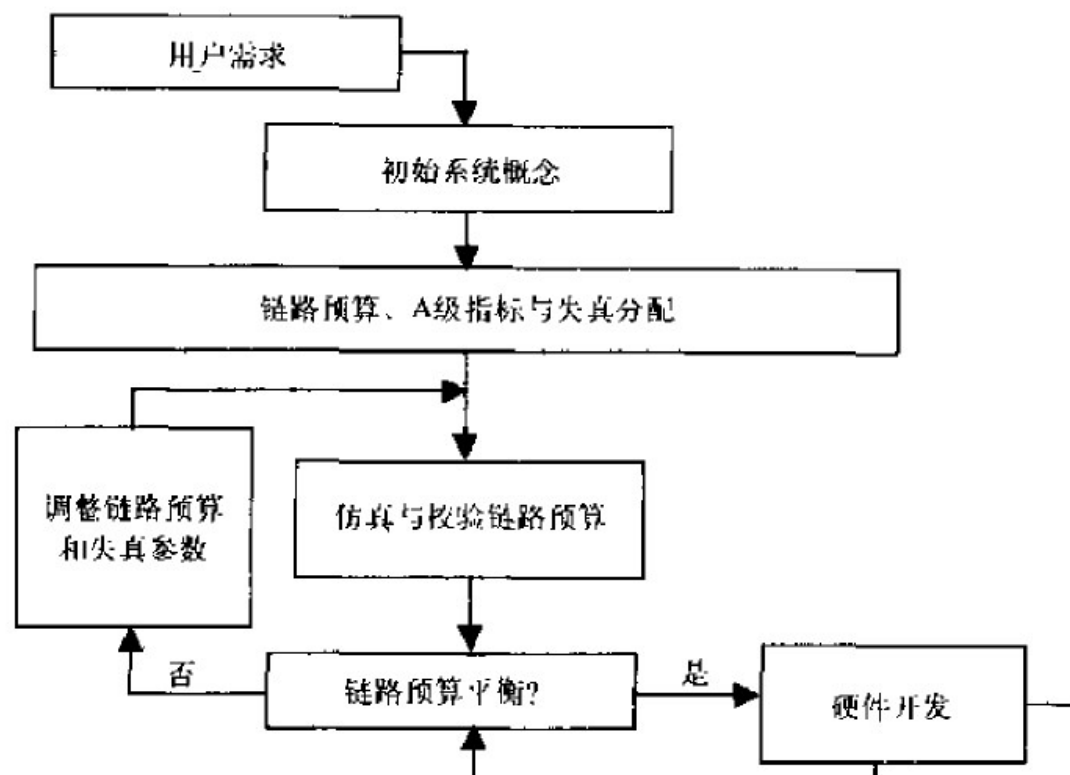
- 链路预算与系统级标校过程
- 关键元件的实现与测试
- 完成硬件原型与验证仿真模型
- 生命终结预测

1.5.1 链路预算与系统级标校过程

确定设计需求：吞吐率、差错率、中断概率以及对带宽、功率、重量、复杂度/成本、系统预期工作的信道和系统生命周期等的约束。（系统工程师）；

形成基本概念：调制方式、编码与均衡技术（如果有必要）

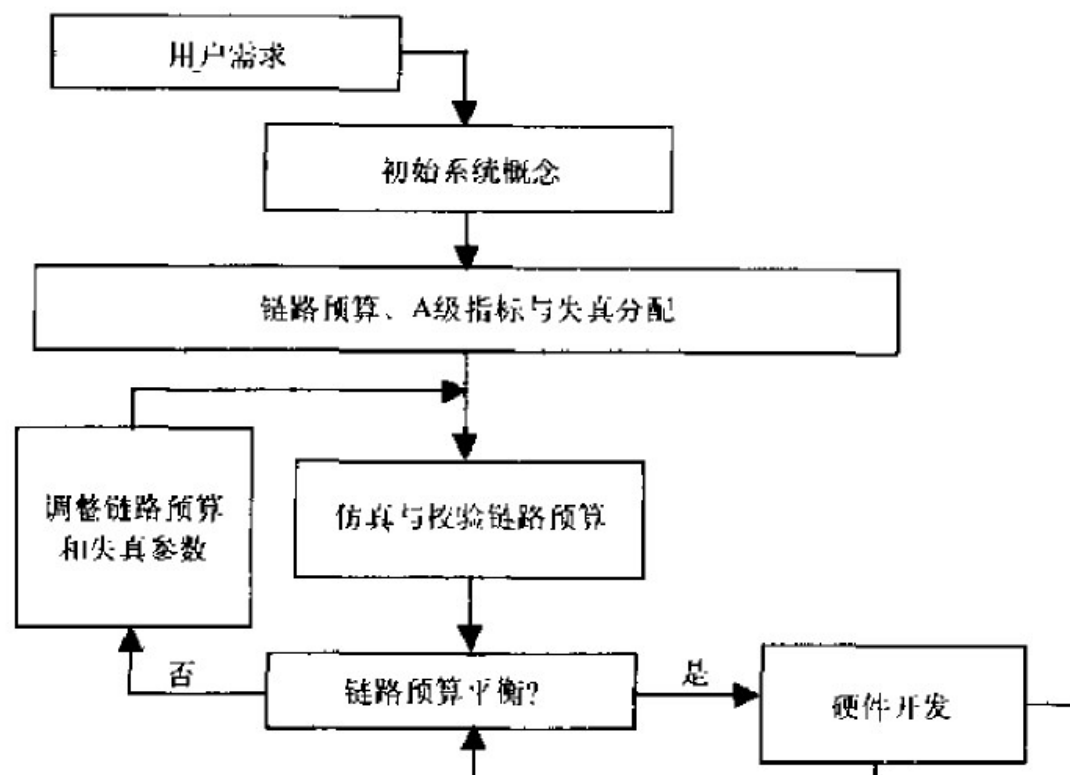
确定A级指标：A级指标的参数值，比如功率级、带宽和调制指数。



1.5.1 链路预算与系统级标校过程

该阶段设计的目标是确定系统拓扑结构和参数值，以便同时满足性能目标 and 设计约束。

系统性能是信噪比（SNR或等价地 E_b/N_0 ）和通信链路中所有元件引入的总失真的函数。信噪比可以通过链路预算来确定。

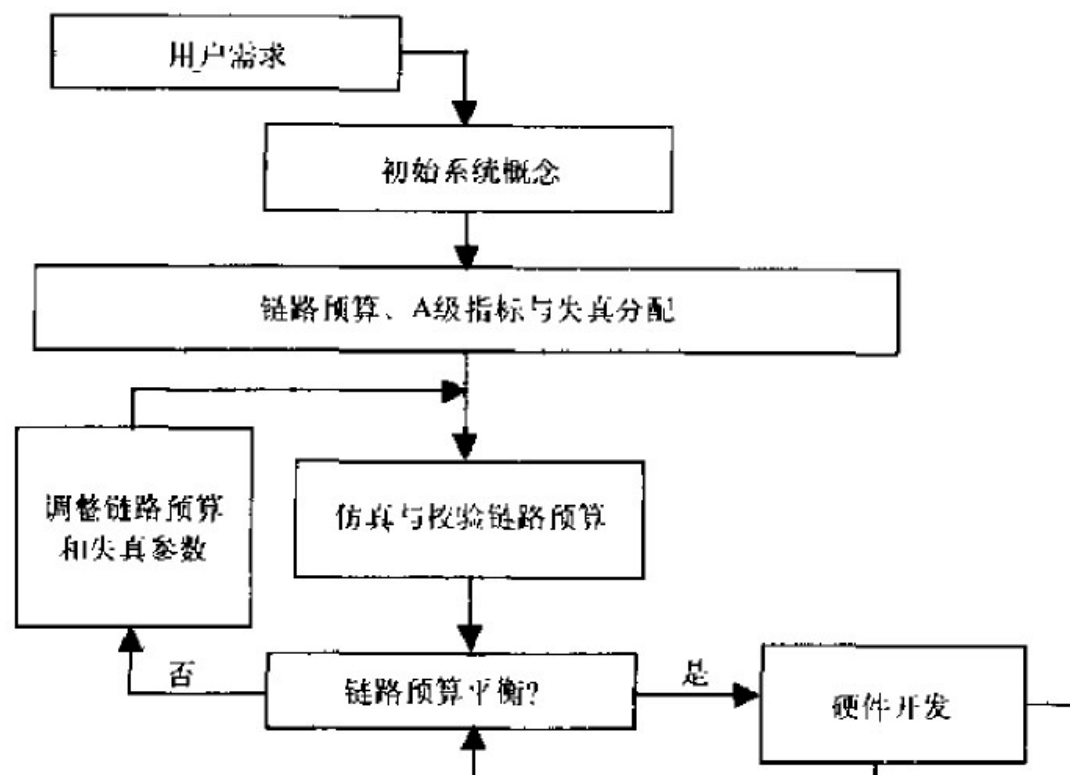


链路预算主要是功率计算，要考虑发送功率、天线增益、路径损耗、功率增益以及放大器和滤波器的噪声系数。可以确定SNR或 E_b/N_0 范围范围

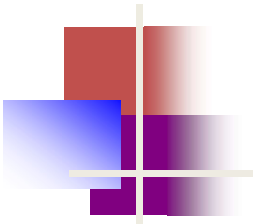
1.5.1 链路预算与系统级标校过程

因为无法制造理想元件，放大器和滤波器等元器件的实际实现会产生非理想特性，引起信号失真，影响系统性能。

实现损耗表示 E_b/N_0 必须能加的大小，以便克服非理想元器件引起的信号失真所造成的影响。



有时实现损耗也叫通信或失真参数，比如滤波器带宽，可能影响系统中多处噪声的功率，这又反过来会影响链路预算。



1.5.1 链路预算与系统级标校过程

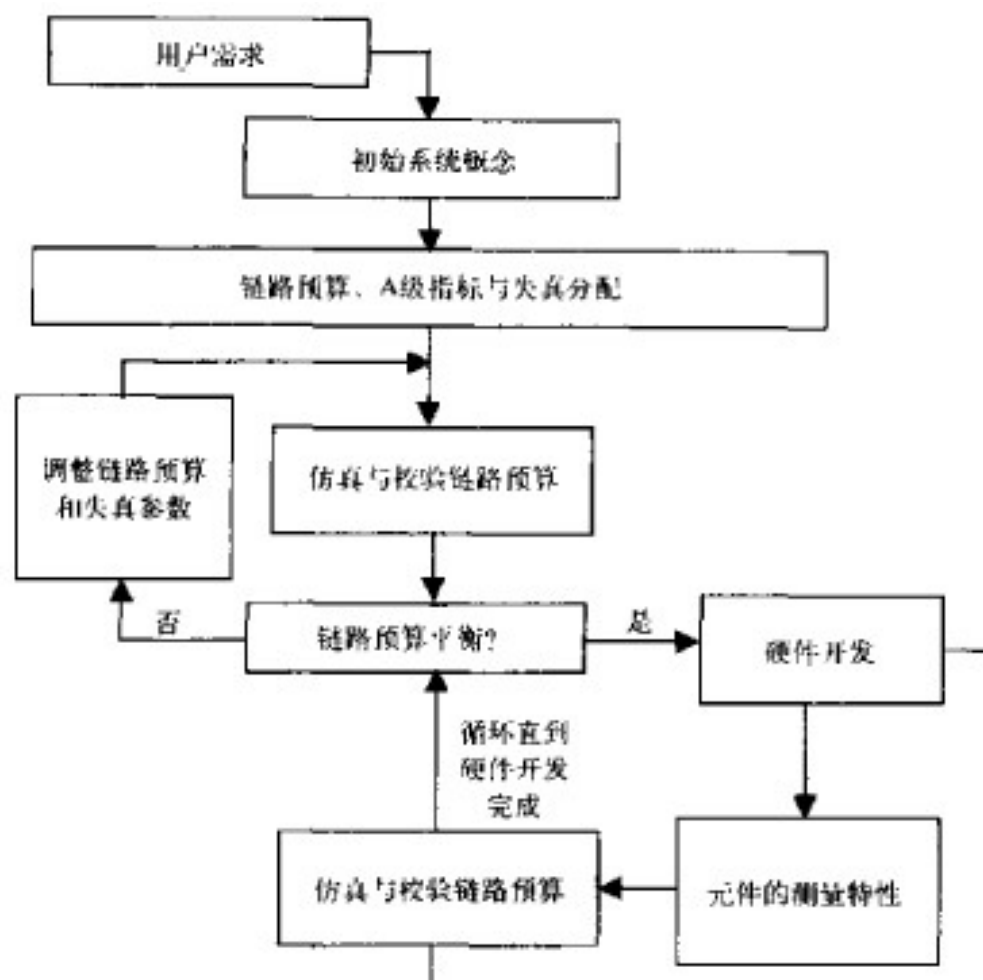
链路预算封闭的、平衡的？

系统设计者配置完系统的初始参数，A级指标等，将考虑所有实现损耗后系统接收机输入端处的净 E_b/N_0 。如果链路具有足够的 E_b/N_0 ，满足可接受的系统性能并具有一定的安全余量，则称此链路预算为“封闭的”或“平衡的”。

链路预算不封闭或不平衡，就是修改A级指标，实现损耗甚至系统结构，并重作链路预算。通过详细的仿真，可以精确估计性能指标和验证非理想实现造成的性能降级，如果链路预算通过仿真验证还是封闭的，就可以进入设计过程的下一阶段

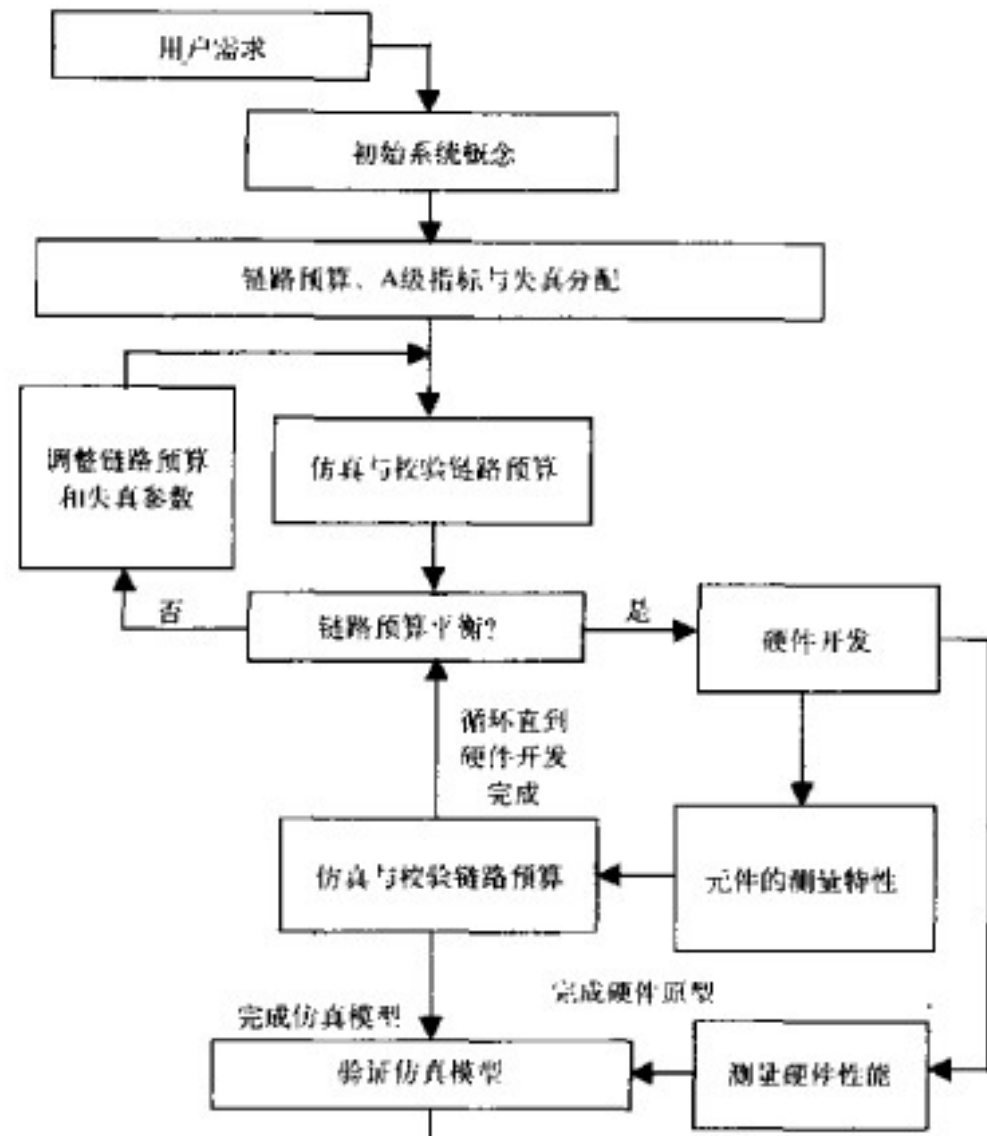
1.5.2 关键元件的实现与测试

- 新信号处理算法和新硬件，仅造出了几个关键元件，要测试整个系统的硬件是不可能的
- 仿真中将待测的元件测量特性代入该元件的仿真模型
- 通过系统仿真评估关键元件是否满足设计要求



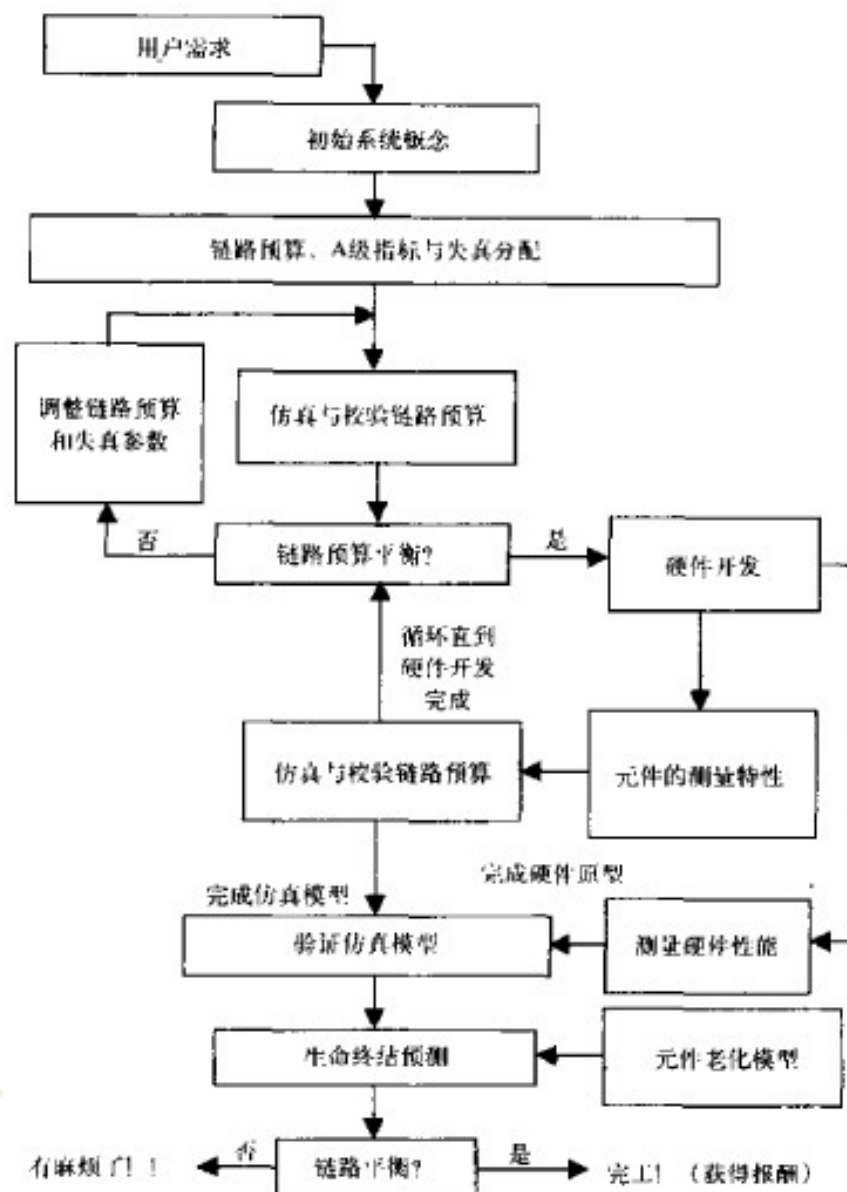
1.5.3完成硬件原型与验证仿真模型

- 整个系统的硬件原型以及与之对应的仿真模型
- 仿真模型包括仿真中大部分元件的测量特征
- 硬件原型上测出整个系统的许多性能指标，还要进行并行仿真
- 获得一个经验证的仿真模型，预测生命终结。



1.5.4 生命周期测试

- 通信卫星、海底电缆
- 生命周期测试（Beginning of Life, BOL）将元件老化模型代入验证过的BOL模型，就得到系统EOL性能指标





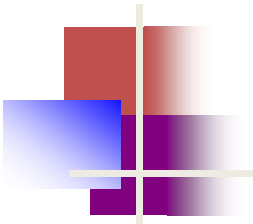
1.6 仿真软件

- 时间驱动（单速度、多速率或可变速率采样）：该增量等于采样频率的倒数，每个模型能根据新仿真时钟的值来更新模型状态“do循环”“for循环”。
- 事件驱动：事件驱动的仿真可把时钟往前拨任意时间长度，到为下一个关心的事件所安排的时间，系统中每个功能模块根据新仿真时间的值更新模型状态。需要内插和重采样，并带来一些跟时间安排相关的额外开销。
- 混合驱动。对通信系统仿真而言，最常用的还是带单速率或多速率采样的时间驱动仿真。如果信号带宽具有很宽的可变范围，这种系统的仿真就要用多速率采样。比如在扩频系统中使用多速率采样就可以大大降低所需仿真运行时间。



1.6 仿真软件

- MATLAB（物理层仿真、信号处理）
- OPNET（协议仿真和业务仿真）
- C,C++(物理层模块仿真)
- NS 2, NS 3（协议仿真）



无论何时都不要认为仿真可以代替传统的分析或硬件测量。在与分析和测量二者联合使用时，仿真的功能才最为强大。很常见的是，通过反复仿真获得洞察，可以据此识别系统的关键参数和简化系统模型，而得到的简化又可用做更多分析。

解决系统级问题总会要求一定程度的分析。举例来说，为了保证系统工作正常和仿真结果合理，人们必须理解性能参数（如误比特率、解调器输出端的均方差或接收机输入端的信噪比等）与系统参数（如发射功率与带宽、调制格式或码率）之间的基本依存关系。换言之，当仿真中参数变化时，必须保证这些变化的观察结果合理，并且不与已知理论相抵触。这些“完整性检查”在验证仿真时很重要，也几乎总是要求一定程度的分析工作。



1.7 选择Matlab

- 70年代中期，Cleve Moler博士及其同事在美国国家基金会的帮助下，开发了LINPACK和EISPACK的FORTRAN语言子程序库，这两个程序库代表了当时矩阵运算的最高水平。
- 到了70年代后期，身为美国新墨西哥州大学计算机系系主任的Cleve Moler，在给学生上线性代数课时，为了让学生能使用这两个子程序库，同时又不用在编程上花费过多的时间，开始着手用FORTRAN语言为学生编写使用LINPACK和EISPACK的接口程序，他将这个程序取名为MATLAB，其名称是由MATrix和LABoratory（矩阵实验室）两个单词的前三个字母所合成。
- 在1978年，Malab就面世了。这个程序获得了很大的成功，受到了学生的广泛欢迎。在以后的几年里，Matlab在多所大学里作为教学辅助软件使用，并作为面向大众的免费软件广为流传



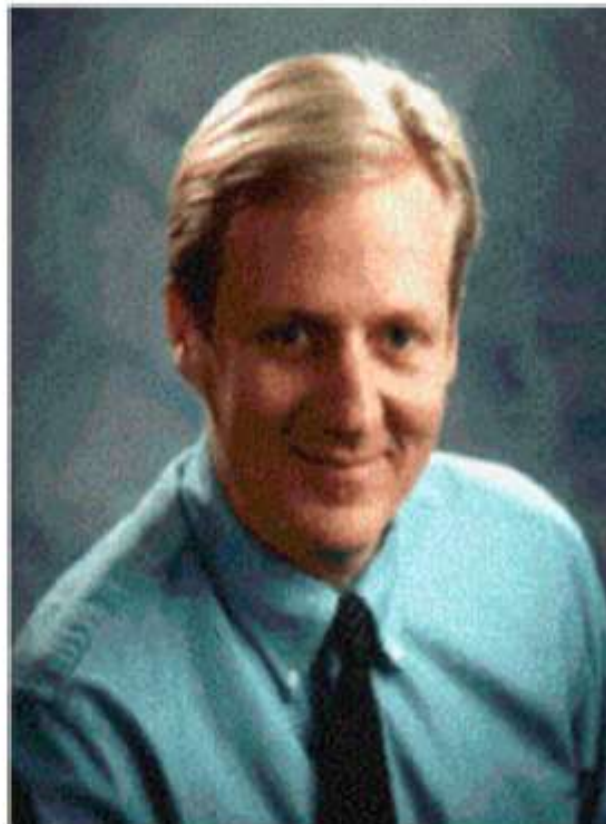
1.7 选择Matlab

- 将MATLAB商品化的不是Cleve Moler,而是一个名叫Jack Little的人。
- 用C语言重新编写了MATLAB的核心。
- 在Moler的协助下,于1984年成立MathWorks公司,首次推出MATLAB商用版。
- 在其商用版推出的初期, MATLAB就以其优秀的品质(高效的数据计算能力和开放的体系结构)占据了大部分数学计算软件的市场,原来应用于控制领域里的一些封闭式数学计算软件包(如英国的UMIST、瑞典的LUND和SIMNON、德国的KEDDC)就纷纷被淘汰或在MATLAB上重建。

1.7 选择Matlab



Cleve Moler



Jack Little



1.7 选择Matlab

- 1992年，支持Windows 3.x的MATLAB 4.0版本推出，增加了Simulink, Control, Neural Network, Signal Processing等专用工具箱。
- 1993年，推出了Matlab 4.1，主要增加了符号运算功能。当升级至Matlab 4.2c，这一功能在用户中得到广泛应用。
- 1997年，Matlab 5.0版本问世了,实现了真正的32位运算，加快数值计算，图形表现有效。
- 2001年，MathWorks公司推出了Matlab 6.0 (R12) 。
- 2002年，推出了Matlab 6.5(R13)，Simulink升级到了5.0，推出程序加速器，计算速度有了明显的提高。
- 2005年，推出了MAILAB 7.1(Release14 SP3)，在这一版本中Simulink升级到了6.3，软件性能有了新的提高，用户界面更加友好。。
- 2009年，推出MATLAB 2009R (7.8.0.447)
- 目前，Matlab软件支持多种系统平台，如常见的WindowsNT/XP、UNIX、Linux等。



1.7 选择Matlab

- MATLAB就是这样经过了近30年的专门打造、20多年的千锤百炼，它以高性能的数组运算（包括矩阵运算）为基础，不仅实现了大多数数学算法的高效运行函数和数据可视化，而且提供了非常高效的计算机高级编程语言，在用户可参与的情况下，各种专业领域的工具箱不断开发和完善，MATLAB取得了巨大的成功，已广泛应用于科学研究、工程应用，用于数值计算分析、系统建模与仿真。
- 早在20世纪90年代初，欧美等发达国家的大学就将MATLAB列为一种必须掌握的编程语言。近几年来，国内的很多大学也将MATLAB列为了本科生必修课程。
- MATLAB软件功能之强大、应用之广泛，已成为21世纪最为重要的科学计算语言。可见学习掌握这一工具的重要性。



1.7 选择Matlab

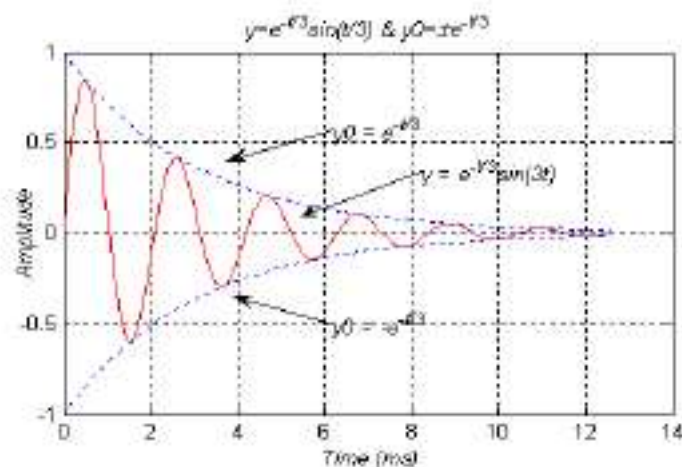
- 语法规则简单
- 基本的语言环境提供了数以千计的计算函数
- 是一种脚本式 (scripted) 的解释型语言
- 平台无关性 (可移植性) 其绘图功能也是平台无关的。

因此, MATLAB是一个简单易用、功能强大的高效编程语言。

1.7 选择Matlab

功能强大

- 数值运算优势
- 符号运算优势 (Maple)
- 强大的2D、3D数据可视化功能
- 许多具有算法自适应能力的功能函数





1.7 选择Matlab

语言简单、内涵丰富

- 语言及其书写形式非常接近于常规数学书写形式；
- 其操作和功能函数指令就是常用的计算机和数学书上的一些简单英文单词表达的，如：help、clear等；
- 完备的帮助系统，易学易用。



1.7 选择Matlab

扩充能力、可开发能力较强

- MATLAB完全成了一个开放的系统
- 用户可以开发自己的工具箱
- 可以方便地与Fortran、C等语言接口

编程易、效率高

- Matlab以数组为基本计算单元
- 具有大量的算法优化的功能函数



1.7 选择Matlab

matlab对变量本身是矩阵操作。matlab的编程思想，与C语言有一个很大的不同点在于循环并不是那么常用。

例：计算一下成绩中大于90的个数。

使用类似于C语言的风格，可以写为：

```
scores = [76,45,98,97];  
count = 0;  
k = 0;  
while k < length(scores)  
    k = k+1;  
    if scores(k)>90  
        count = count+1;  
    end  
end  
disp(count)
```



1.7 选择Matlab

matlab对变量本身是矩阵操作。matlab的编程思想，与C语言有一个很大的不同点在于循环并不是那么常用。

例：计算一下成绩中大于90的个数。
使用matlab的观点，可以写为：

```
count = length(find(scores>90))
```

find是一个很好的函数，可以免去很多时候的遍历向量，查找其中的一个是否符合条件的程序。

也可以认为这种写法类似于对C++中STL库的调用。



1.7 选择Matlab

switch-case语句的一般表达形式为：

switch 〈选择判断量〉

Case 选择判断值1

选择判断语句1

case 选择判断值2

选择判断语句2

.....

otherwise

判断执行语句

end



1.7 选择Matlab

与其他的程序设计语言的switch-case语句不同的是，在MATLAB语言中，当其中一个case语句后的条件为真时，switch-case语句不对其后的case语句进行判断，也就是说在MATLAB语言中，即使有多条case判断语句为真，也只执行所遇到的第一条为真的语句。这样就不必像C语言那样，在每条case语句后加上break语句以防止继续执行后面为真的case条件语句。



1.7 选择Matlab

C语言中，高维数组是通过1维数组递归定义的，1维数组的每个元素又是一个数组，这虽然满足了控制了语言规模的优点，但是在使用时对于新手而言还是会频繁出错的。而matlab的方法就很奇特（以3维数组为例，因为matlab将2维矩阵定义为基本的数据类型）：你先写好多个2维矩阵，然后把它们“串”起来，变成一个3维矩阵，举一个例子：

```
A = [1 2 3;4 5 6;7 8 9];
```

```
B = [11 22 33;44 55 66;77 88 99];
```

```
C = [111 222 333;444 555 666;777 888 999];
```

```
my_3D_arr(:,:,1) = A;
```

```
my_3D_arr(:,:,2) = B;
```

```
my_3D_arr(:,:,3) = C;
```

然后通过下标访问：`my_3D_arr(1,1,2)`