三 、 SIMULINK的仿真步骤：

1. 打开SIMULINK 寻找模块，并连接为系统框图：

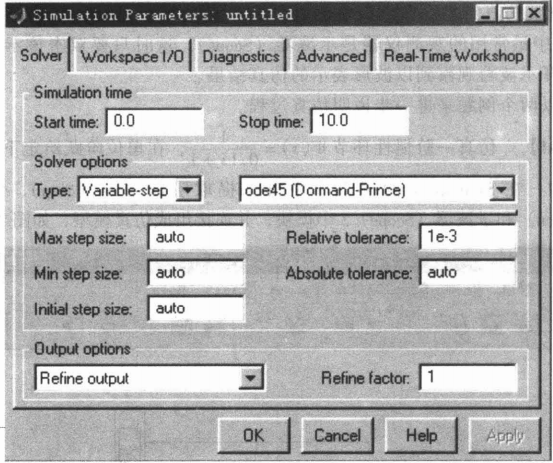
至少有一个源模块，一个输出模块；

1. 设置模块参数；
2. 设置仿真参数：

确定仿真的步长，时间，和选取仿真的算法等。

点击simulation ，下拉子菜单中，点击simulation parameter 弹出对话框；

最常用为 解算器 Solver ；



1. 仿真时间 ( 开始时间 start time 和 终止时间 stop time )：

一般连续系统中，从零开始，并预设一个终止时间；

时间越长，计算时间也越长；

② 算法选择 :

1. 计算类型( Tape )：

可变步长 variable step 和 固定步长 fixed step；

1. 仿真误差( tolerance )：

相对误差 relative tolerance 和 绝对误差 absolute tolerance ；

系统默认相对误差为 1/1000。

选择合适的计算误差，对仿真的速度和能否收敛的影响很大，

当仿真不能够收敛的时候，适当放宽误差，有时候可以取得效果。

绝对误差一般可取为自动 auto ；

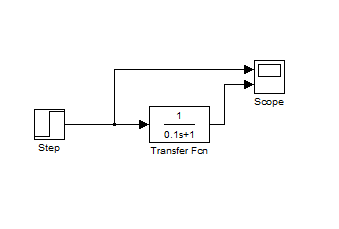
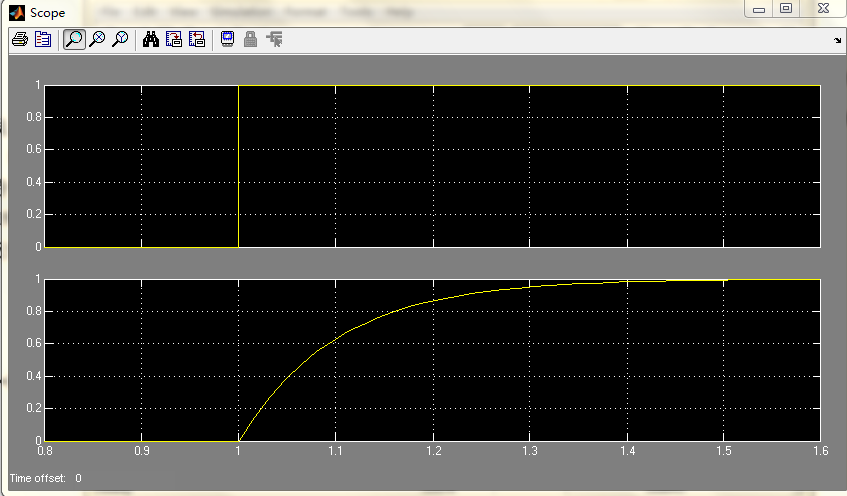
1. 启动仿真：

Start 或者 Ctrl + T；

1. 观察仿真结果：

常用的仪器是示波器： scope ；

例1 ： 仿真一阶惯环节W(s) = 1/( 0.1s＋１)，在单位阶跃给定下的响应。

1. 找到各个模块，并连接成仿真模型：
2. 模块赋值；
3. 设置仿真参数，如开始时间0.8s，终止时间1.6s。
4. 可以看到结果为0.8s – 1.6s；

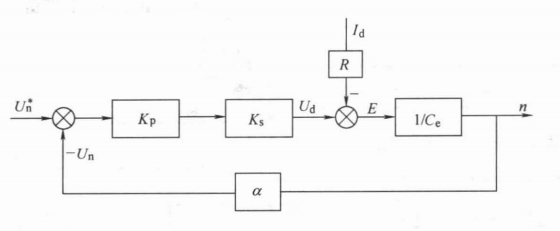
例2、 转速反馈有静差直流调速系统的稳态结构如图，仿真该系统的稳态特 性。已知额定参数：

220v 55A 1000r/min，

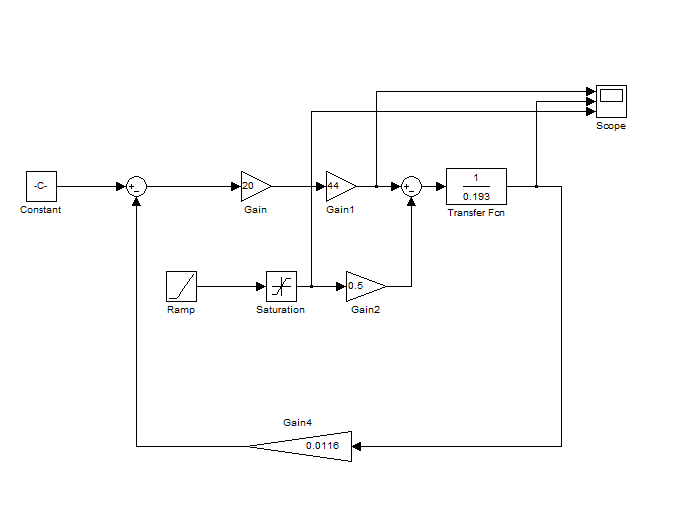
电动势常数 Ce = 0.193V\*min/r

晶闸管放大倍数Ks = 44，

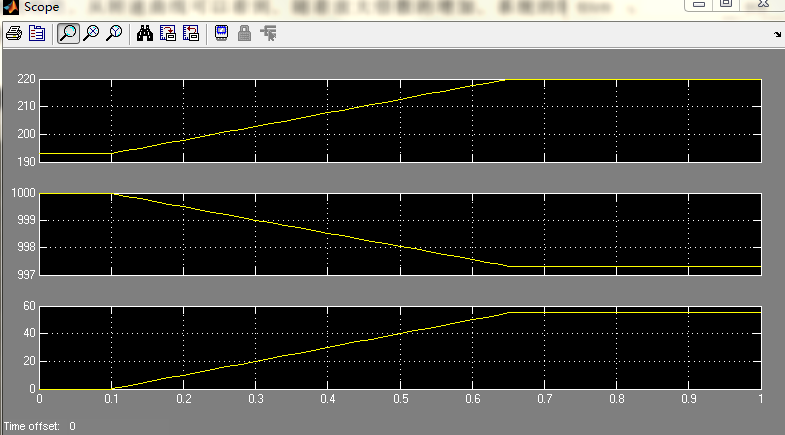
电枢回路总电阻 R = 1Ω

 转速反馈系数 α = 0.0116;

电枢电流的加载采用斜坡函数( Ramp )：起始时间0.1s，上升斜率100；



放大器放大倍数 Kp = 20；

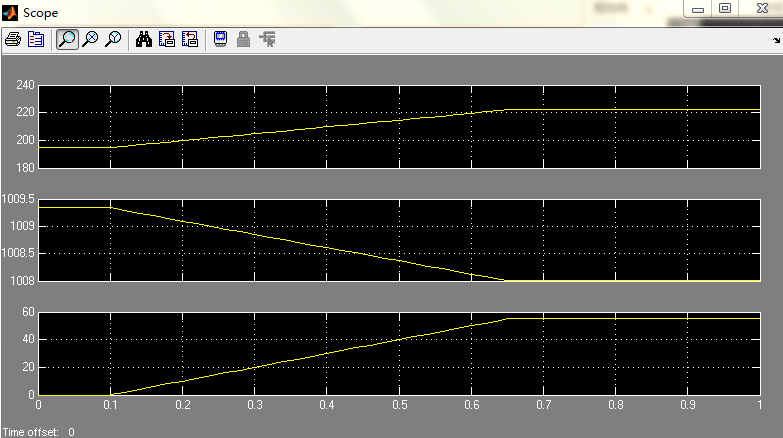


分析可知，空载时，空载转速为1000r/min；

当负载电流增加时，转速略有下降，在0.64s时，电流达到额定，此时的

转速约为997.4r/min， 系统的速降为△n = 2.6 r/min。

放大器放大倍数 Kp = 40；



可以看到，随着放大器的倍数增加，转速降减少，硬度增加，抗负载扰动增加。

2. 1.5 SIMULINK 的仿真算法：

在 SIMULINK 的仿真过程中选择合适的算法是很重要的，仿真算法是求常微  
 分方程、传递函数、状态方程解的数值计算方法，这些方法主要有欧拉法  
 ( Euler) 、阿达姆斯法 (Adams) 、龙格·库塔法 (Rung-Kutta) ，这些算法都主要建  
 立在泰勒级数的基础上。欧拉法是最早出现的一种数值计算方法，它是数值计  
 算的基础，它用矩形面积来近似积分计算，欧拉法比较简单，但精度不高，现  
 在已经较少使用。阿达姆斯法是欧拉法的改进，它用梯形面积近似积分计算，  
 所以也称梯形法，梯形法计算每步都需要经过多次迭代，计算盘较大，采用预  
 报.校正后只要迭代一次，计算盘减少，但是计算时要用其他算法计算开始的几  
 步。龙格，库塔法是间接使用泰勒级数展开式的方法，它在积分区间内多预报几  
 个点的斜率，然后进行加权平均，用作计算下一点的依据，从而构造了精度更  
 高的数值积分计算方法。如果取两个点的斜率就是二阶龙格·库塔法，取四个点  
 的斜率就是四阶龙格.库塔法。

Simulink 方法分为两大类，可变步长和固定步长算法；

1. ode45 (Dormand-Prince)  
 基于显式 Rung-Kutla (4 , 5) 和 Dormand-Prince 组合的算法，它是一种一步解  
 法，即只要知道前一时间点的解 r *(t.* \_I ) ，就可以立即计算当前时间点的方程  
 解 r (t.) 。对大多数仿真模型来说，首先使用 od e45 来解算模型是最佳的选择，  
 所以在 SIMULINK 的算法选择中将 ode45 设为默认的算法。

2. ode23 (Bogacki-Shampine)  
 基于显式 Rung-Kutta (2 , 3) 、 Bogacki 和 Shampine 相结合的算法，它也是一种  
 一步算法。在容许误差和计算略带刚性的问题方面，该算法比 ode45 要好。

3. ode113 ( Adams)  
 这是可变阶数的 Adams-Bashforth-Moulton PECE 算法，在误差要求很严时，  
 odel13 算法较 ode45 更适合。 odel13 是一种多步算法，也就是需要知道前几个时  
 间点的值，才能计算出当前时间点的值。

4. ode15s (Stiff/NDF)  
 一种可变阶数的 Numerical Differentiation Formulas (NDFs) 算法，它相对  
 Backward Differentiation Formulas 算法(简称 BDFs 算法，也称Gear 算法)较好。它  
 是一种多步算法，当遇到带刚性( Stiff) 问题时或者使用 ode45 算法不行时，可  
 以试试这种算法。

5. ode23s (Stiff/Mod. Rosenbrock)  
 这是一种改进的二阶 Rosenbrock 算法。在容许误差较大时， ode23s 比 ode15s  
 有效，所以在解算一类带刚性的问题时用 ode15s 处理不行的话，可以用 ode23s  
 算法。

6. ode23t (Mod. Stiff/Trapezoidal)  
 一种采用自由内插方法的梯形算法。如果模型有一定刚性，又要求解没有  
 数值衰减时，可以使用这种算法。

7. ode23tb (stiff/TR- BDF2)  
 采用 TR-BDF2算法，即在龙格.库塔法的第一阶段用梯形法，第二阶段用二  
 阶的 Backward Differentiation Formulas 算法。从结构上讲，两个阶段的估计都使用  
 同一矩阵。在容差比较大时， ode23tb 和 ode23t 都比 ode15s 要好。

8. discrete (No Continuous States)  
 这是处理离散系统(非连续系统)的算法。

2. 1.5.2 固定步长类算法  
 固定步长类算法，顾名思义，是在解算模型(方程)的过程中步长是固定不变的，在 SIMULINK 的算法中固定步长类算法有如下几种:

1. ode5 (Dormand-Prince)  
 采用 Do口nand-Prince算法，也就是固定步长的ode45 算法。

2. ode4 (Rung-Kutta)  
 四阶的龙格·库塔法。

3. ode3 (Bogacki-Shampine)  
 采用 Bogacki-Shampine 算法。

4. ode2 (Heun)  
 一种改进的欧拉算法。

5. odel (Euler) 欧拉算法

6. discrete 'No Continuous States)  
 不含积分的固定步长解法，它适用于没有连续状态仅有离散状态模型的计算。

电力电子电路和调速控制系统仿真中一般都是用可变步长算法；

2.1.6 示波器：

1. 示波器的参数：
2. 、 Number of axes ：

设置Y 轴输入的数量；

1. Time range：

用于设定示波器时间轴的最大值，这一般可以选自动(auto) 这 样 X 轴就自动以系统仿真参数设置中的起始和终止时间作为示 披器的时间显示范围。

③ Tick labels ：

选择标签的贴放位置。

④ Sampling：

选择数据的取样方式。

1. Decimation

当右边设为3时，每3个数据取一个，设为5时，每五个数据取一个；

设的数字越大，显示的波形就越粗糙，但是数据存储的空间可以减少。

一般该值保持为1；

二. Sample Time:

右边输入的是采样的时间间隔，这时将按采样间隔提取数 据显示。

⑤ Floating scope：

选中后示波器成为浮动的示波器，没有输入接口，但是可以接 收其他模块发送过来的数据。

第二页是数据页:

①. 数据点数：

预制值为5000，既可以显示5000个数据，若超过5000个数 据，则删掉前面的保留后面的。

也可以不选择该项，所有数据都显示，计算量大，对内存要求高。

②. Save data to workspace:

即将数据放到工作间去，则仿真的结果可以保存起来，并可以 用 MATLAB 的绘图命令来处理，也可以用其他绘图软件画出更 漂亮的图形。

Save data to workspace 下，还有两个设置：

1. Variable name ：

保存的数据命名，给数据起个名字，以便将来调用。

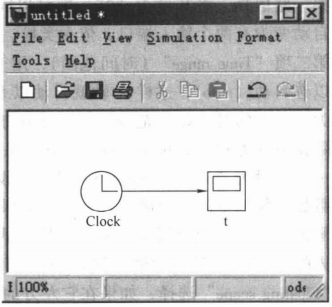
1. Format ：
2. Array：

适用于只有一个输入变量的情况；

二. Structure with time 和 Structure：

适用于以矢量表示的多个变量的情况；

前者同时保存数值和时间，后者仅保存数据。

用 Arrary 格式保存的变量，为了以后可以用 MATLAB 命令重画，同时需要将时间也保存起来，这时可以在模型平台上调用一个 Sources 模型库中的时钟模块 (Clock) ，并将其连接一个示波器，用示波器的 Save data toworkspace 功能将时间作为一个变量同时保存起来。

4浮动示波器 (Floating Scope )：

浮动示波器是示波器使用的一项特殊功能，它不需要将示波器与外部模 块图2-21 Y 轴范围设定用线连接，就可以选择示波器的显示信号，使 用是很方便的。

将一个示波器变为浮动示波器，只要在示波器参数页上选中"floating scope" 选项关闭参数对话框后，示波器图标的输入端口就没有了，这 时该普通示波器就改变为一个浮动示波器。也可以从 Sinks 模型库中 直接调用 floating scope 模块，效果是相同的。

在仿真模型图上放置一个浮动示波器模块后，双击模块图标出现示波器 窗口，在窗口的图形区域用右键单击，在弹出的功能项中选择" SignalSelector" 栏，则可以打开信号选择对话框(见图 2-22) ，对话 框右边列出了可供显示的信号名称，在信号名前的小方框(口)内打 "v" ，则可以在示波器上观察该信号。