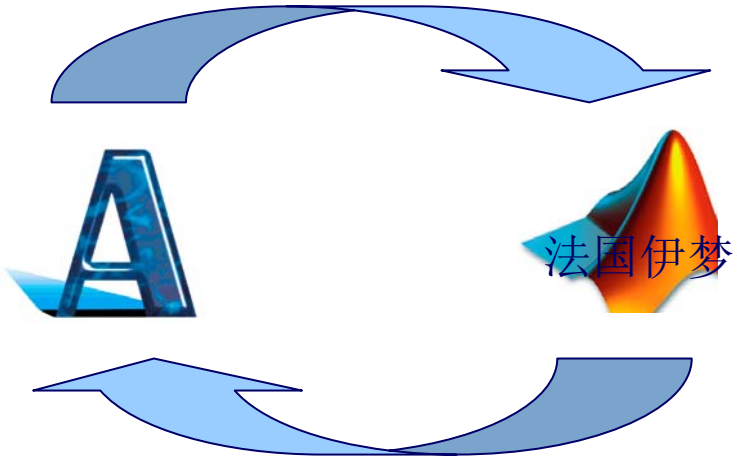
Matlab® & Simulink®

接口



法国伊梦镜公司上海代表处

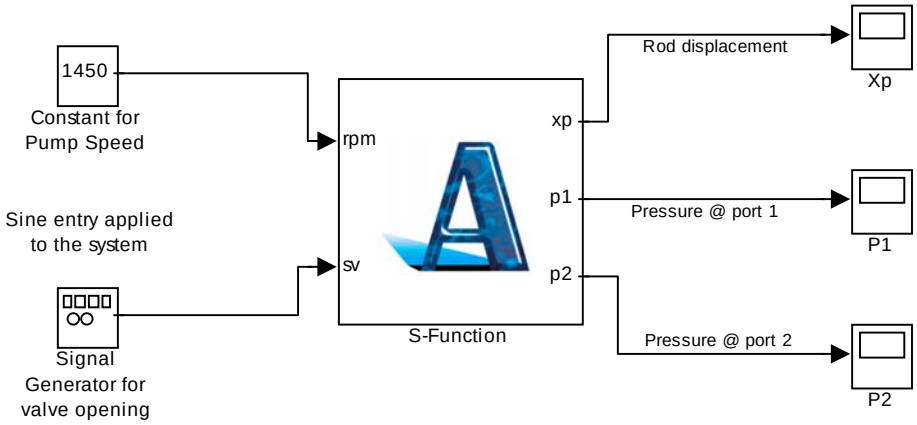
内容

- Matlab® ↔ AMESim®：采用预定义的M文件从Matlab中控制AMESim
- 将状态空间矩阵输入至AMESim中
- AMESim® → Simulink® - S-函数: 将AMESim模型输入至Simulink环境中
- AMESim® → Simulink® - 共仿真: AMESim和Simulink的共仿真

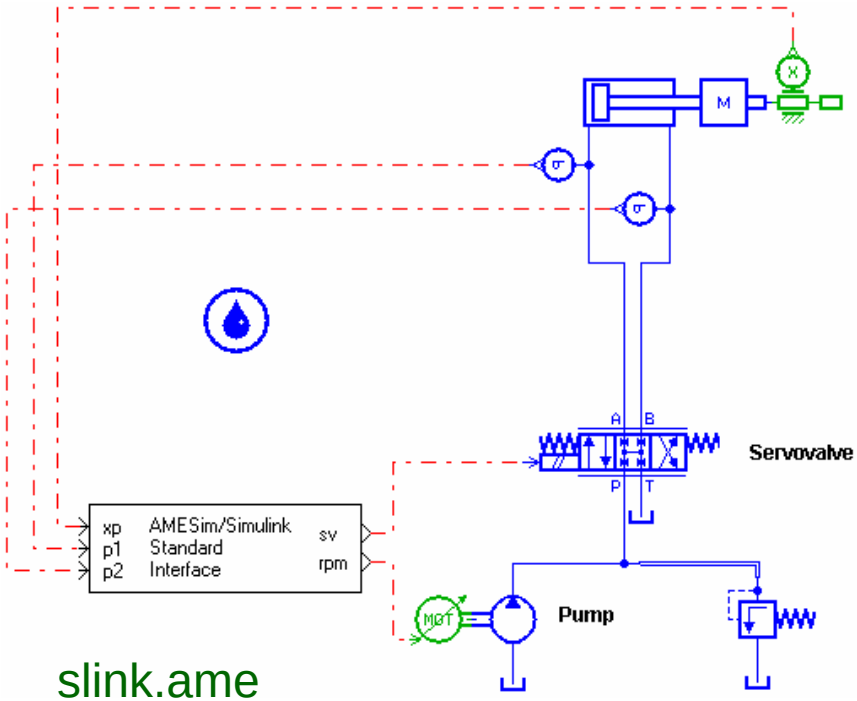
AMESim® – Matlab® & Simulink® 接口

Use Ode15s to run this model

AMESim-Simulink Standard Interface



- ✓ 从Matlab中采用M文件控制AMESim
- ✓ S-函数
- ✓ 共仿真



slink.ame

Matlab® ↔ AMESim®

- AMESim® ↔ Matlab®
  - ✓ 将AMESim的仿真结果输入至Matlab中用于特殊的后处理
  - ✓ 互换传递函数(雅可比Jacobian矩阵)
  - ✓ 在Matlab中获取AMESim的参数
  - ✓ 从Matlab中修改AMESim参数
  - ✓ 从Matlab中修改AMESim仿真运行参数
  - ✓ 从Matlab中运行AMESim模型
  - ➔ 可以从Matlab中完全控制AMESim仿真， 例如从Matlab中定义批处理运行或者在Matlab中定义优化方案（除了AMESim内置的优化功能之外）

Matlab® ↔ AMESim®

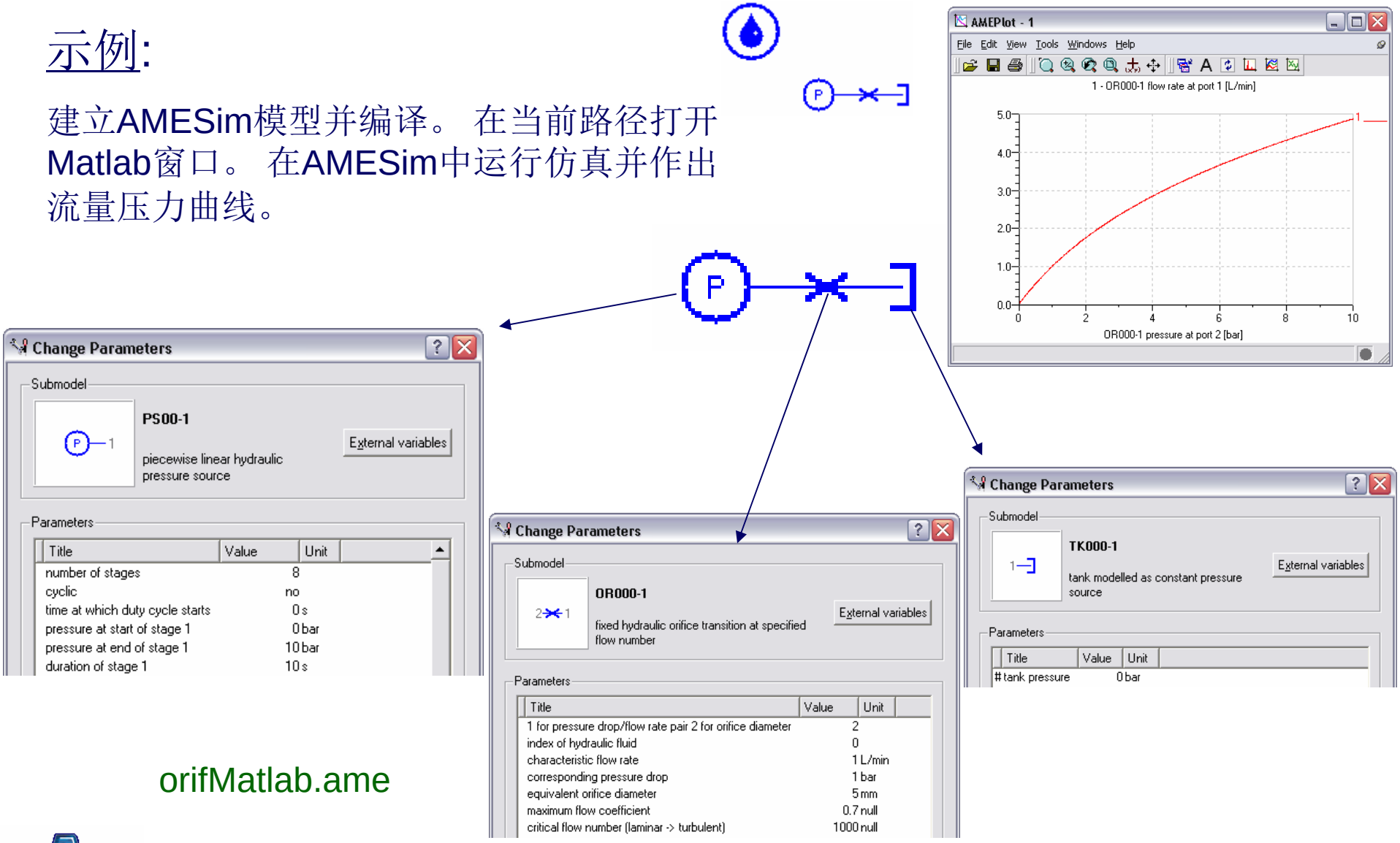
在Matlab环境中（所有的这些M文件是在%AME%/scripting\matlab\amesim\中， 因此需要在Matlab中通过Set Path将该路经包含进来。）：

```
>> help amesim
  AMESim-MATLAB Interface toolbox.

  AMESim --> MATLAB Interface.
    ame2data     - Load AMESim format plot file
    ame2ma       - Create nickname for each variable of AMESim .results file
    amebode      - Bode frequency response for linearized AMESim systems
    ameloadj     - Load AMESim .JAC format jacobian files
    ameloadt     - Load AMESim .RESULTS format temporal files

  MATLAB --> AMESim Interface
    data2ame     - Save data in file readable by AMESim plot facility
    fx2ame       - Save table in file for 1-D interpolation AMESim function
    fxy2ame      - Save table in file for 2-D interpolation AMESim function
    ss2ame       - Save state space matrix on an external file readable by AMESim
    tf2ame       - Save transfer function on an external file readable by AMESim

  Batch facilities for AMESim
    amegetp      - Get AMESim parameters
    amegetgpar   - Get AMESim global parameters
    amegetcuspar- Get AMESim parameters for a customized submodel/supercomponent
    amela        - Set linearization times
    ameputp      - Set an AMESim parameter
    ameputgpar   - Set AMESim global parameters
    ameputcuspar- Set AMESim parameters for a customized submodel/supercomponent
    amerun       - Run an AMESim exectuable
```

Matlab® ↔ AMESim®

示例:

建立AMESim模型并编译。 在当前路径打开Matlab窗口。 在AMESim中运行仿真并作出流量压力曲线。



orifMatlab.ame

Matlab® ↔ AMESim®

- 从Matlab®中输入AMESim仿真结果:

```
>> [R,S]=ameloadt('orifMatlab');

There are 9 variables
There are 101 points per variable
>>
```

返回模型中变量的数量以及每个变量记录的数据点数:

Time simulation(10)/Communication interval(0.1)+1

- ✓S 矢量， 存放变量的名字
- ✓R 矩阵， 存放每个变量的仿真结果

Matlab® ↔ AMESim®

- S矢量和R矩阵

1

```
S =

time [s]
OR000_1 cross sectional area [mm**2]
OR000_1 flow coefficient (Cq)
OR000_1 flow number (lambda)
OR000_1 flow rate at port 1 [L/min]
OR000_1 mean fluid velocity [m/s]
OR000_1 sign reversed duplicate of flow rate at port 1 [L/min]
PS00_1 user defined duty cycle pressure [bar]
TK000_1 tank pressure [bar]
```

2

```
>> R(:,1:3)

ans =

         0    0.1000    0.2000
    1.7052    1.7052    1.7052
         0    0.2338    0.3247
         0  119.1246  168.4679
         0    0.1161    0.2279
         0    1.1343    2.2277
         0   -0.1161   -0.2279
         0    0.1000    0.2000
         0         0         0
```

Matlab® ↔ AMESim®

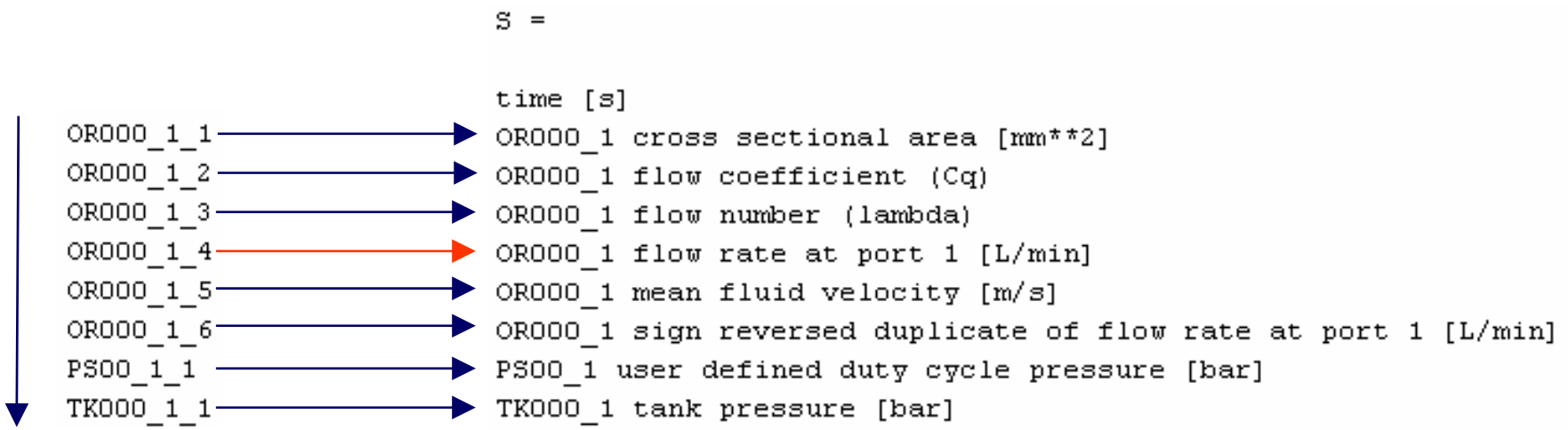
- 以子模型名字命名为每个变量创建一个矢量: ‘ame2ma’

```
>> ame2ma
>> whos
  Name                      Size                    Bytes  Class

  OR000_1_1                 1x101                     808  double array
  OR000_1_2                 1x101                     808  double array
  OR000_1_3                 1x101                     808  double array
  OR000_1_4                 1x101                     808  double array
  OR000_1_5                 1x101                     808  double array
  OR000_1_6                 1x101                     808  double array
  PS00_1_1                  1x101                     808  double array
  R                         9x101                    7272  double array
  ResultsFromAMESim         9x101                    7272  double array (global)
  S                         9x62                     1116  char array
  TK000_1_1                 1x101                     808  double array
  VarNamesFromAMESim        9x62                     1116  char array (global)
  ans                       1x2                        16  double array
  t                         1x101                     808  double array
```

- 现在Matlab® 从R矩阵中创建了8个新的矢量。

Matlab® ↔ AMESim®

- 所创建的矢量对应于:

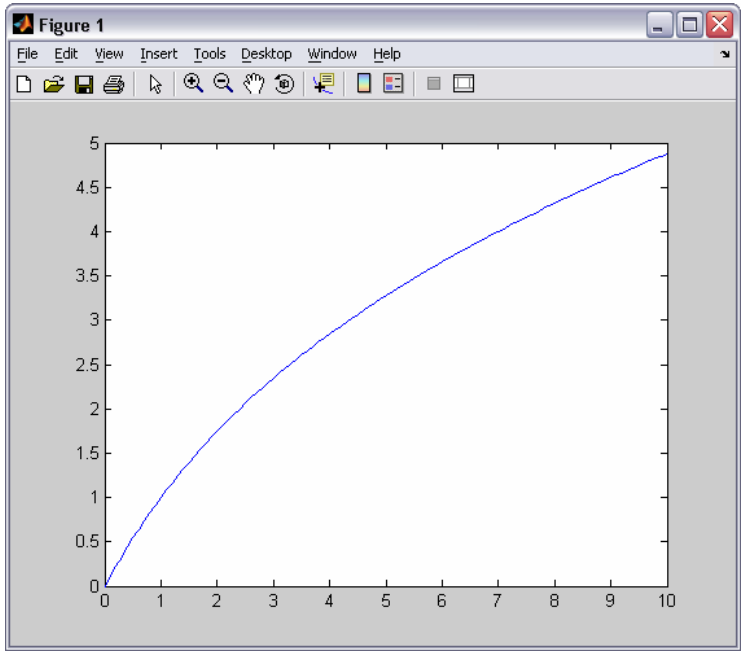


和S矢量中排列的顺序一致

Matlab® ↔ AMESim®

- 在Matlab® 中做出流量/压力曲线：

```
>> plot(PS00_1_1,OR000_1_4)
```



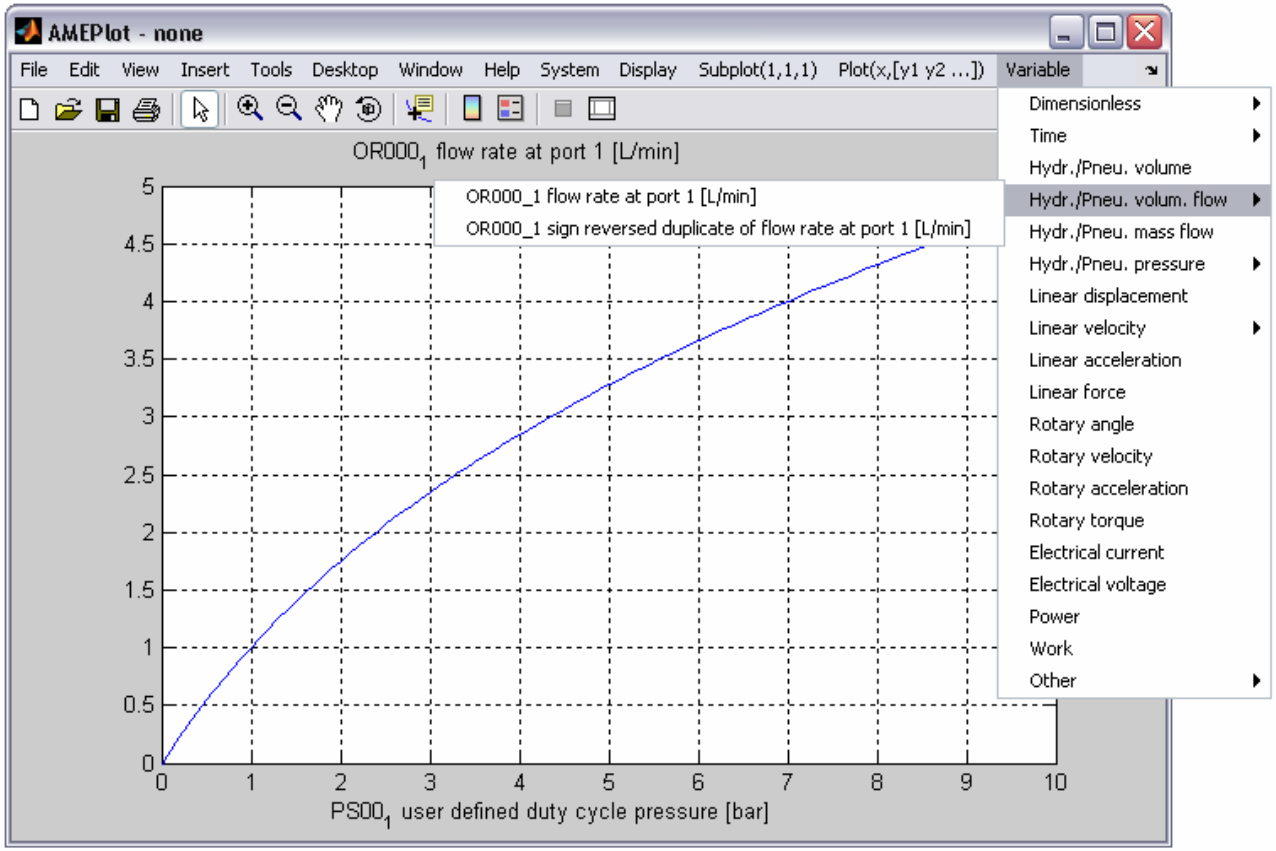
- 您可以使用所有Matlab® 的功能来处理该曲线

例如：加标题，注释， 最大值， 改变图的设置， 保存…

Matlab® ↔ AMESim®

- 另外一种采用Matlab做出AMESim仿真结果的方法为：

```
>> ameplot('orifMatlab')
```



可以从Matlab中选择变量，子图和显示等

Matlab® ↔ AMESim®

- 访问AMESim模型中所有参数:

  >> amegetp('orifMatlab')

- 对于给定子模型的所有参数：

```
>> amegetp('orifMatlab','OR000')
OR000 instance 1 characteristic flow rate [L/min]
 = 1
OR000 instance 1 corresponding pressure drop [bar]
 = 1
OR000 instance 1 equivalent orifice diameter [mm]
 = 5
OR000 instance 1 maximum flow coefficient [null]
 = 0.7
OR000 instance 1 critical flow number (laminar -> turbulent) [null]
 = 1000
OR000 instance 1 1 for pressure drop/flow rate pair 2 for orifice diameter
 = 1
OR000 instance 1 index of hydraulic fluid
 = 0
```

请在Matlab窗口中输入: *'help amegetp'* 以获取详细的说明

Matlab® ↔ AMESim®

- 从Matlab中运行AMESim模型:

```
AMERUN('SYS') starts the executable SYS_

AMERUN('SYS',TO,Tf,Cint,Mts,Tol,Stp,Ord) starts the executable SYS_ with
  Start time              = TO
  Final time              = Tf
  Communication interval = Cint
  Maximum time step       = Mts
  Tolerance               = Tol
  Step                    = Stp
  Order                   = Ord

The two last arguments are used by the fixed step integrator.

AMERUN('SYS',...,OPTION) starts the executable SYS_ and uses
the following options:

  OPTION(1) Error control
            0 : Mixed error test
            1 : Relative error test
            2 : Absolue error test

  OPTION(2) Monitor time
            2 : No output
            0 : Time output

  OPTION(3) Discontinuity printouts
            0 : No extra discontinuity printouts
            1 : Extra discontinuity printouts
```

Matlab® ↔ AMESim®

AMERUN Options (1):

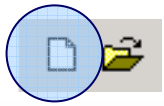
```
OPTION(4) Run statistics
          0 : No run statitics
          1 : Run statitics

OPTION(5) Run types
       Bit 0: Continuation Run (0: off, 1: on)
       Bit 1: Use old final values (0: off, 1: on)
       Bit 2: Stabilizing run (0: off, 1: on)
       Bit 3: Dynamic run (0: off, 1: on)
       Bit 4: Hold inputs constant (0: off, 1: on)
       Bit 5: Standard or fixed step integrator (0: standard, 1: fixed step)
       Bit 6: Fixed step integrator method (0: Adams Bashforth, 1: Runge Kutta)
       Bit 8: Optimized solver or in compatibility mode: (0: Optimized, 1: compatibility mode)

Examples:
          4 : Stabilizing run, new run
          6 : Stabilizing run with old final values
          8 : Dynamic run, new run (probably the most used)
          9 : Dynamic run, continuation run
         10: Dynamic run with old final values
         12: Stabilizing run + dynamic run
         14: Stabilizing run + dynamic run, with old final values
         40: Dynamic run using Adams-Bashforth fixed step integrator
        104: Dynamic run using Runge-Kutta fixed step integrator
        264: Dynamic run, new run, NOT using the optimised solver (compatibility mode)
```

Matlab® ↔ AMESim®

- AMERUN Options (2):

```
OPTION(6) Solver type
          0 : standard solver
          1 : cautious - equivalent to AMESim 2.5.1

OPTION(7) Options for stabilizing runs
       Bit 0: Lock non propagating states automatically
       Bit 1: Diagnostics

OPTION(8) Discontinuity handling
          0 : Normal (default)
          1 : Minimal

OPTION(9) Activity index calculation (optional): not practical
          to change this from Matlab as it needs additional
          treatment. Use the AMESim/AMERun GUI instead.
          0 : Off
          1 : On

If no options are set, the run parameters are not changed
and the run options are set to [0 2 0 0 8 0 0 0 0].  If the
option for the activity index is left out it will keep
the value previously set.
```

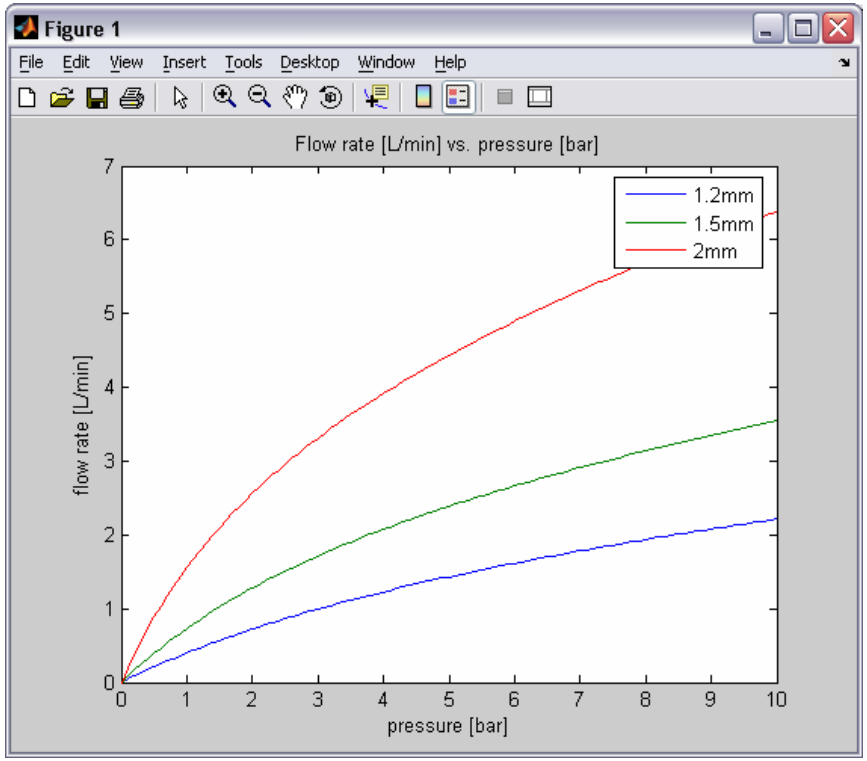
Matlab® ↔ AMESim®

- 在Matlab®中进行批仿真设置
- 打开一个新M文件 

```
%AMESim batch-setup file
%
% orif.m
%
%load the AMESim file's parameter
[R,S]=amegetp('orifMatlab');

%generate the batch and save the new flow rate results
parameter = [1.2 1.5 2];

for i=1:3
    ameputp('orifMatlab','OR000 instance 1 equivalent orifice diameter*',parameter(i));
    [U,V]=amerun('orifMatlab');
    ame2ma
    Batch(i,:)=OR000_1_4;
end

plot(PS00_1_1,Batch);
legend([num2str(parameter(1)) 'mm'], [num2str(parameter(2)) 'mm'], [num2str(parameter(3)) 'mm']);
title('Flow rate [L/min] vs. pressure [bar]');
ylabel('flow rate [L/min]');
xlabel('pressure [bar]');

BatchAme = [PS00_1_1' Batch'];
data2ame(BatchAme,'result_batch');
```

Matlab® ↔ AMESim®

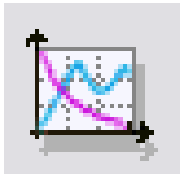
- 结果:
  - ✓在Matlab®, 运行脚本文件‘orif.m’。 生成作图并且流量数据保存至‘*results_batch*’ 文件， 该文件AMESim®可以读取。

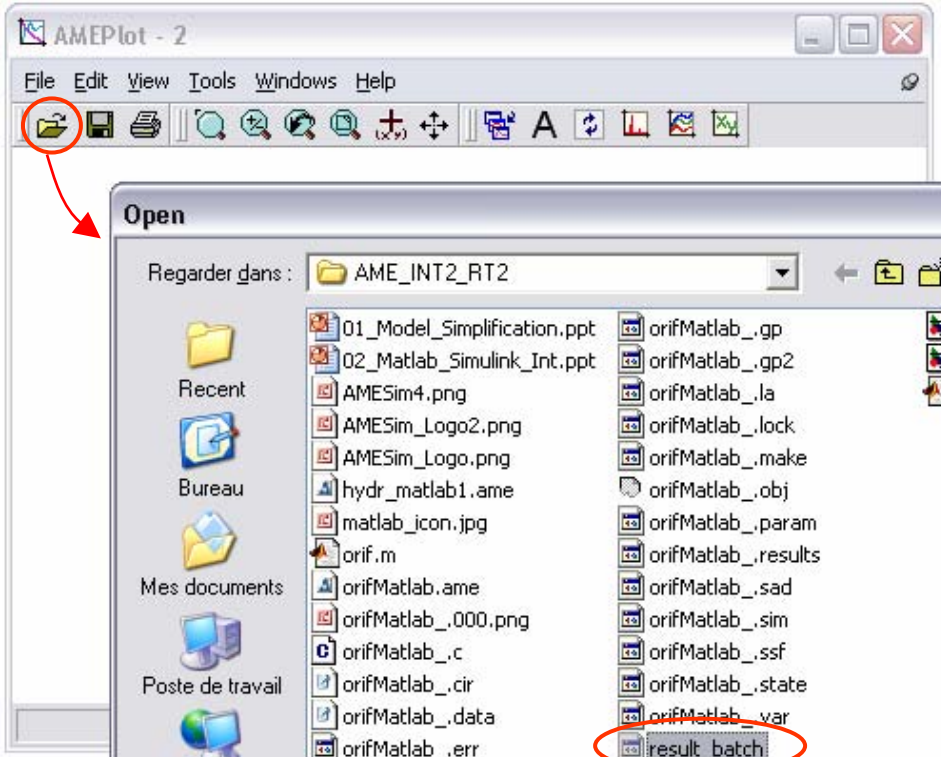


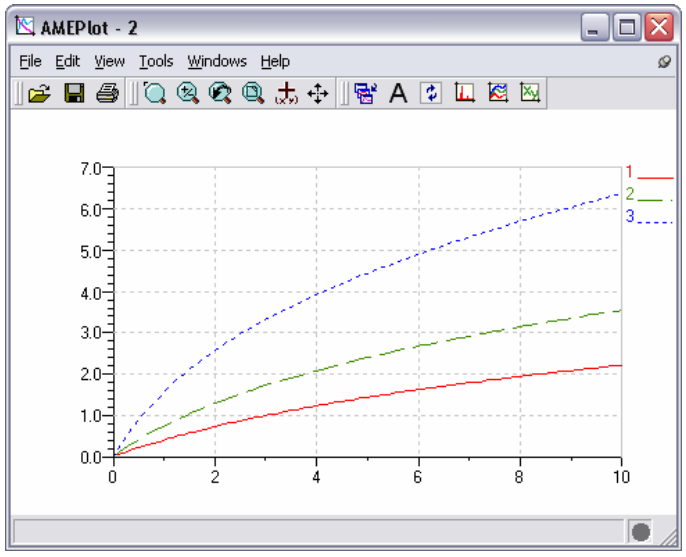
Matlab® plot

Matlab® ↔ AMESim®

- 在AMESim中加载在Matlab中创建的结果文件：





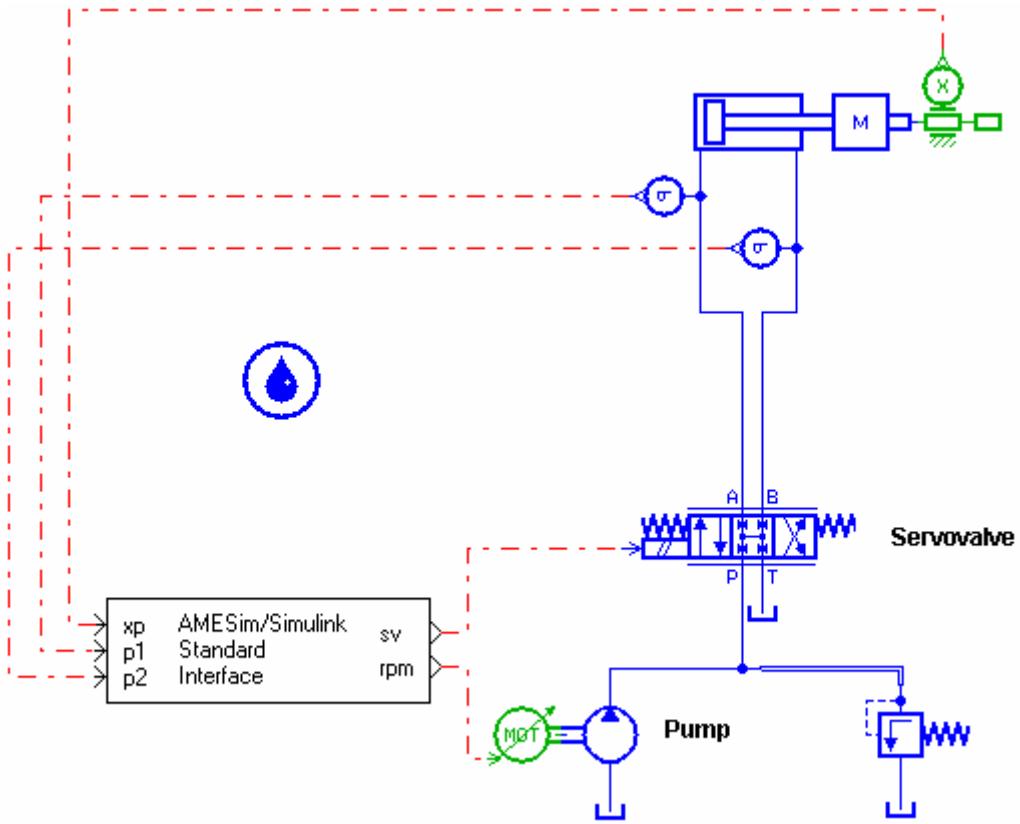


AMESim® plot

Simulink® ↔ AMESim®

- 创建AMESim模型的S函数
- 共仿真：同时采用AMESim® 和 Simulink® 求解器
- 将Simulink的C模型输入至AMESim中

AMESim® → Simulink® - S-Function



slink.ame

AMESim® → Simulink® - S-函数

- 对Windows操作系统:

  需要定义Simulink接口的环境变量:

  - ✓*MATLAB = Matlab® 的安装路径*
    (例如. C:\MATLAB7)
  - ✓*PATH = Matlab® 的bin路径*
    (例如. C:\MATLAB7\bin\win32)

- 对Unix 操作系统:

  采用Unix命令来设置MATLAB环境变量
  (假定MATLAB安装在/opt/matlabr12.1):

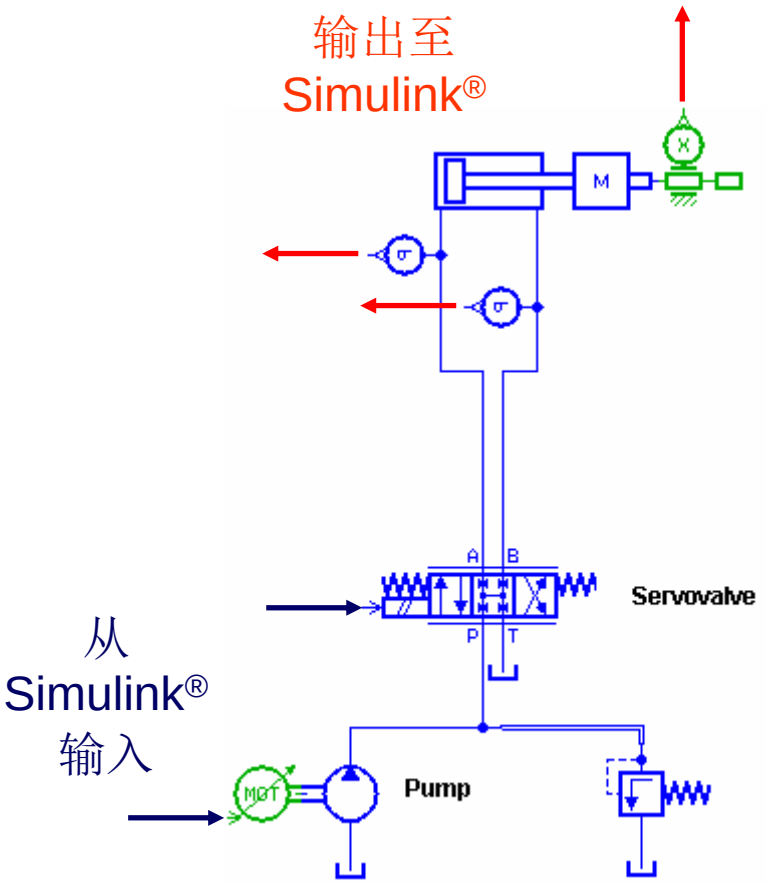
  setenv MATLAB_ROOT /opt/matlabr12.1
  set path=($path $MATLAB_ROOT/bin)
  setenv MATLABPATH $AME/matlab/amesim

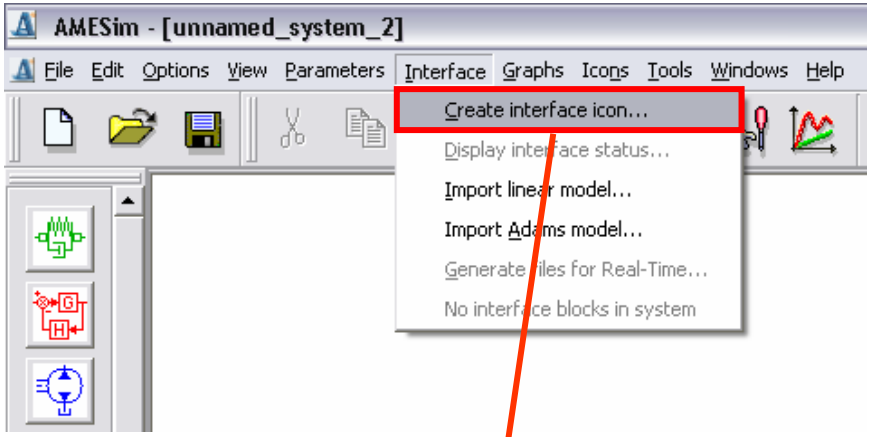
- 在Matlab中定义Matlab接口的Path: (Windows/Unix操作系统均需要)

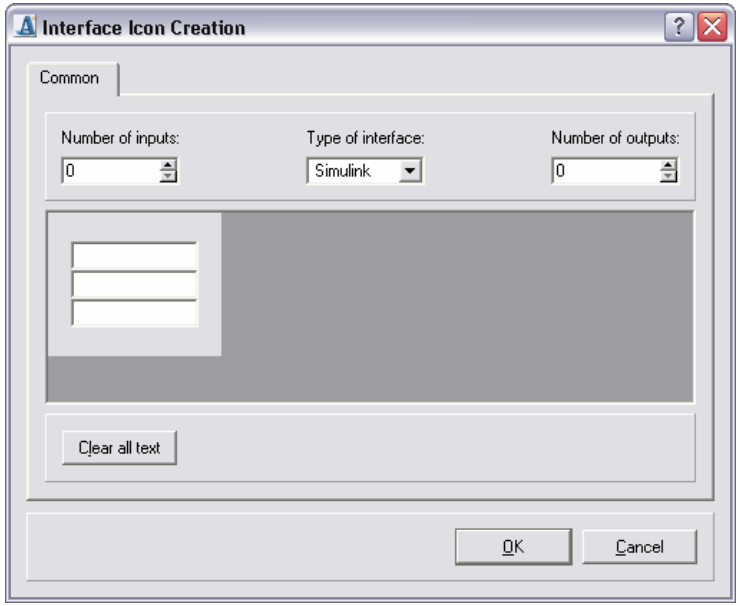
  - ✓*PATH = $AME\matlab\amesim的路径， 其中$AME = AMESim® 安装路径*
    (例如. C:\AMESim4.3.0\matlab\amesim)

AMESim® → Simulink® - S-函数

- 构建系统 (1):



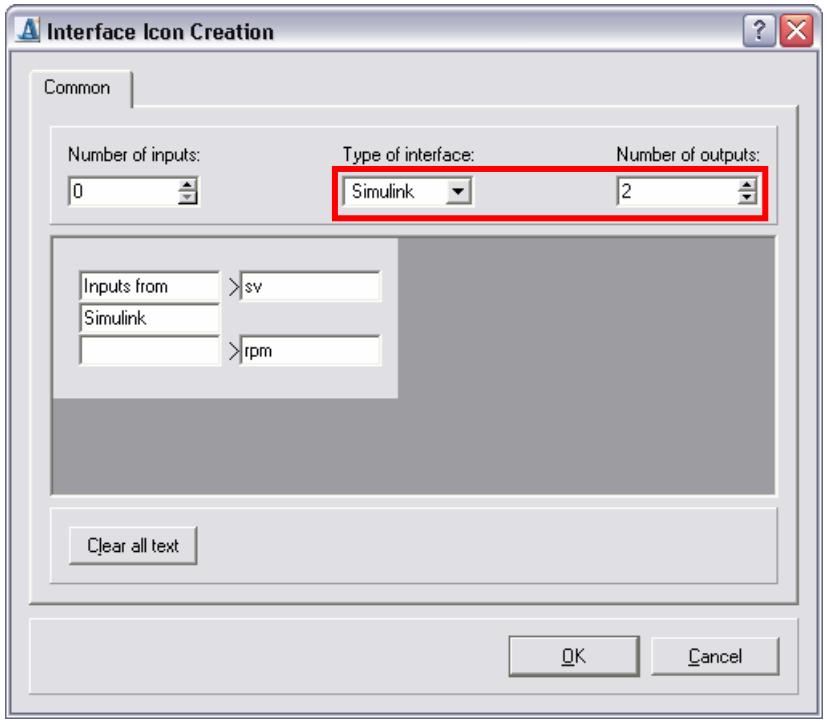


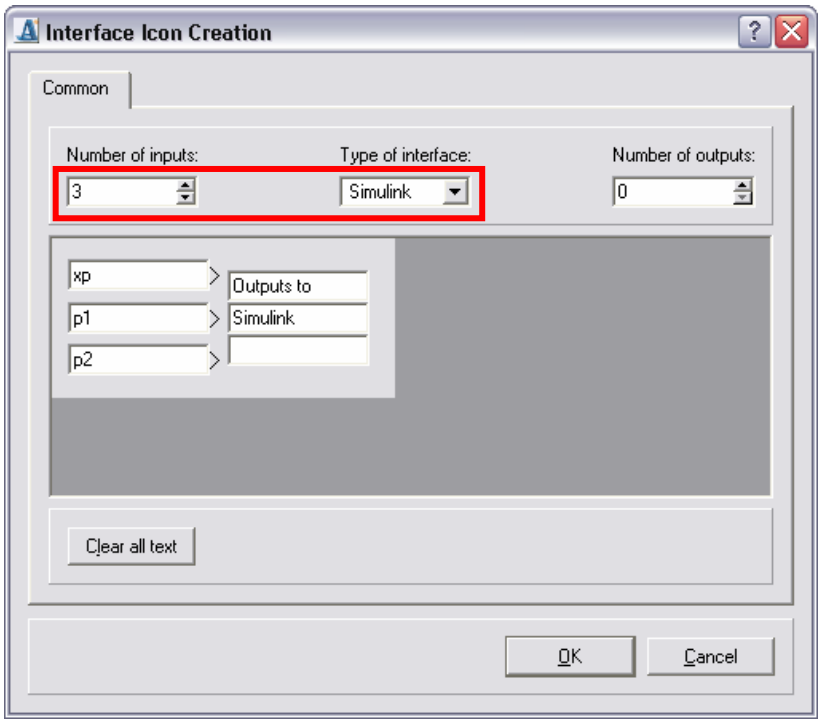


AMESim® → Simulink® - S-函数

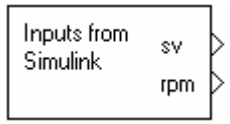
- 构建系统(2a):

分别按照下图所示定义输入和输出方块:

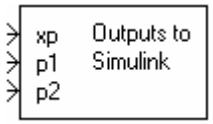




从Simulink®的输入
=
Simulink 的输出



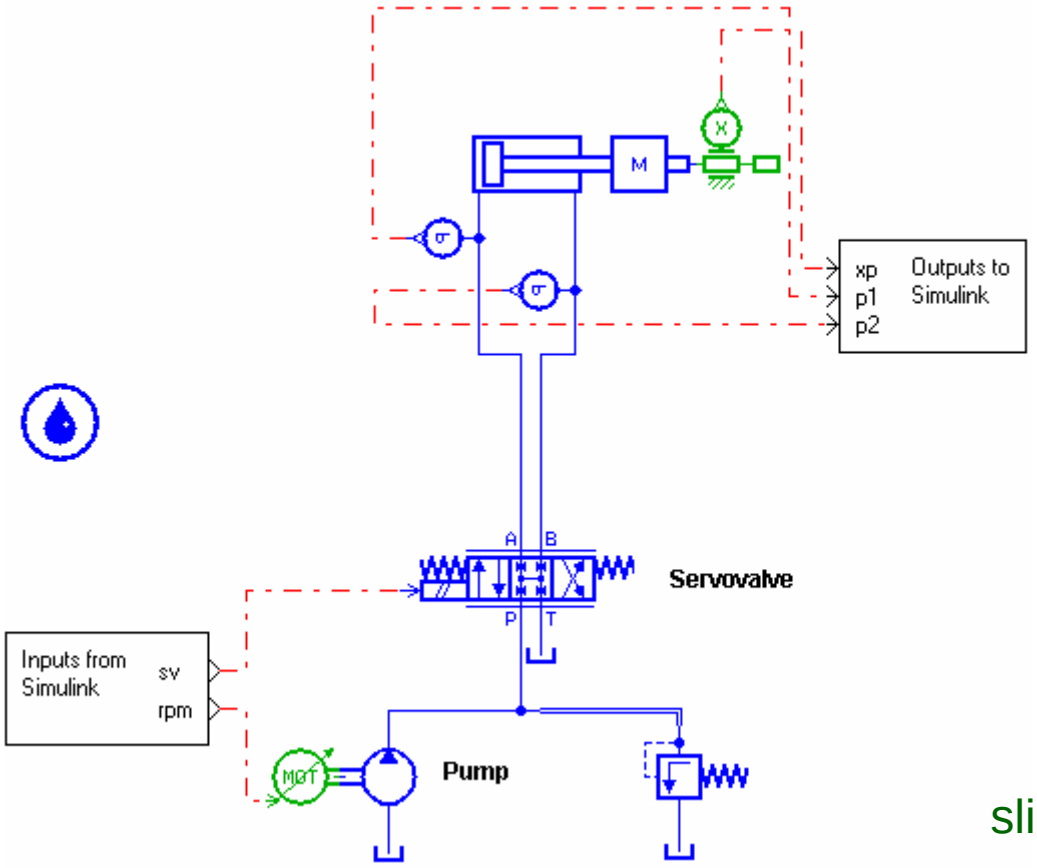
输出至Simulink®
=
Simulink 的输入



AMESim® → Simulink® - S-函数

- 构建系统(2b):

分别按照下图所示连接定义好的输入和输出方块:



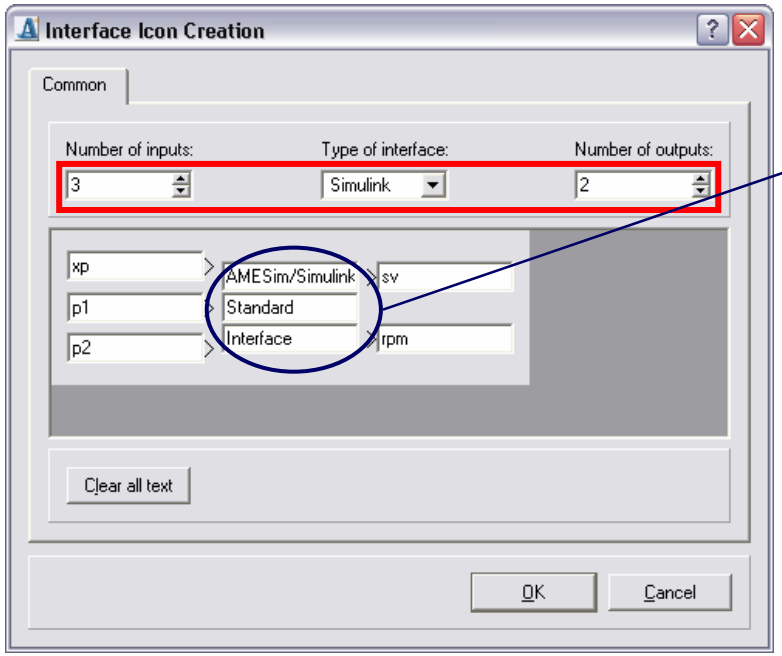
slink2.ame

AMESim® → Simulink® - S-函数

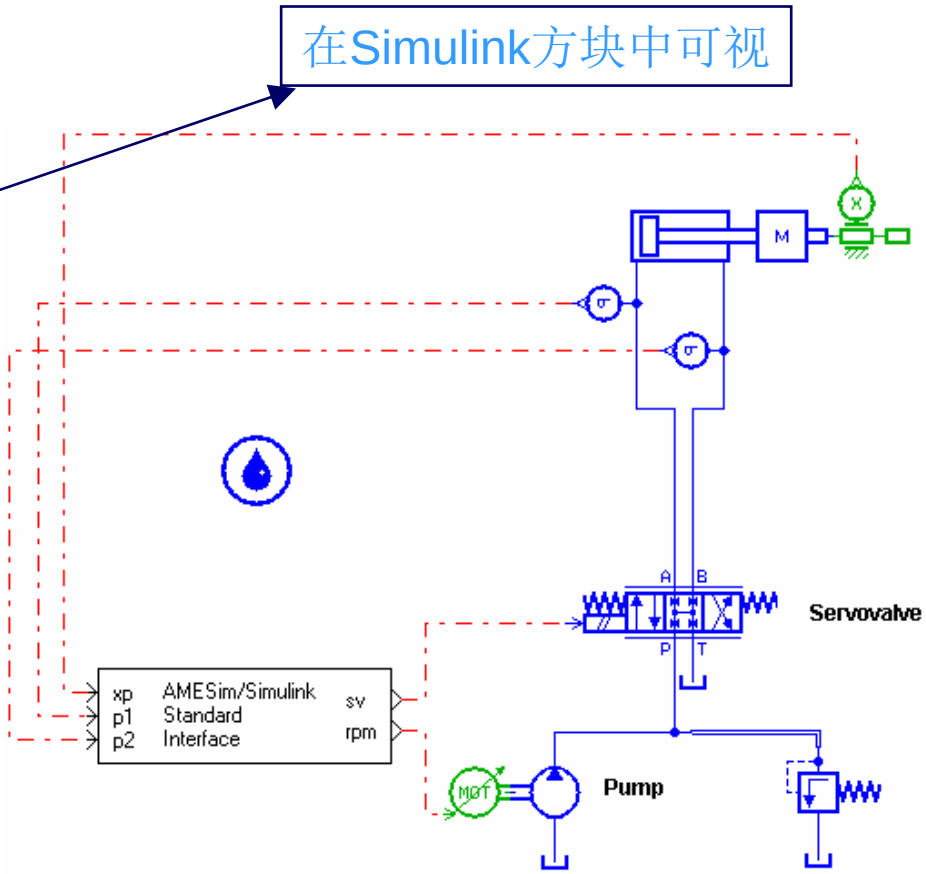
➢构建系统(3):

按下图所示将输入和输出定义成一个方块:

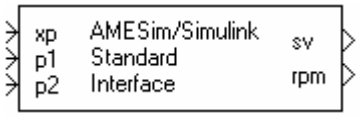
模型看上去相对简单



在Simulink方块中可视



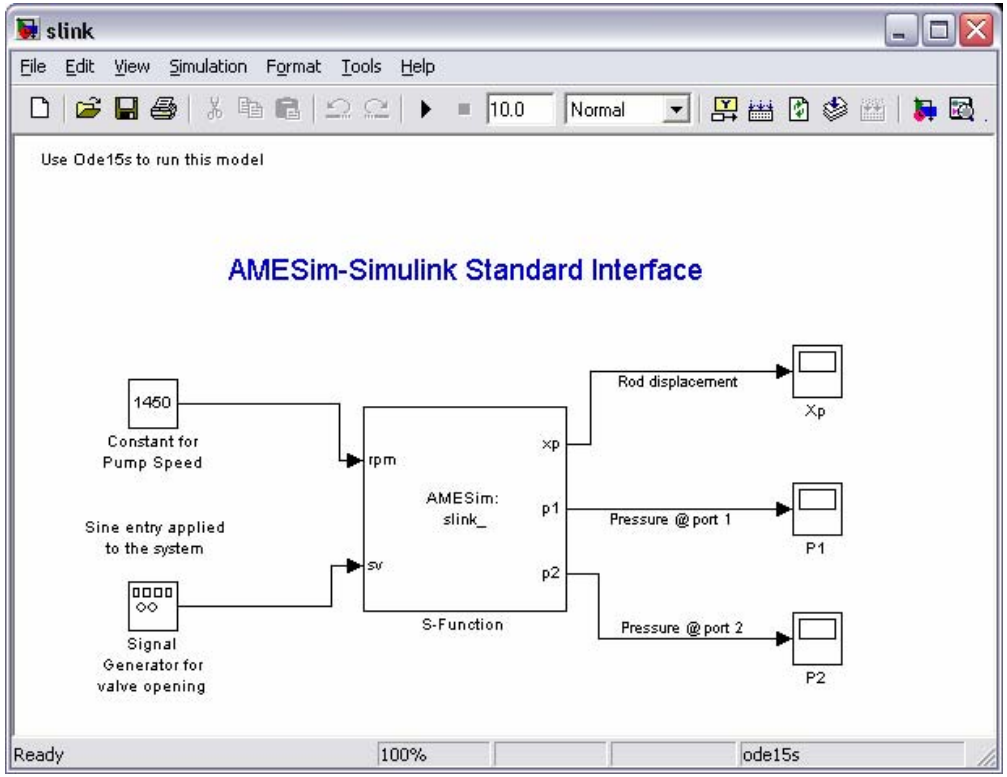
输出



输入

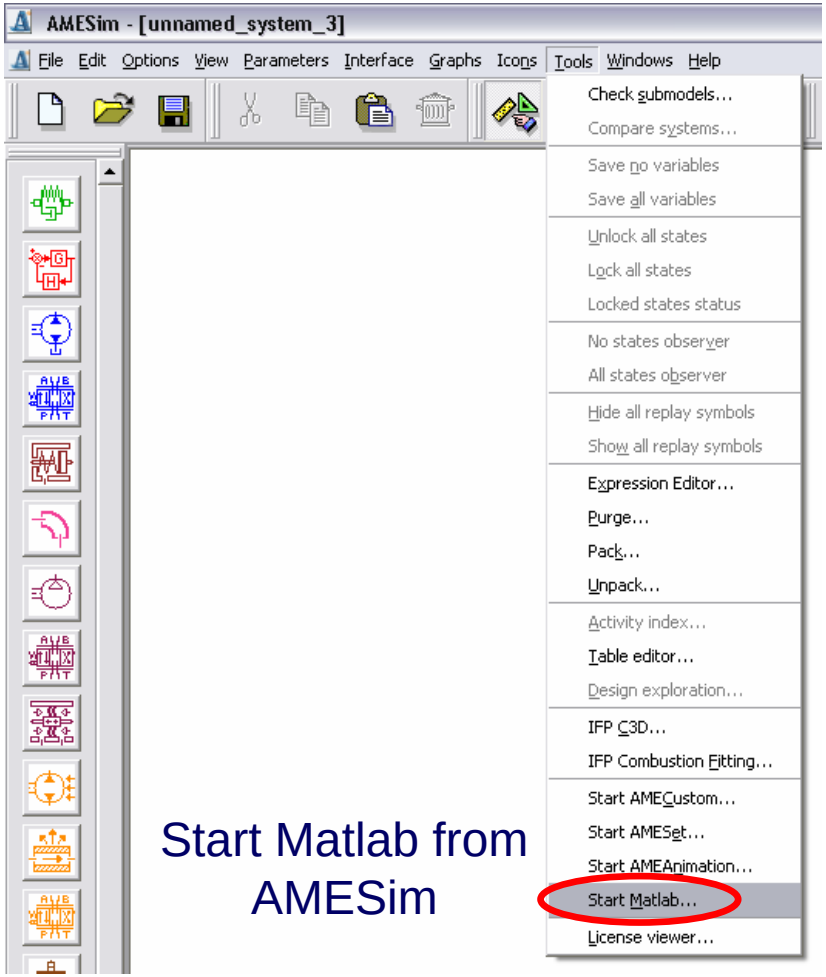
AMESim® → Simulink® - S-函数

- 构建系统(4):



新建一个 .mdl文件

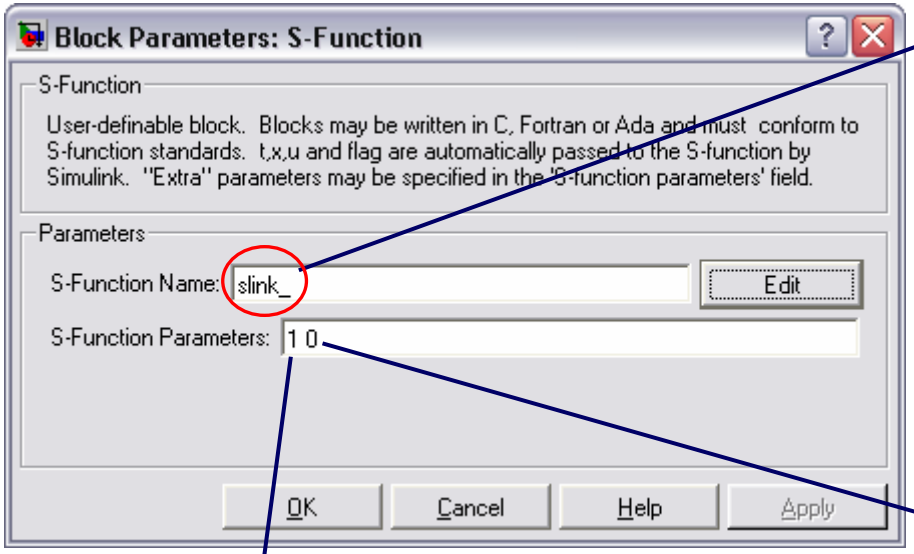
slink.mdl



Start Matlab from AMESim

AMESim® → Simulink® - S-函数

- S-函数参数:



AMESim® 文件 = slink.ame

➔ S-函数的名字 = slink_

1st 参数:

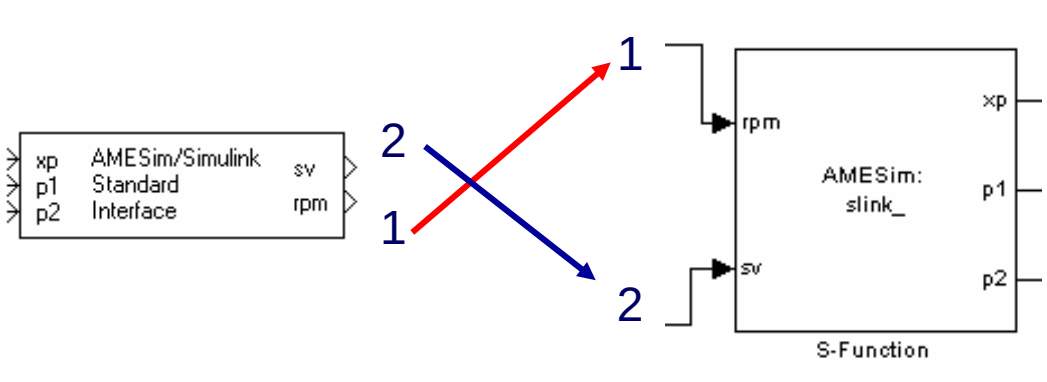
创建AMESim® 结果文件(1)或否(任何其它数字)

2nd 参数:

AMESim® 在结果文件中记录的时间间隔(参数 ≤ 0 : 和在Simulink®中定义的一致)

AMESim® → Simulink® - S-函数

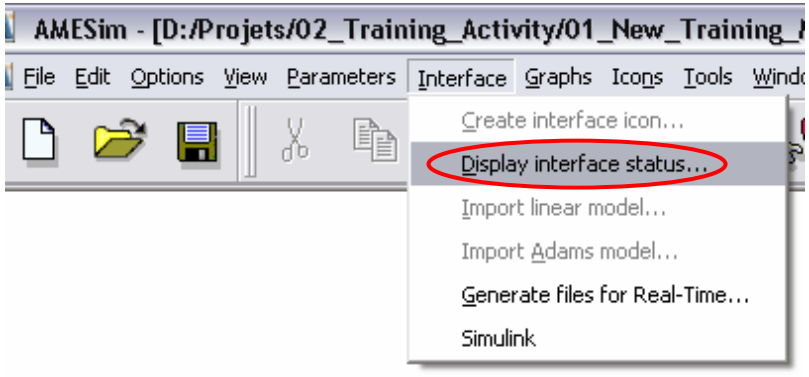
- 构建系统(4):
  - 注意在Simulink®中S函数的输入的顺序和AMESim接口方块中的输出顺序正好相反
  - 在Simulink® 中S函数的输出和AMESim接口方块中的输入顺序一致。
  - 这是由于在AMESim和Simulink中定义端口序号的约定不一致造成的。 在AMESim中采用逆时钟， 而Simulink中采用顺时钟。

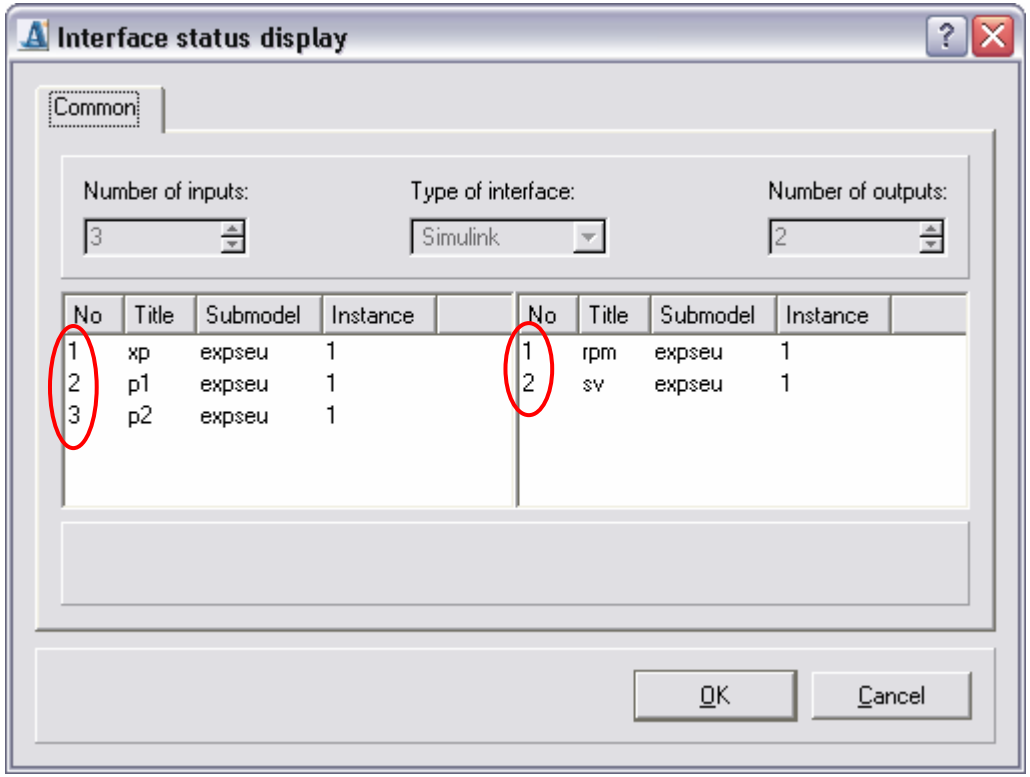


AMESim® → Simulink® - S-函数

- 构建系统(5):

输入输出变量的序号可以通过“Display Interface Status”来查看。



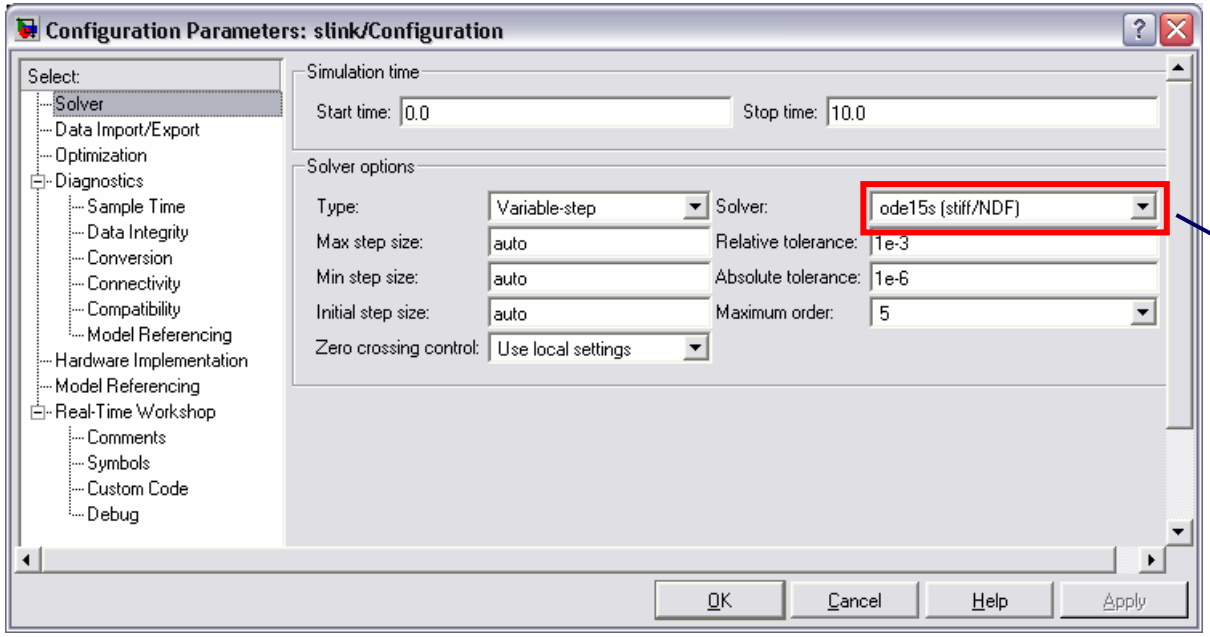


AMESim® → Simulink® - S-函数

- 为了能够在另一台计算机中运行， 请确认下述所有文件均和Simulink模型文件一并拷贝了。

*.oil, *.data, *.gp (如果您有全局参数的话) *.dll (Windows NT/2000), *.mexsg(Unix-SG), *.mexsol (Unix-Sun), *.mexhp7 (Unix-HP), *.var, *.param

如果用户需要在Matlab中更改参数的话， 那么最后2个文件也需要 (*.var and *.param) 。

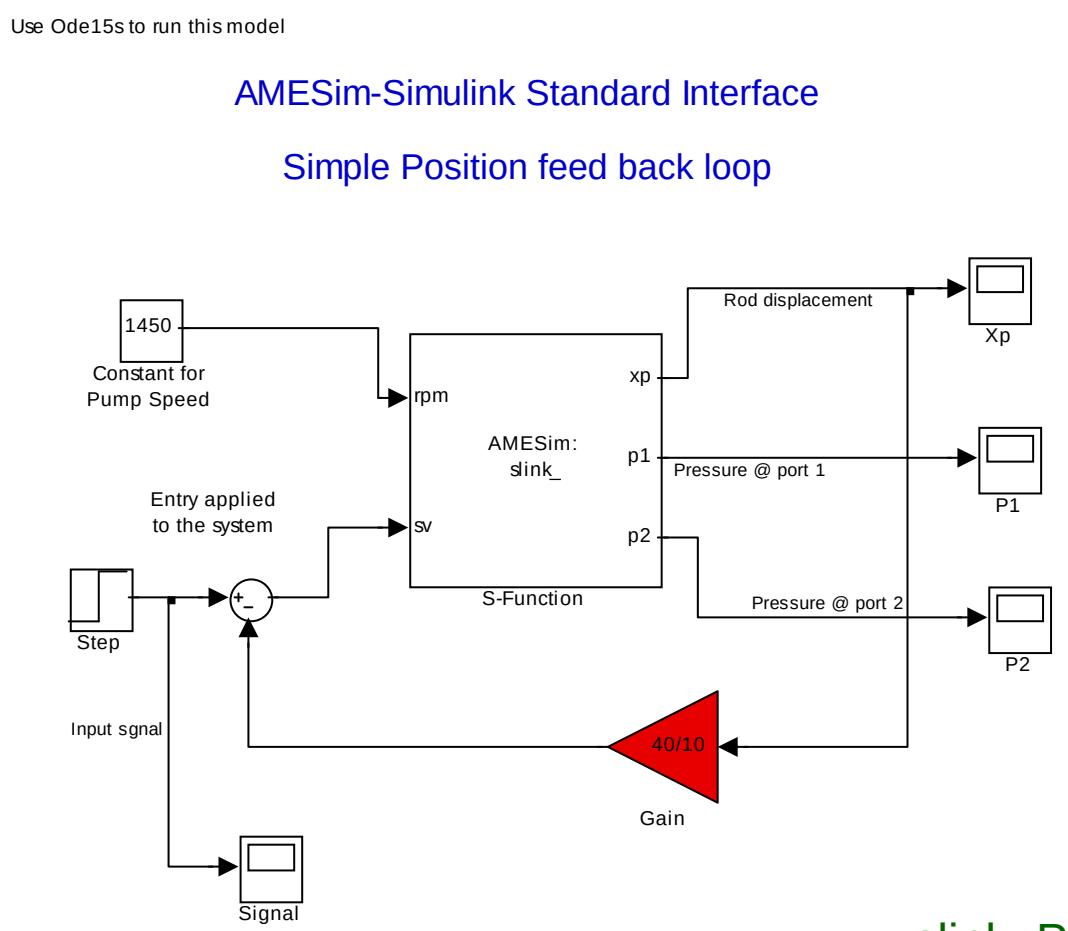


Simulink 仿真运行参数

液压系统通常刚性的， 为了在Simulink中求解此类系统，需要选择考虑刚性的积分算法(ode23s, ode15s)。

AMESim® → Simulink® - S-函数

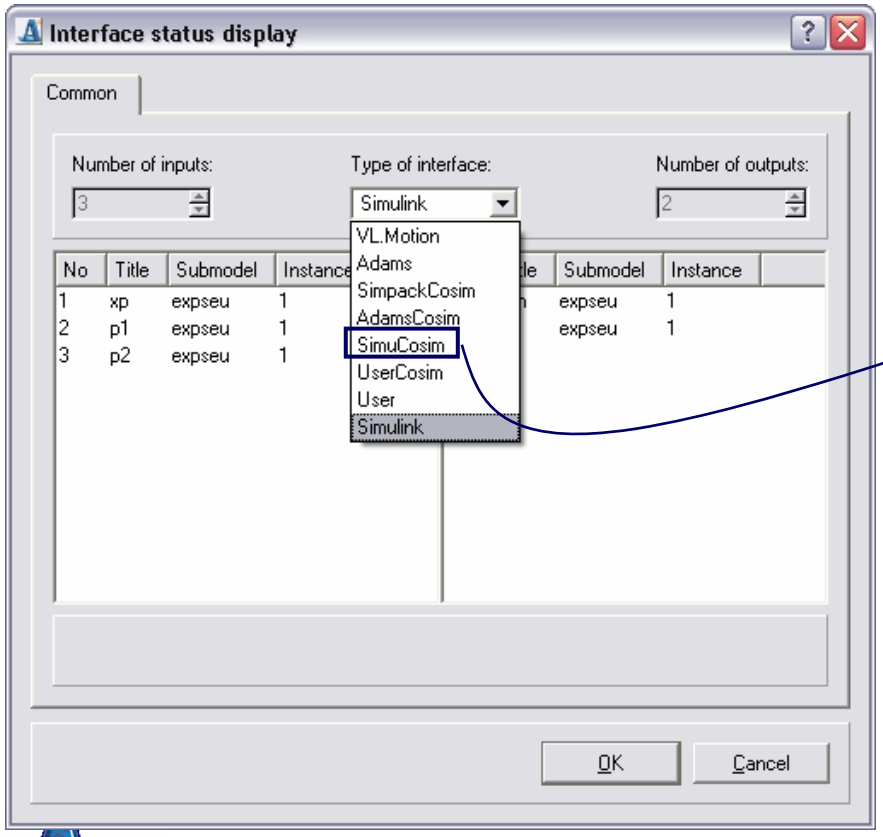
- 更进一步 ... 考虑系统的闭环控制

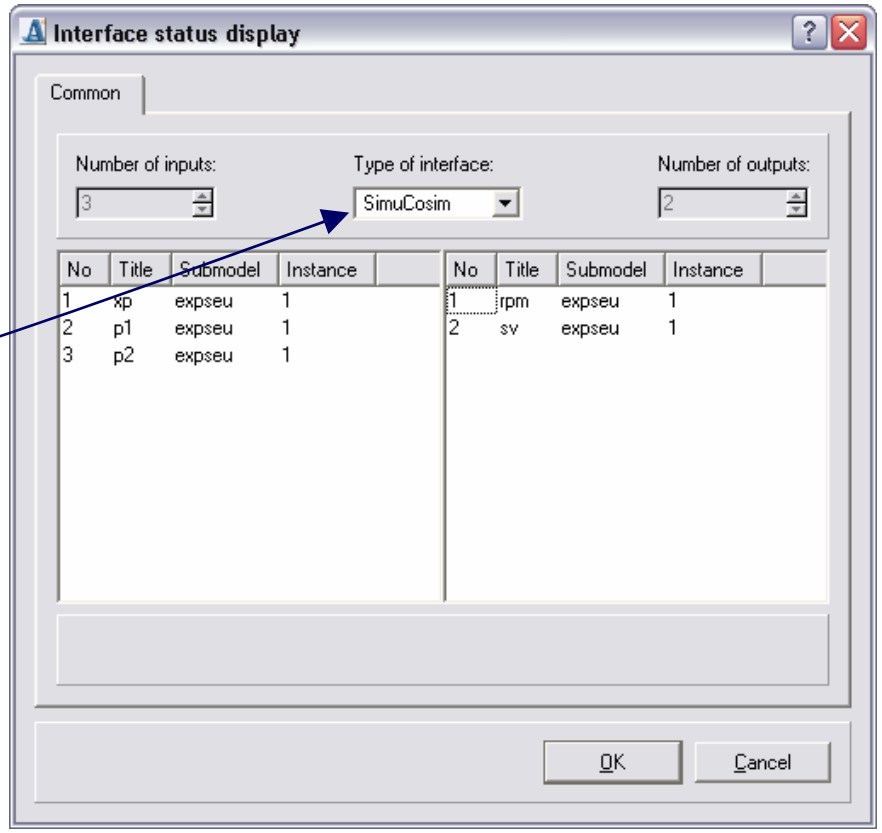


slink_P.mdl

AMESim® → Simulink® - 共仿真

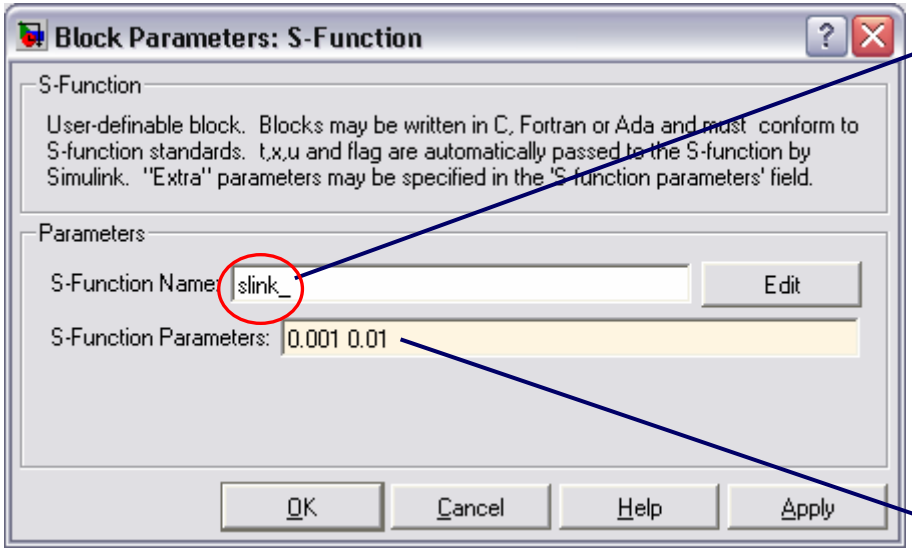
- 在Sketch或者Submodel Mode中，通过“Display interface status”可以改变接口的类型。
- 在编译时将创建一个新的.dll文件。





AMESim® → Simulink® -共仿真

- S-函数参数 (共仿真):



AMESim® 文件 = slink.ame

➔ S-函数的名字 = slink_

参数:

见下页表格

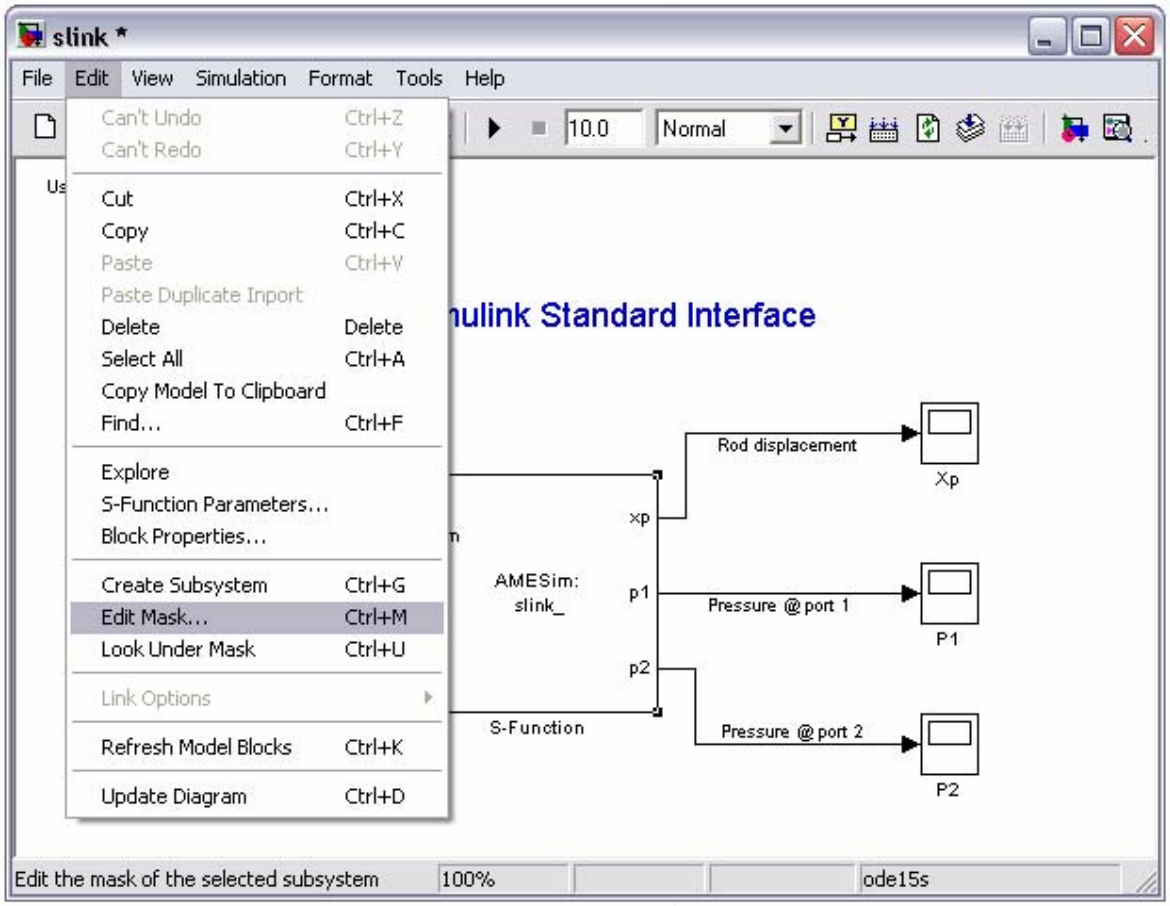
AMESim® → Simulink® -共仿真

- S-函数(共仿真):

#	Parameter	Description	Default value
1	Sample time	Time between exchange of values between AMESim® and Simulink®	Required, no default value
2	AMESim® communication interval	As in AMESim®	Required, no default value
3	Tolerance	The same as in the AMESim run parameters popup	1.0e-5
4	Max time step	The same as in the AMESim run parameters popup	1.0e20s
5	Time range	Helps DASSL to decide initialize the time step	100s (do not change)
6	Show run statistics	0 or 1, 1 for displaying run statistics	1
7	Extra discontinuity points	0 or 1, 1 for extra discontinuity printouts	0
8	Output details	0 or 1, 1 for output of time on screen (not useful on PC)	0

在S-函数方块中显示图片

- 在Simulink模型中，选择S-函数方块并且选择‘Edit’菜单中的'Edit mask'功能



在S-函数方块中显示图片

- 在Simulink模型中，选择S-函数方块并且选择‘Edit’菜单中的'Edit mask'功能