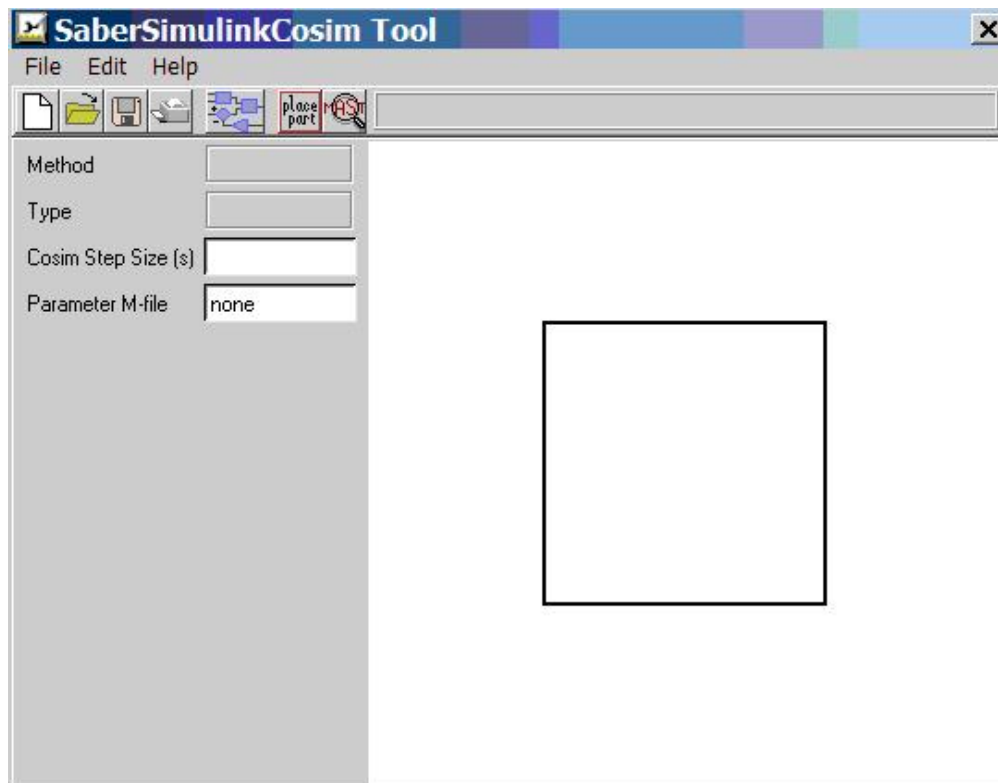


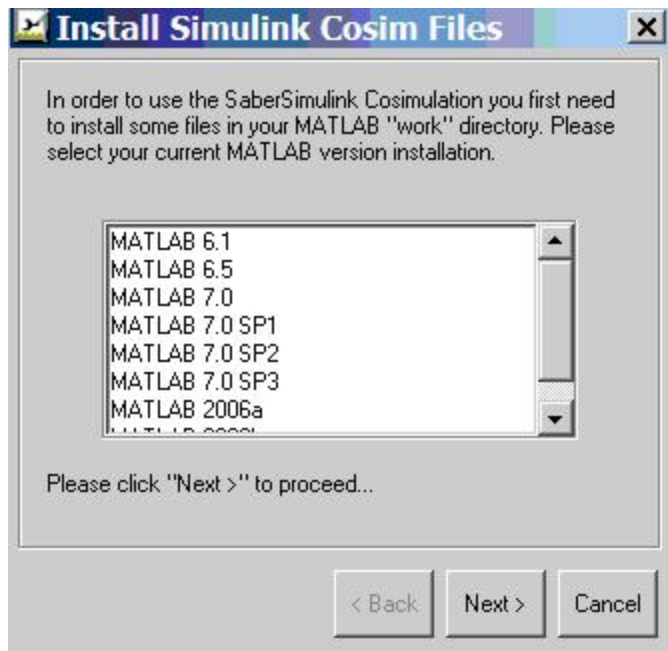
Saber 可以和 Simulink 实现协同仿真(Co-Sim),仿真时以 Saber 为主机,调用 Simulink,两个软件以固定时间步长交换数据.运用 Saber 和 Simulink 进行协同仿真的关键在于接口定义,需要在两个软件环境中分别进行适当的定义,才能实现 Co-Sim.下面以 Saber 软件自带的实例来介绍一下如何实现 Saber/Simulink 协同仿真.

首先确定是否已经安装了 Saber 和 Simulink 软件,本文 Saber 的版本是 2006.06, Simulink 的版本是 7.0.接下来需要在 Sketch 中安装与 Simulink 版本匹配的 Co-Sim 文件,具体过程如下:

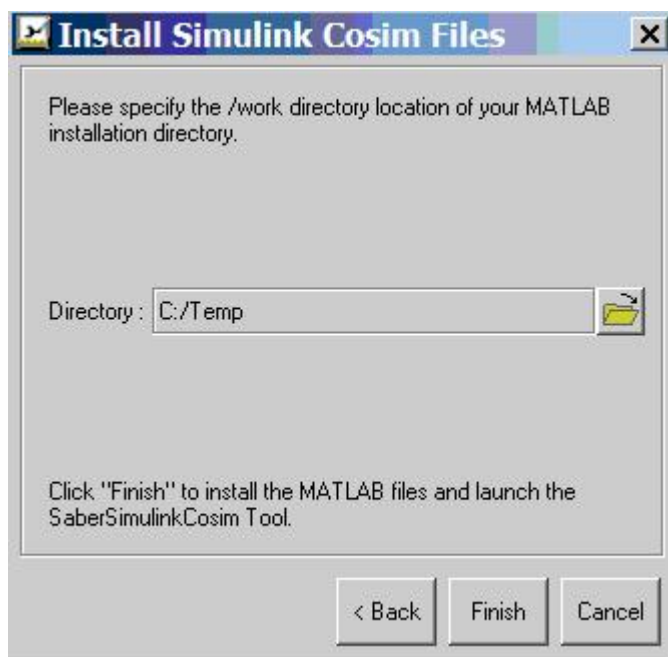
1 在 Sketch 环境中打开 Open the Saber/Simulink co-simulation tool 出现如下图所示的界面.



2 选择 File->Install Cosim Files... 命令出现如下图所示的对话框.



3 在该对话框中选择合适的 MATLAB 版本并单击 Next 按钮,出现如下图所示对话框:



4 将需要的文件安装在 MATLAB 安装目录下的 work 目录下,共有三个文件,具体如下:

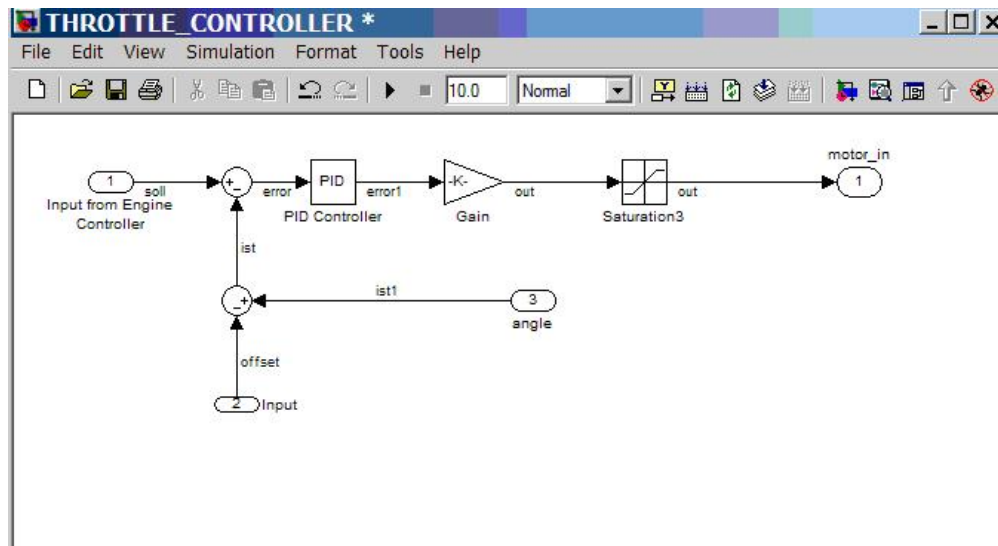
SaberSimulinkCosim.dll

SaberCosim.mdl

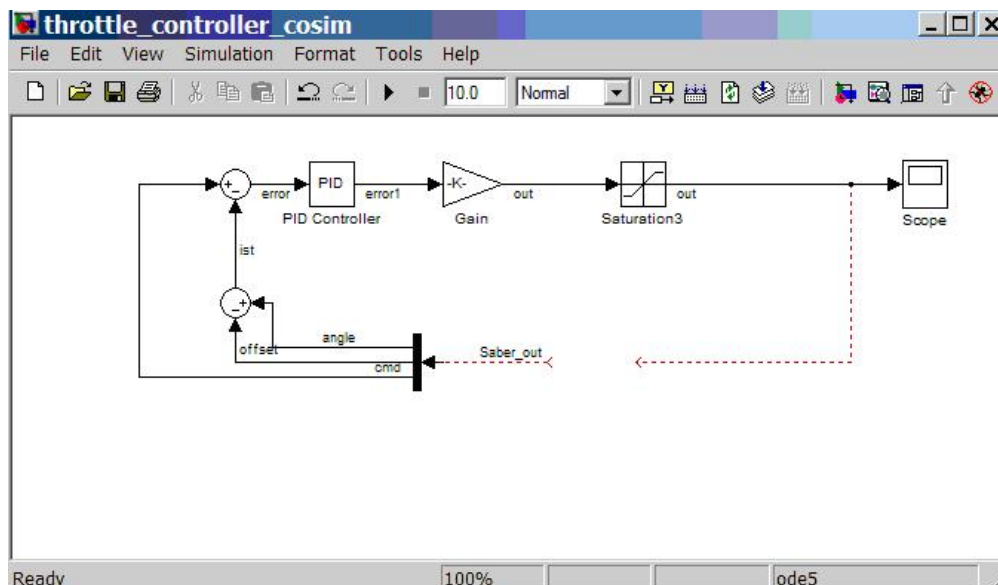
saber.jpg

接下来需要在 Simulink 中定义有 Saber 仿真的那部分模型的输入输出接口. 本文选择的是 Saber 软件自带的 Saber/Simulink RTW 实例, 位于 $\$SABER_HOME\backslashSaber\lib\tool_model\backslashSimulink2SaberRTWexport_Matlab7_1\backslashthrottle_controller$ 目录下, 对该实例进行适当的修改, 就可以用来实现 Saber/Simulink 的协同仿真. (这里需要注意 MATLAB 版本, 不同的版本有不同的目录, 都在 $\$SABER_HOME\backslashSaber\lib\tool_model\backslash$ 目录下), 具体过程如下:

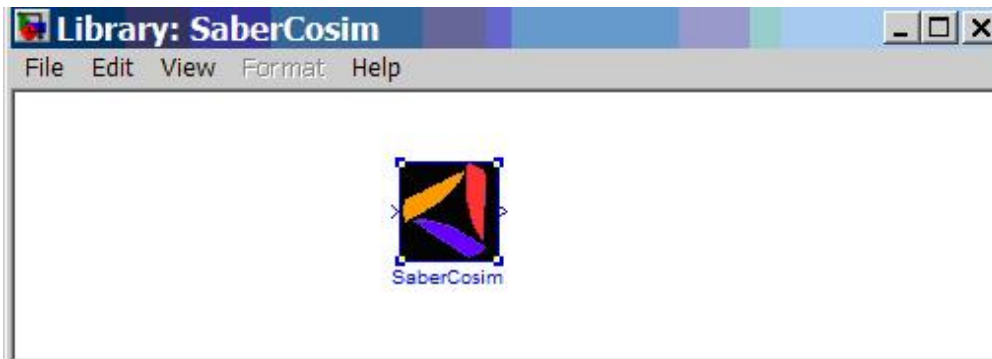
1. 启动 Simulink 并打开实例文件 THROTTLE_CONTROLLER.mdl, 打开以后的文件如下图所示:



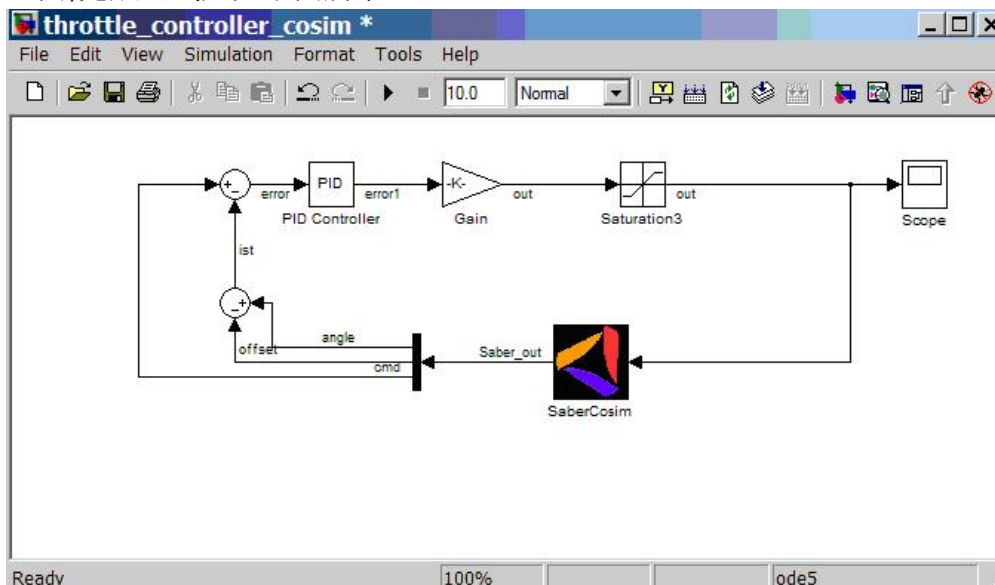
2. 对该图进行修改, 修改完毕后将原理图另存为 throttle_controller_cosim.mdl 如下图所示:



3. 在 Simulink 中打开 SaberCosim.mdl 文件, 文件位于 MATLAB 安装目录下的 work 目录, 如下图所示:



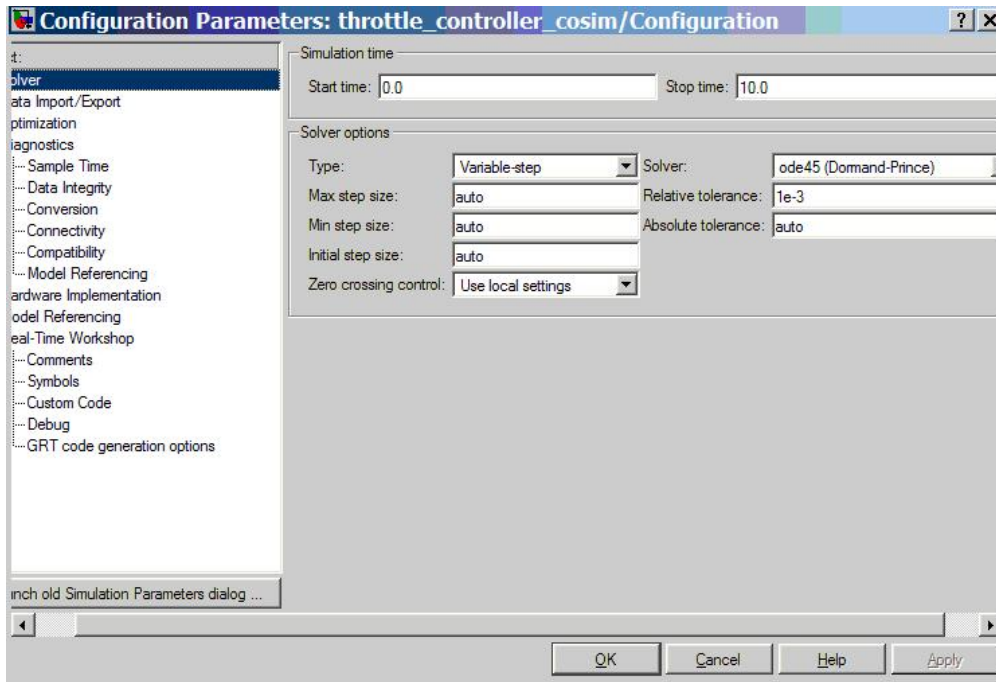
4.将 SaberCosim 图标放入刚才修改另存的 throttle_controller_cosim.mdl 原理图,完成连线后如下图所示:



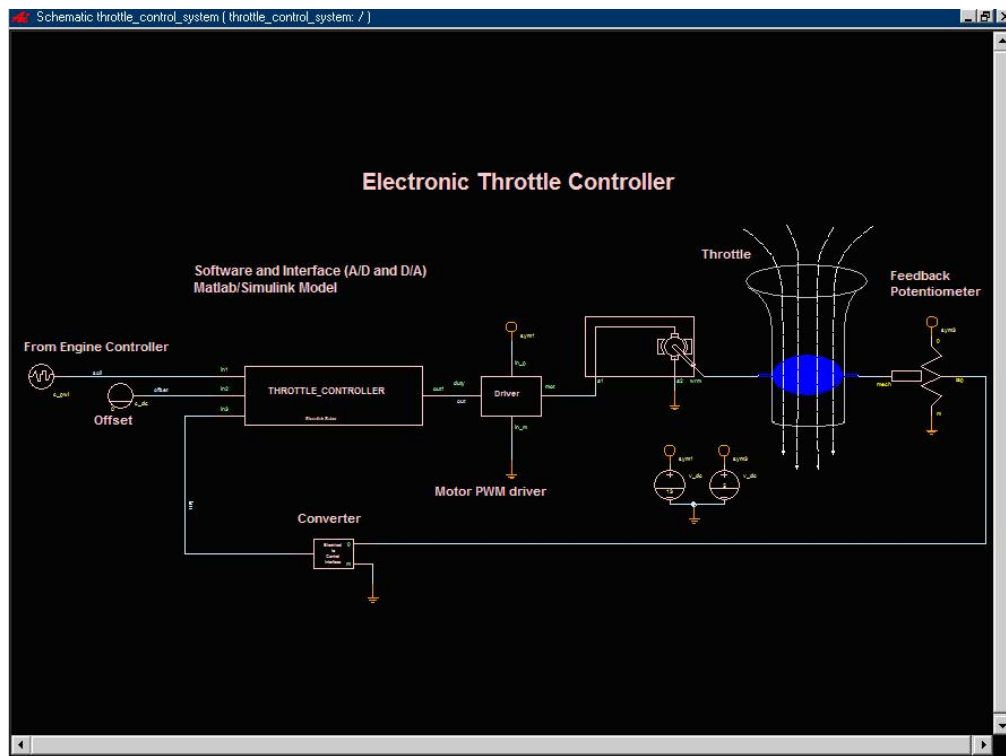
5.双击 SaberCosim 图标,设置输入输出端口数如下图所示:



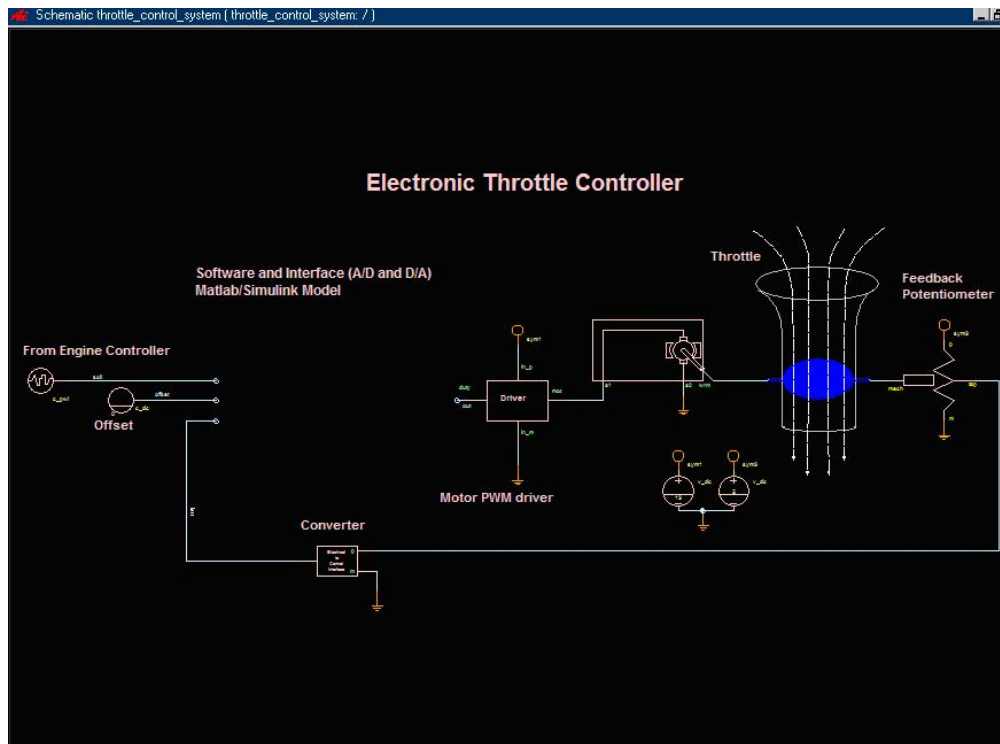
6.选择 Tools/Real Time/Option 命令,弹出对话框,在对话框中的左边选择 Solver,在右边 Solver Options 栏中设置 Type 为 Variable-step,如下图所示,之后保存文件并关闭 MATLAB 程序.



接下来需要在 Saber 中定义输入输出接口以便进行协同仿真,具体过程如下
1. 启动 Sketch 并打开 throttle_control_system.ai_sch 文件,如下图所示:

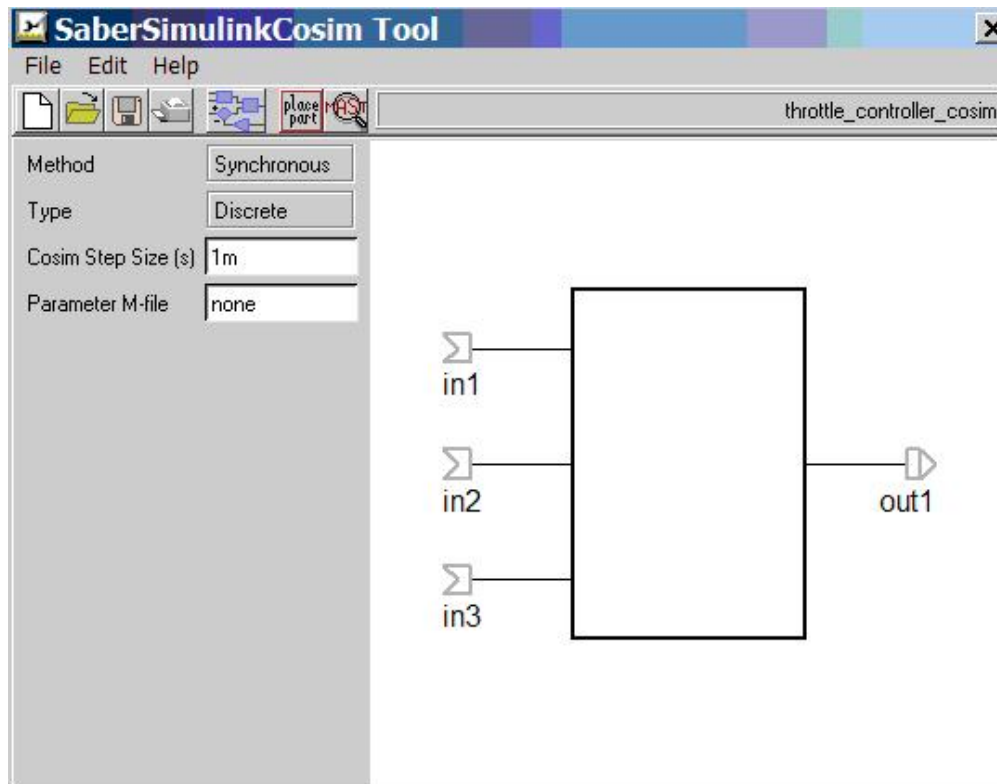


2. 删除图中的 throttle_controller 符号,如下图所示:



3 在 Sketch 启动 SaberSimulinkCosim Tool,并在其界面中选择 File/Import Simulink 命令,在弹出的对话框中选择 throttle_controller_cosim.mdl 文

件,SaberSimulinkCosim Tool 会自动为该 MATLAB 模型建立相关 Saber 符号,如下图所示:



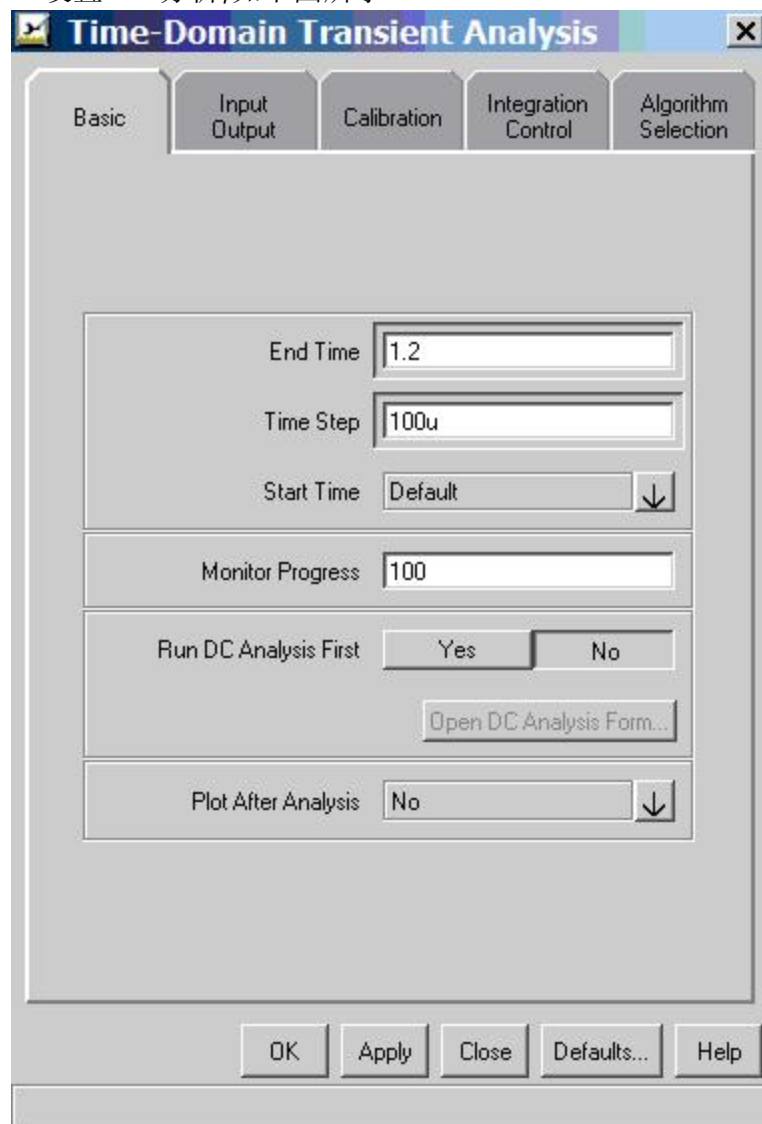
注意: 上图中左上方的 Cosim Step Size(s)栏可以设置 Saber 和 SIMULINK 数据同步的步长,默认值为 1ms, 根据系统时间常数来设置.

4 保存上一步创建的符号并利用 Sketch 中的 Schematic/Get Part/By Symbol Name 命令将该符号放入第 2 步修改好的原理图中,完成连线后,将该图另存为 throttle_control_system_cosim.ai_sch.

接下来可以在 Sketch 环境中运行 Saber/Simulink Co-Sim 了,具体过程如下:

1.对 throttle_control_system_cosim.ai_sch 执行 Design/Netlist 命令,之后运行 Design/Simulate 命令,如果一切顺利,在这一过程中自动会启动 MATLAB.

2.设置 TR 分析,如下图所示:



3.单击OK按钮,分析结束后可以在 Scope 中查看分析结果,这里就不贴出来了,有兴趣的网友可以自己去试试.

从整个 Saber/Simulink 协同仿真的过程看,关键是要合理的定义 Simulink 和 Saber 的接口,把握好这个环节,协同仿真就能正常工作了.在整个协同仿真过程中,Saber 作为主机调用 Simlink,从仿真设置到观察结果都可以在 Saber 环境中完成,Simulink 只是做后台运行.那么如果想在 Saber 中观察 Simulink 部分仿真的内部信号如误差信号、PID 输出等,该如何处理呢?这就涉及到 Saber 和 Simulink 的另一种接口形式-静态数据交换,在 Saber 中是通过 Saberlink 接口实现的,在以后的博客中会有这方面的介绍.