ENTERPRISE ARCHITECT

用户指南系列

SysML参数仿真

Author: Sparx Systems

Date: 2022-08-30

Version: 16.0

创建于 ENTERPRISE ARCHITECT

目录

SysML参数仿真

Enterprise Architect提供与 OpenModelica 和 MATLAB Simulink 的集成，以支持对 SysML模型在不同情况下的行为方式进行快速而可靠的评估。

OpenModelica 库是提供许多有用类型、函数和模型的综合资源。在Enterprise Architect中创建 SysML 模型时，您可以参考这些库中可用的资源。

Enterprise Architect的 MATLAB 集成通过 MATLAB API 进行连接，允许您的Enterprise Architect仿真和其他脚本根据任何可用的 MATLAB 函数和表达式的值进行操作。您可以通过 Solver类调用 MATLAB，或将您的模型导出到 MATLAB Simulink、Simscape 和/或状态流。

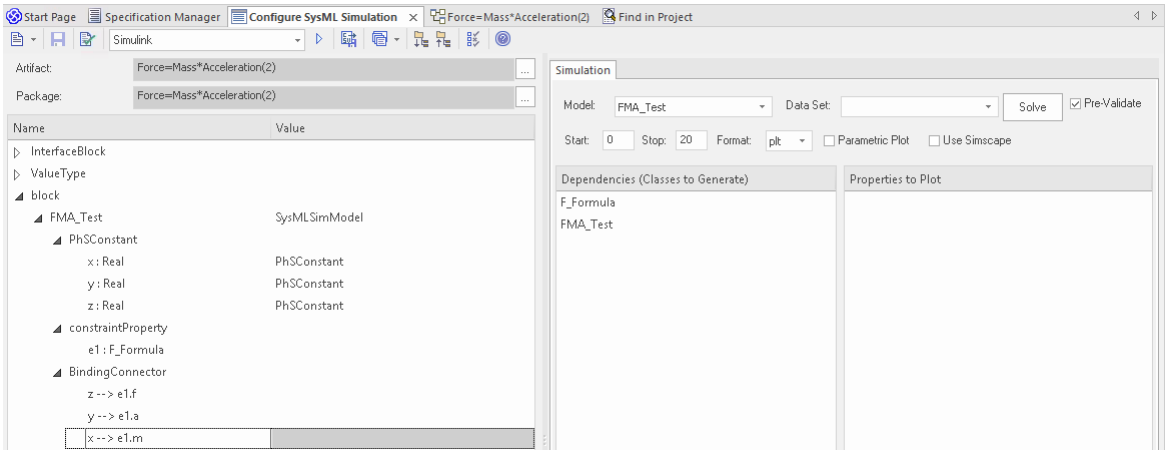
SysML仿真特征

这些部分描述了定义参数模型、使用附加信息对模型进行注释以驱动模拟以及运行模拟以生成结果图的过程。

部分	描述
SysML 参数模型简介	SysML 参数模型支持关键系统参数的工程分析，包括评估关键指标，如性能、可靠性和其他物理特性。这些模型通过捕获基于复杂数学关系的可执行约束，将需求模型与系统设计模型相结合。参数图是专门的内部块图，可帮助建模者将行为和结构模型与工程分析模型（例如性能、可靠性和质量属性）结合起来。 有关 SysML 参数模型概念的更多信息，请参阅 OMG SysML 官方网站及其链接资源。
创建参数模型	概述开发用于仿真的 SysML模型元素、在配置SysML仿真窗口中配置这些元素以及观察仿真结果。
工件适合	Enterprise Architect帮助您扩展 SysML 参数模型的实用性，方法是使用允许模拟模型的额外信息对它们进行注释。然后将生成的模型生成为可以使用 MATLAB Simulink 或 OpenModelica 求解（模拟）的模型。 模拟属性针对您的仿真模型进行工件。这保留了您的原始模型并支持针对单个 SysML模型配置的多个模拟。仿真工件在工件的工具箱上找到
用户接口	SysML 模拟的用户界面在配置模拟仿真主题中进行了描述。
使用数据集进行模型分析	使用仿真配置，SysML块可以有多个针对它定义的数据集。这允许在 SysML模型的模拟上运行可重复的变化。
SysPhS 标准支持	*SysPhS* 标准是交互物理和辅助信号流仿真的 *SysML*扩展。它定义了在 SysML模型和 Modelica模型或 Simulink/Simscape模型之间转换的标准方法，为共享仿真提供了一种更简单的基于模型的方法。请参阅*SysPhS* 标准支持帮助帮助。
例子	为了帮助您了解如何创建和模拟 SysML 参数模型，我们提供了三个示例来说明三个不同的领域。所有三个示例都使用 OpenModelica 库。 *SysML*仿真示例主题中描述了这些示例以及您可以从中学到的东西。

配置SysML仿真

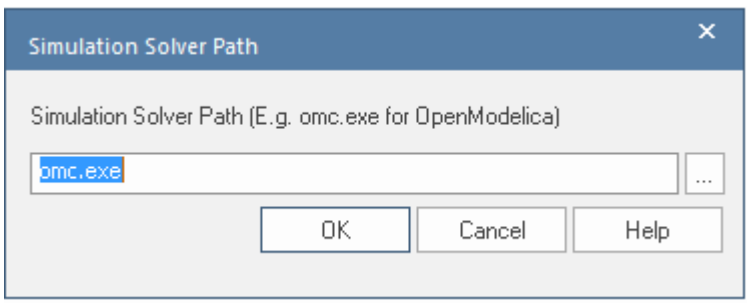
配置SysML仿真窗口是一个界面，您可以通过该界面提供运行时参数以执行 SysML模型的模拟。工件中的模拟基于元素定义的模拟配置。



访问

功能区	仿真>系统行为> Modelica/Simulink >配置管理器
其它	双击具有工件构造型的属性。

工具栏选项

选项	描述
	单击下拉箭头并从以下选项中进行选择： • （如果尚未选择工件配置，则从一个工件配置中选择一个配置和负载） • 创建资产— 创建一个新的工件或选择并加载现有工件 • 选择包- 选择一个包以扫描 SysML 元素以配置模拟 • Reload — 重新加载配置管理器并更改当前包 • 配置仿真求解器 - 显示 仿真求解器路径”对话框，您可以在其中键入或浏览要使用的求解器的路径 

	单击此按钮将配置保存到当前工件。
	单击此图标现在专门针对现在配置验证模型。验证结果显示在系统输出窗口的 "SysML仿真"选项卡中。您还可以选择一个选项以在执行每个模拟之前自动预验证模型。请参阅仿真标签表中的 "预验证"选项。
	单击此图标可展开窗口 "名称"列中层次结构中的每个项目。
	单击此图标可折叠窗口 "名称"列中模型层次结构中所有展开的项目。
	单击此图标可显示可在模拟中抑制的object类型列表。单击要抑制的每个object的复选框，或单击 "全部"按钮以选择要抑制的所有项目。 您还可以使用 "选项"列顶部的过滤器栏来仅显示名称中具有指定字母或文本string的项目。
	单击下拉箭头并选择正在运行仿真的应用程序，例如运行或 Simulink。
	单击此按钮可生成、编译和运行当前配置，并显示结果。
	仿真后，以 plt、mat 或 csv 格式生成结果文件。也就是说，使用文件名： • ModelName_res.mat（OpenModelica 的默认值） • ModelName_res.plt 或 • 模型名称_res.csv 单击此按钮指定Enterprise Architect将复制结果文件的目录。
	单击此按钮可从以下选项中进行选择： • 运行Last Code - 执行最近生成的代码 • 生成代码——生成编译或运行代码 • Open仿真— 打开将生成 OpenModelica 或 Simulink 代码的目录 • 编辑模板——使用代码模板编辑器自定义为 OpenModelica 或 Simulink 生成的代码

仿真工件模型

字段	行动
工件	单击 ... 并选择现有的工件图标，然后创建一个新的工件。
包	如果您有一个现有的工件配置模型包，该字段指定了与该属性相关联的默认工件。 否则，单击 ... 图标并浏览并选择包含 SysML模型的包以进行仿真配置。您必须在选择包之前指定（或创建）工件。

包对象

本表讨论了将在配置SysML仿真窗口的 "名称"列下列出的 SysML模型中的object类型，这些对象将在模拟中进行处理。每个object类型展开以列出该类型的命名对象，以及需要在 "值"列中配置的每个object的属性。

很多级别的object类型、名称和属性都不需要配置，因此对应的 "Value"字段不接受输入。如果输入合适且被接受，则在字段的右端会显示一个下拉箭头；当您单击此选项时，将显示一个可能值的简短列表以供选择。某些值（例如部件的 "部件"）添加更多的参数层和属性，您可以单击 ... 按钮再次选择和设置参数的值。对于数据集，输入对话框允许您输入或导入值，例如初始值或默认值；请参阅使用数据集的模型分析主题。

元素类型	行为
值类型	ValueType 元素要么从原始类型泛化，要么被 SysMLSimReal 替代以进行仿真。
块	映射到 SysMLSimClass 或 SysMLSimModel 元素的块元素支持数据集的创建。如果您在上下文中定义了多个数据集（可以泛化），则必须将其中一个标识为默认值（使用时间菜单选项 "设置为默认数据集"）。 由于 SysMLSimModel 可能是模拟的顶级元素，并且不会被概括，如果您定义了多个数据集，则在模拟期间选择要使用的数据集。
属性	指定常量或变量及其设置的首选方法是在属性本身上使用 SysPhS 构造型 PhSConstant 和 PhSVariable。 PhSVariable 构造型具有内置属性*isContinuous*、 *isConserved*和*changeCycle* 。 该属性将列在 PhSConstant 或 PhSVariable 下，并且该值不能更改。 也可以在配置SysML仿真窗口中定义设置。在这种情况下，它们将列在 "属性"下。 块内的属性可以配置为 SimConstants 或 SimVariables。对于 SimVariable，您可以配置以下属性： • isContinuous — 确定属性值是连续变化（'true'，默认值）还是离散变化（' false '） • 属性— 确定属性的值是否保留（'true'）（' false '，默认值）；在为物理相互作用建模时，相互作用包括保守物理物质的交换，例如电流、力或流体流动 • changeCycle — 指定离散属性值更改的时间间隔；默认值为 "0" - changeCycle 只能设置为 0 以外的值 isContinuous = ' false ' - changeCycle 的值必须为正或等于 0
端口	无需配置。
模拟函数	函数被创建为块或约束块中的操作，原型为 "SimFunction"。 配置仿真窗口无需配置。
概括	无需配置。
捆绑连接器	将属性绑定到约束属性的参数。 无需配置；但是，如果属性不同，系统会提供同步它们的选项。
连接器	连接两个端口。 配置仿真视图中不需要配置。但是，您可能必须通过确定属性属性是否应该设置为 "False"（对于潜在的，以便建立相等耦合）或 "True"（对于流/保守的属性）来配置端口类型的属性, 从而建立和到零的耦合)。

约束块	无需配置。

仿真标签

此表描述了配置SysML仿真视图上的 “仿真”选项卡的字段。

字段	行动
模型	单击下拉箭头并选择模拟的顶级节点（SysMLSimModel元素）。该列表填充有定义为顶级模型节点的块的名称。
数据集	单击下拉箭头并选择所选模型的数据集。
预验证	选中此复选框可在执行模型的每次模拟之前自动验证模型。
开始	类型在开始模拟之前的初始等待时间，以秒为单位（默认值为 0）。
停止	类型模拟将执行的秒数。
格式	单击下拉箭头并选择 “plt”、 “csv”或 “mat”作为结果文件的格 ，其他工具可能会使用这些格式。
参数图	● 选中此复选框以在 y 轴上绘制图例A ，在 x 轴上图例B ● 取消选中复选框以在 y 轴上绘制图例，在 x 轴上绘制时间 注记：选中复选框后，您必须选择两个要绘制的属性。
使用Simscape	（如果选定的数学工具是 Simulink）如果您还想在 Simscape 中处理仿真，请选中该复选框。
依赖项	列出模拟该模型必须生成的类型。
属性的属性	提供与模拟有关的变量属性列表。选中要绘制的每个属性对应的复选框。

创建参数模型

在本主题中，我们将讨论如何开发用于仿真的 SysML模型元素（假设已有 SysML 建模知识），在配置SysML仿真窗口中配置这些元素，并观察在一些不同定义和建模方法下的仿真结果。本章提供的 SysML仿真示例中的图表和屏幕快照说明了这些要点。

创建参数模型时，您可以应用以下三种方法之一来定义约束方程：

- 在块元素上定义内联约束方程
- 创建可重用的约束块，以及
- 使用连接约束属性

您还将考虑：

- 物理交互中的流动
- 默认值和初始值
- 仿真函数
- 价值分配，以及
- 包导入

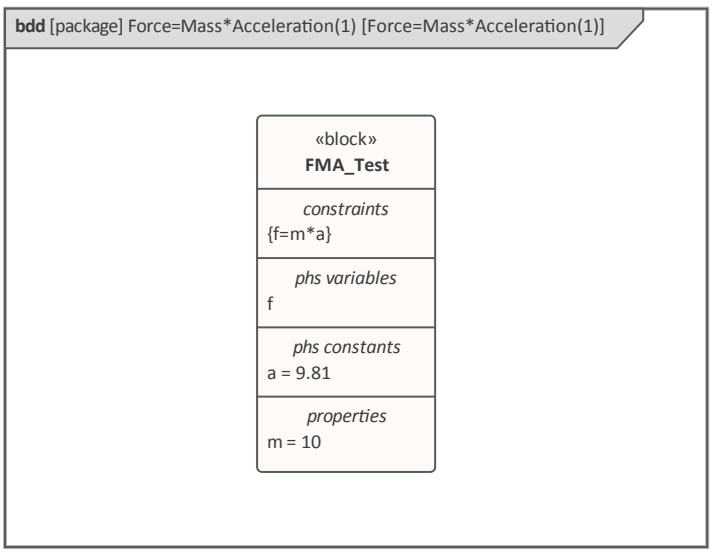
访问

功能区	仿真>系统行为> Modelica/Simulink >配置管理器

在块上定义内联约束方程

直接在块中定义约束很简单，也是定义约束方程的最简单方法。

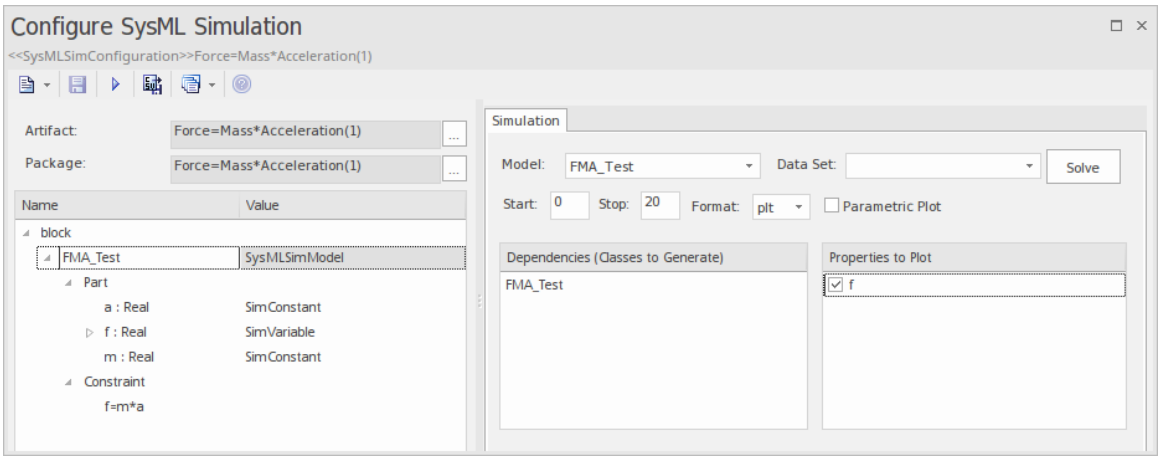
在这个图中，约束'f = m * a'被定义在一个块元素。



提示：您可以在一个块中定义多个约束。

1. 创建一个SysMLSim配置工具工件=Mass*Acceleration(1)'并将其指向包'FMA_Test'
2. 对于 F"MA_Test”，在 值″列中设置 S"ysMLSimModel″。

3. 对于 "a"、"m"和 "f"部分，在 "值"列中：将 "a"和 "m"设置为 "PhSConstant"，（可选）将 "f"设置为 "PhSVariable"。
4. 在 "仿真"选项卡的 "要绘制的属性"面板中，选中 "f"复选框。
5. 单击求解按钮运行模拟。

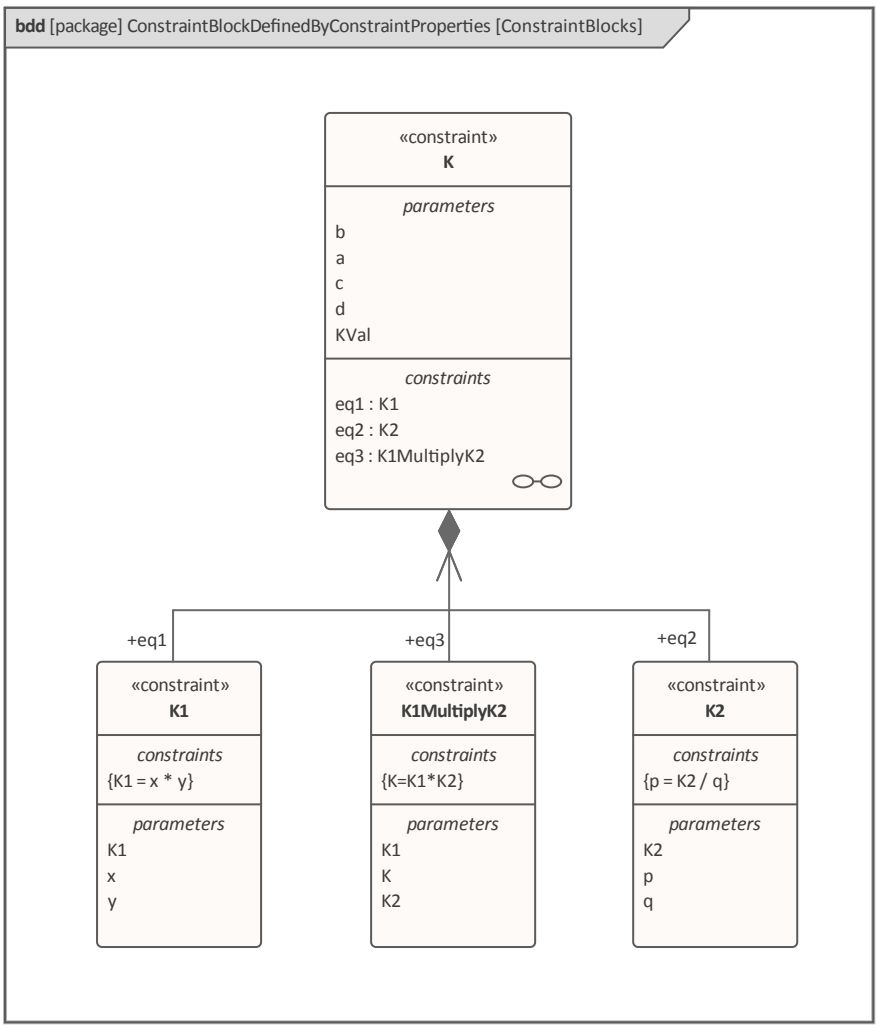


应A f = 98. 1 （来自 10 * 9.81）绘制图表。

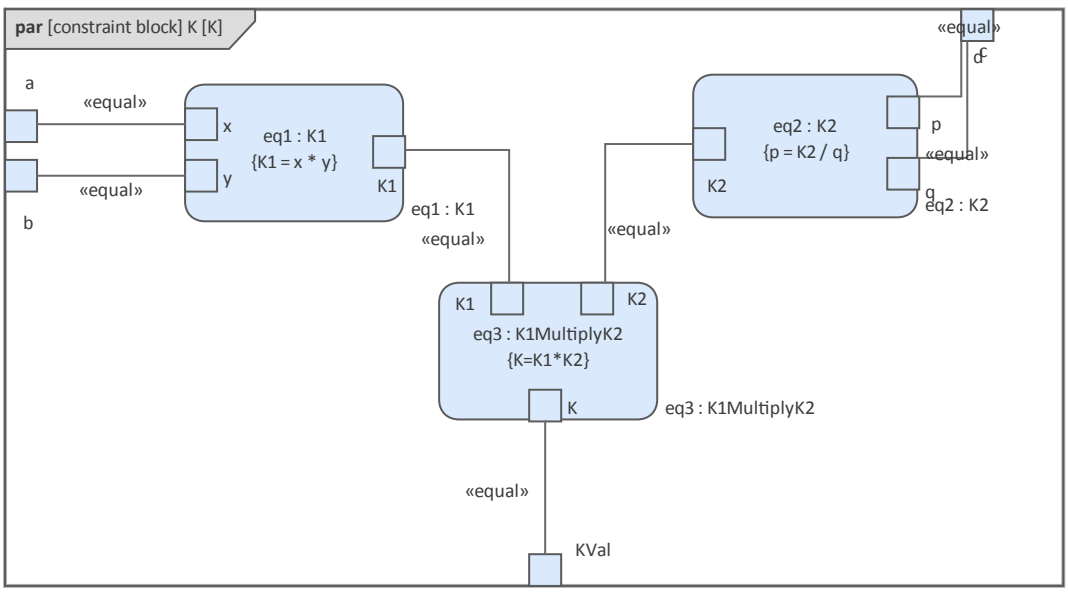
连接约束属性

在 SysML 中，ConstraintBlocks 中存在的约束属性可用于在定义约束时提供更大的灵活性。

在这个图中，ConstraintBlock ' K ' 定义了参数'a'、'b'、'c'、'd'和'属性'，以及三个约束'eq1'、'eq2'和'eq3'，类型为 'K1 '、'K2' 和 'K1MultiplyK2' 分别。

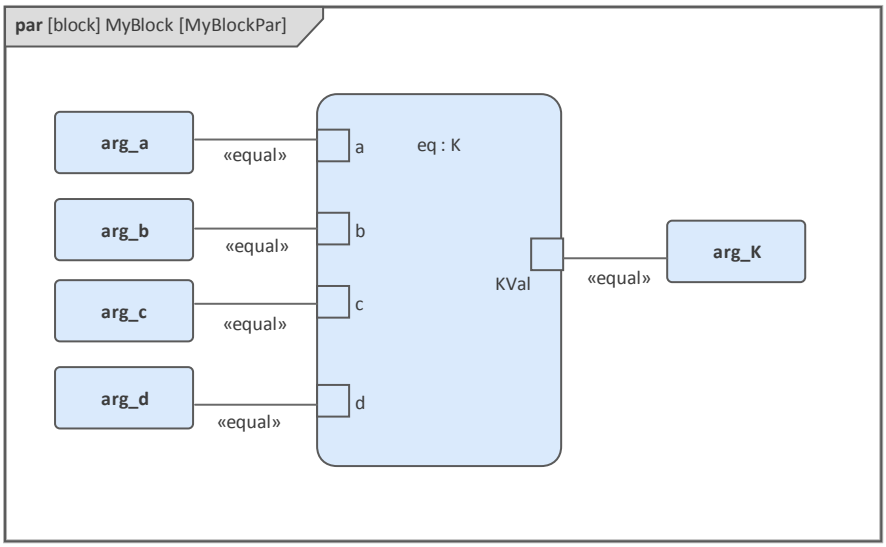


在 ConstraintBlock ' K ' 中创建一个参数图，并使用捆绑连接器将参数连接到约束属性，如图所示：

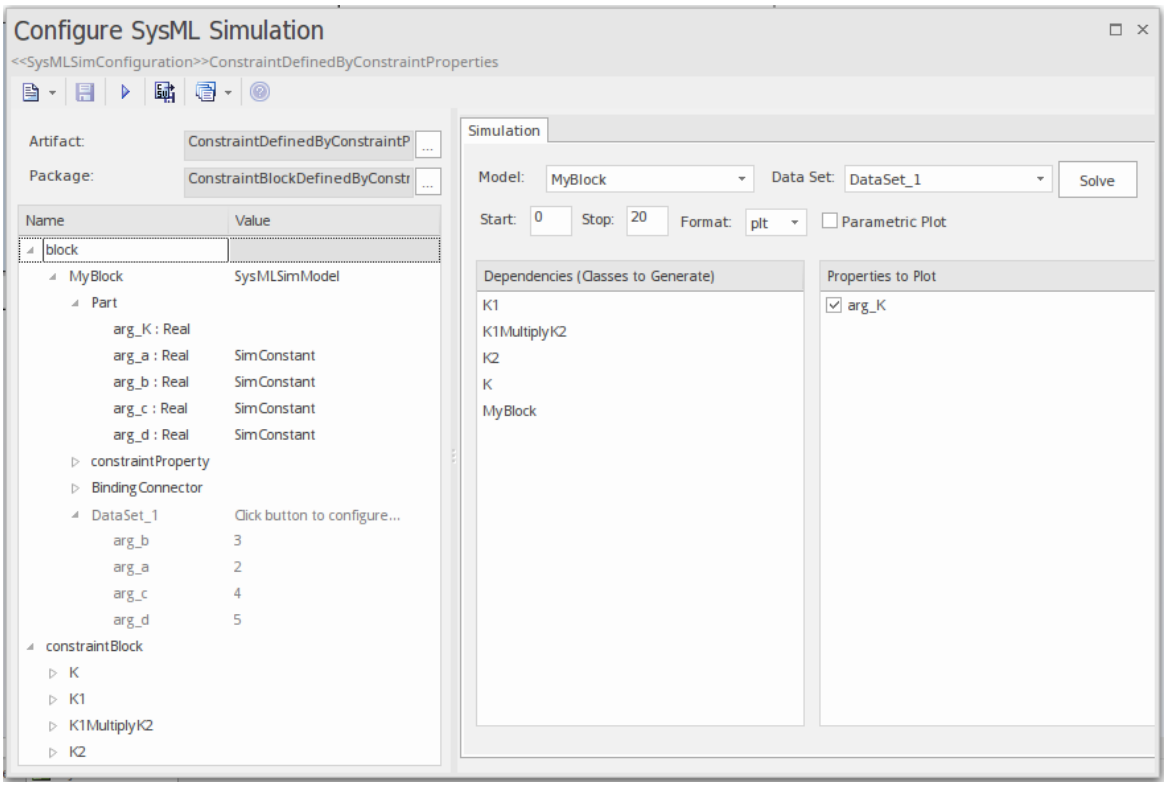


- 创建具有五个属性（零件）的模型属性
- 为属性创建一个约束属性 'eq' 并显示参数

- 将属性绑定到参数



- 在数据集中提供值 (arg_a = 2, arg_b = 3, arg_c = 4, arg_d = 5)
- 在 "配置SysML仿真"对话框中，将 "模型"设置为 "MyBlock"，将 "数据集"设置为 "DataSet_1"
- 在 "propertyto Plot"面板中，选中 "属性"复选框
- 点击 Solve 按钮运行模拟



将计算并绘制结果 120（计算为 2 * 3 * 4 * 5）。这与我们用笔和纸进行扩展时相同： K = K1 * K2 = (x*y) * (p*q)，然后绑定值 (2 * 3) * (4 * 5)；我们得到 120。

有趣的是，我们有意将 K2 的方程定义为 "p = K2 / q"，这个例子仍然有效。

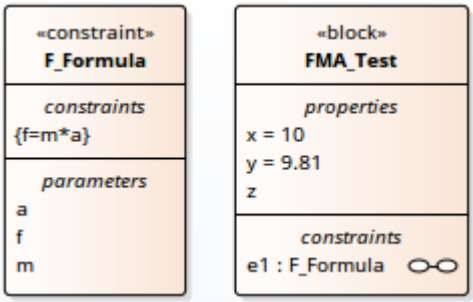
在这个例子中，我们可以很容易地将 K2 求解为 p * q，但在一些复杂的例子中，从方程中求解变量是非常困难的；但是， Enterprise Architect SysMLSim 仍然可以做到这一点。

总之，该示例向您展示了如何通过构造约束属性来定义具有更大灵活性的 ConstraintBlock。虽然我们只在 ConstraintBlock 中演示了一层，但这种机制将适用于任意级别的复杂模型。

创建可重用的约束块

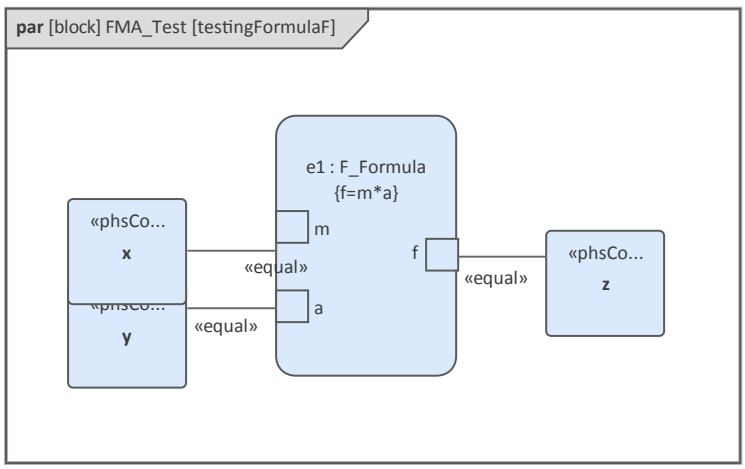
如果一个方程在许多块中通用，则可以创建一个 ConstraintBlock 用作每个块中的约束属性。这些是我们根据前面的示例所做的更改：

- 创建一个 ConstraintBlock元素'F_Formula' 具有三个参数 'a'、'm' 和 'f'，以及一个约束 'f = m * a'

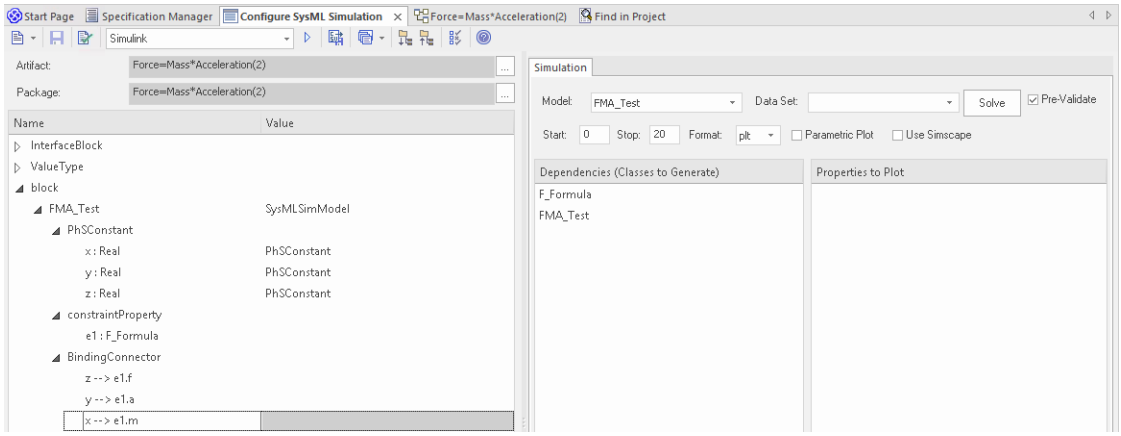


提示：如果属性类型为空，将应用原始类型 *'Real'*

- 用'x'、'y'和'z'三个属性创建一个块'属性'，并分别赋予'x'和'y'默认值'10'和'9.81'
- 在 "FMA_Test" 中创建一个参数图，显示属性 "x"、"y" 和 "z"
- 创建一个 ConstraintProperty 'e1' 类型为 'F_Formula' 并显示参数
- 在'x-m'、'y-a'和'f-z'之间划捆绑连接器，如图所示：



- 在元素中创建一个工件并对其进行配置，如图所示：
  - 在 "值" 列中，将 "FMA_Test" 设置为 "SysMLSimModel"
  - 在 "值" 列中，将 "x" 和 "y" 设置为 "PhSConstant"
  - 在 "要绘制的属性" 面板中，选中 "Z" 复选框
  - 单击求解按钮运行模拟



应A f = 98. 1 （来自 10 * 9.81）绘制图表。

物理交互中的流动

在对物理相互作用进行建模时，应将电流、力、扭矩和流速等守恒物理物质的交换建模为流，并且应将流变量设置为属性 "isConserved"。

连接建立两种不同类型的耦合，取决于流属性是潜在的（默认）还是流（守恒）：

- 等式耦合，用于潜在（也称为努力）属性
- 和零耦合，用于流（守恒）属性；例如，根据电学领域的基尔霍夫电流定律，电荷守恒使所有电荷流入一个和为零的点

在 "ElectricalCircuit"示例的生成 OpenModelica 代码中：

连接器充电端口

流动电流 i； // 如果 'isConserved' = true 将生成流关键字

电压v；

结束充电端口；

电路模型

源；

电阻电阻器；

接地；

方程

连接（源.p，电阻器.n）；

连接（接地.p，源.n）；

连接（电阻器.p，源.n）；

结束电路；

每个连接方程实际上扩展为两个方程（ChargePort 中定义了两个属性），一个用于等式耦合，另一个用于和零耦合：

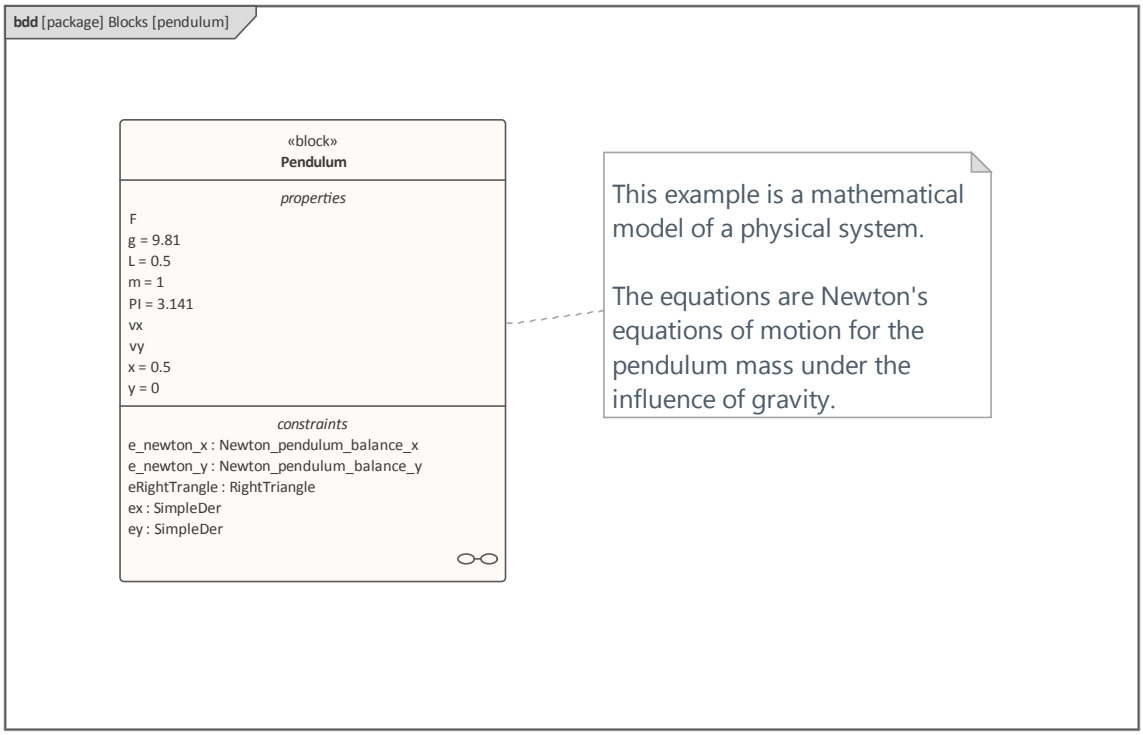
源.pv = 电阻器.nv;

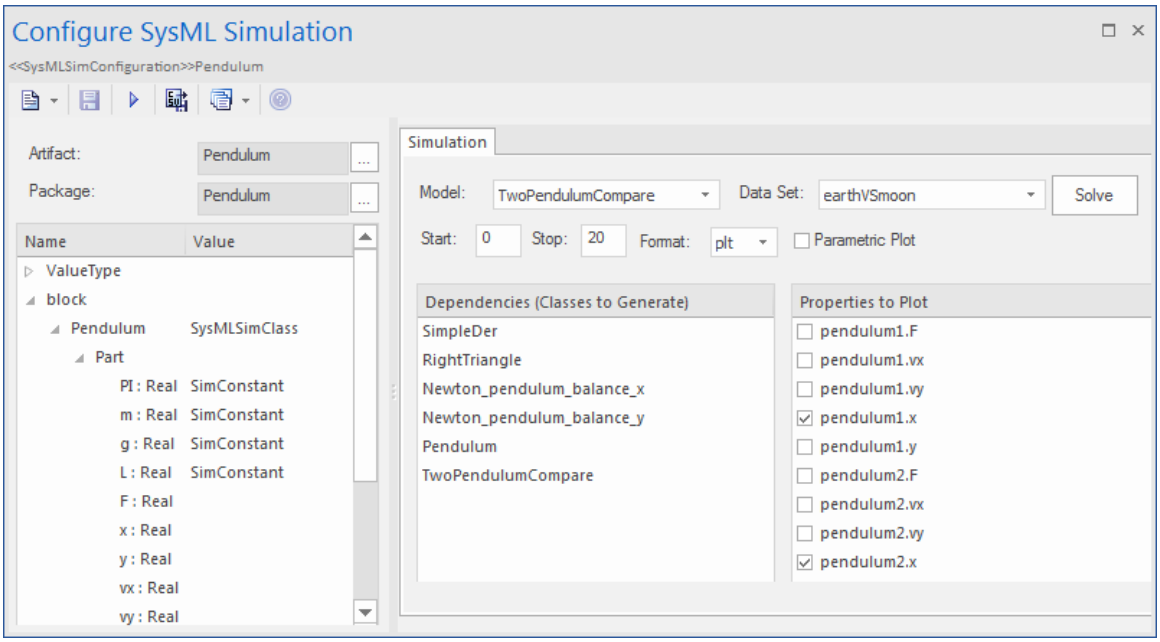
源.pi + 电阻.ni = 0;

默认值和初始值

如果在 SysML属性元素（ 属性”对话框 > 属性”页面 > 初始”字段）中定义了初始值，则它们可以作为 PhSConstant 的默认值或 PhSVariable 的初始值加载。

在这个 Pendulum 示例中，我们为属性 g”、 L ”、 m”、 PI”、 x”和 y”提供了初始值，如图左侧所示。由于 “PI”（数学常数）、 m”（摆锤的质量）、 g”（重力因子）和 L ”（摆锤的长度）在模拟过程中不会发生变化，因此将它们设置为 PhSConstant”。





生成的 Modelica 代码如下所示：

类摆
参数 Real PI = **3.141**；
参数 Real m = **1** ;
参数 Real g = **9.81** ;
参数 Real L = **0.5** ;
实F；
实数 x **(start=0.5)** ;
实 y **(start=0)** ;
真正的vx；
真正的 vy;
……
方程
……
结束摆；

- 属性'PI'、'm'、'g'和' L '是常数，并作为声明方程生成
- 属性 "x" 和 "y" 是可变的；它们的起始值分别为 0.5 和 0，初始值作为修改生成

仿真函数

仿真函数是编写复杂逻辑A有用工具，并且易于用于约束。本节介绍 TankPI 示例中的一个函数。
在 ConstraintBlock 'Q_OutFlow' 中，函数'LimitValue' 被定义并在约束中使用。

- 在块或约束块上，创建一个操作（本例中为 "LimitValue"）并打开特征窗口的 "操作"选项卡

- 给操作定型 "SimFunction"
- 定义参数并将方向设置为 "输入/输出"

提示：多个参数可以定义为*'out'*，调用者取值的格式为：

*(out1, out2, out3) = function_name(in1, in2, in3, in4, ...); //*方程形式

*(out1, out2, out3) := function_name(in1, in2, in3, in4, ...); //*报表形式

- 在属性窗口的 "代码"选项卡的文本字段中定义函数体，如图：

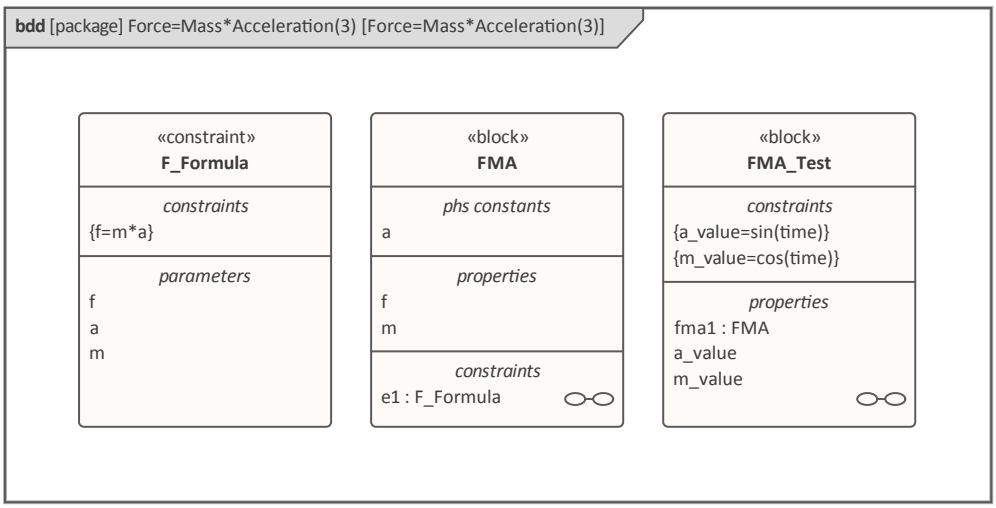
```
pLim :=
如果 p > pMax 那么
最大
else如果 p < pMin 那么
pMin
else
p;
```

生成代码时， Enterprise Architect将收集所有在 ConstraintBlocks 和 Blocks 中定义为 "SimFunction"的操作，然后生成类似于以下内容的代码：

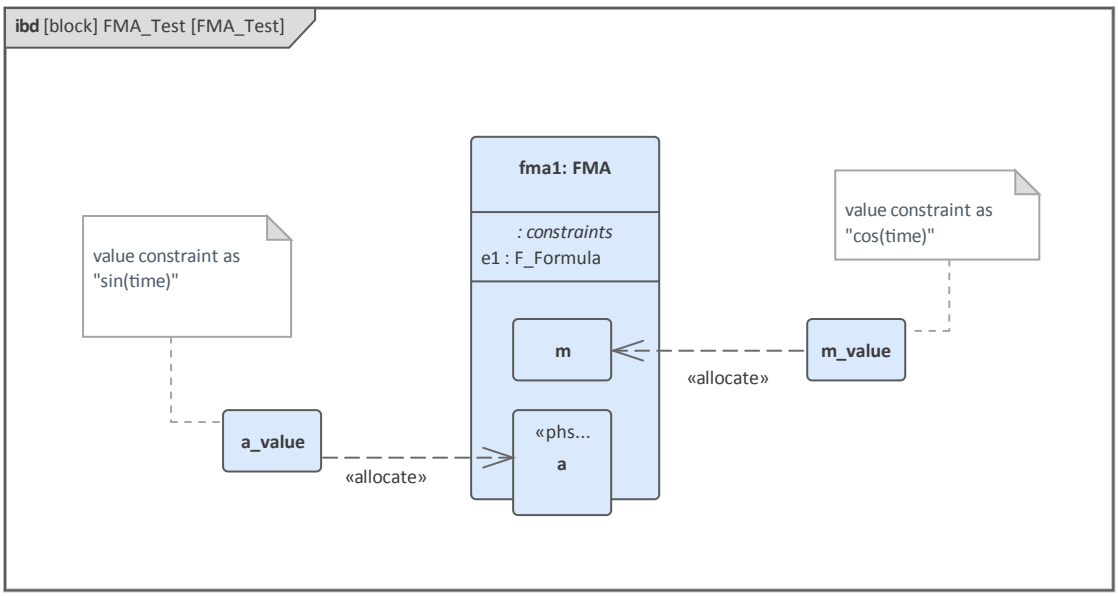
```
函数极限值
输入Real pMin；
输入真实 pMax；
输入实数 p;
输出实 pLim；
算法
pLim :=
如果 p > pMax 那么
最大
else如果 p < pMin 那么
pMin
else
p;
结束极限值；
```

价值分配

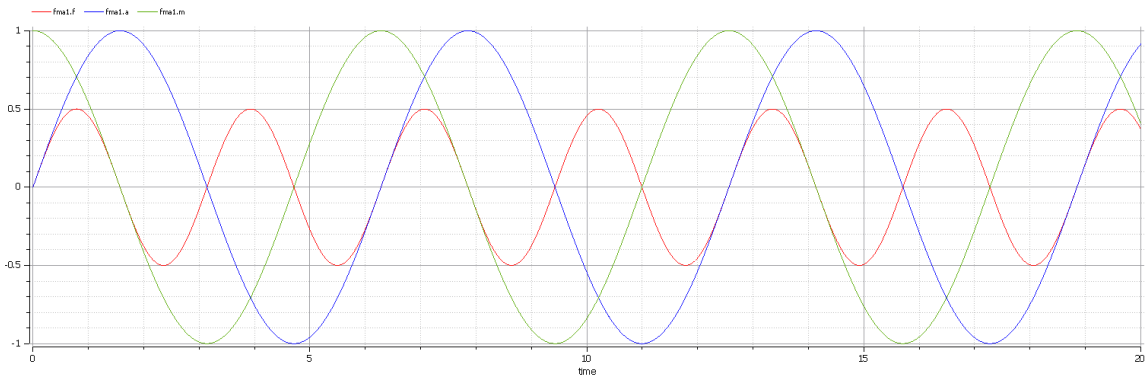
该图显示了一个名为 "力=质量*加速度"的简单模型。



- 块 ‘FMA’ A属性 ‘a’、 ‘f’和 ‘m’以及约束属性 ‘e1’建模，输入约束块 ‘F_Formula’
- 块'FMA'的属性没有设置任何初始值，并且属性'a'、'f'和'm'都是可变的，所以它们的值变化取决于它们被模拟的环境
- 创建一个块'FMA_Test'作为SysMLSimModel并添加属性'fma1'来测试块'FMA'的行为
- 约束'a_value' 为' sin (time)'
- 约束'm_value'为' cos (time)'
- 划分配连接器将值从环境分配到模型'FMA'



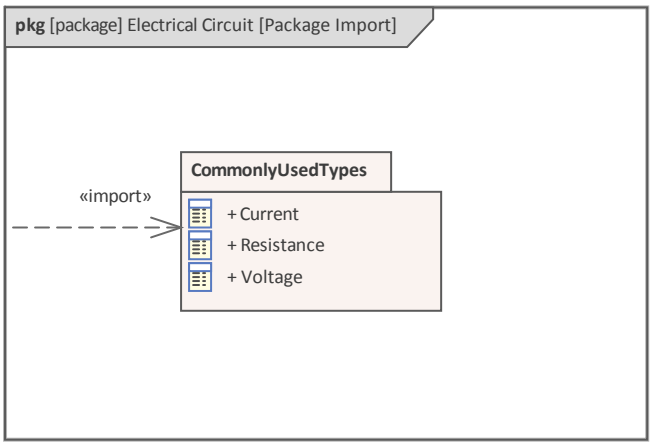
- 选中针对 ‘fma1.a’、 ‘fma1.m’和 ‘fma1.f’的 要绘制的属性”复选框
- 单击求解按钮以模拟模型



包导入

工件选择一个包的元素（如Blocks、ConstrainBlocks和Value Types）如果模拟依赖于不属于这个包的元素，比如Reusable库， Enterprise Architect在包元素之间提供了一个导入连接器来满足这个需求。

在电气电路示例中，工件被配置为 ‟ElectricalCircuit″包，其中包含模拟所需的几乎所有元素但是，一些属性被键入为值类型，例如 ‟电压″、 ‟电流″和 ‟电阻″，这些值类型在多个 SysML 模型中常用，因此放置在单个 SysML 模型之外的一个名为 ‟CommonlyUsedTypes″的包中。如果使用导入连接器导入此包，则导入包中的所有元素都将出现在导入配置管理器中。



使用数据集进行模型分析

参数模型中使用的每个 SysML块都可以在仿真配置中定义多个数据集。这允许使用相同的 SysML模型进行可重复的模拟变化。

块A键入为 SysMLSimModel（不能被概括或构成组合的一部分的顶级节点）或 SysMLSimClass（可以被概括或构成组合的一部分的较低级别的元素）。在 SysMLSimModel元素上运行模拟时，如果您定义了多个数据集，则可以指定要使用的数据集。但是，如果模拟中的类有多个数据集，则您无法选择在模拟期间使用哪一个，因此必须将一个数据集标识为该类的默认值。

访问

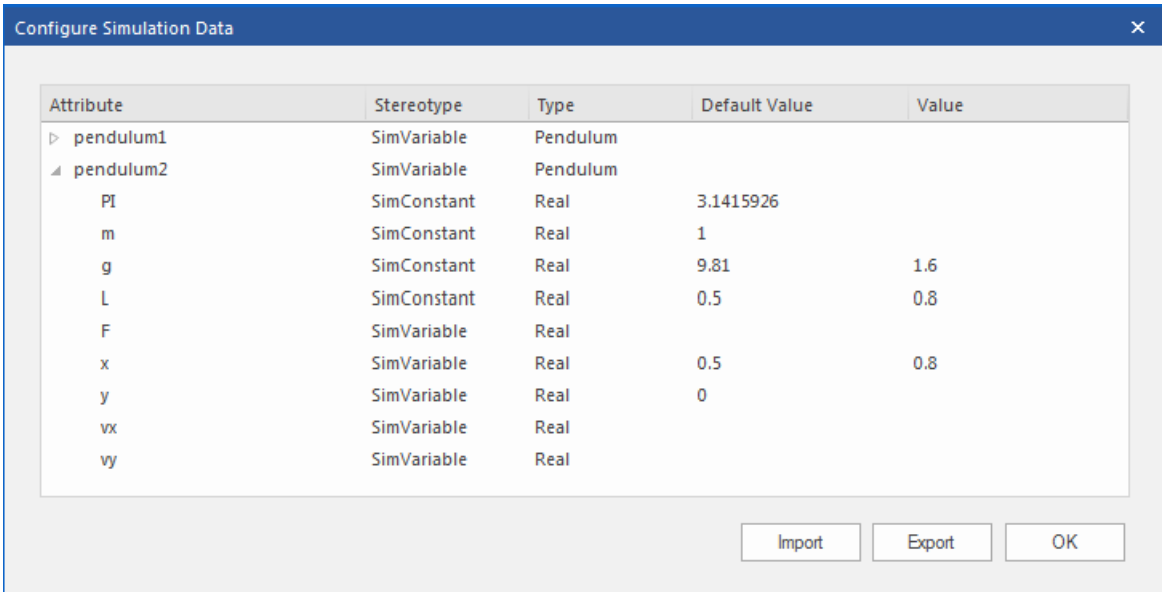
功能区	仿真>系统行为>Modelica/Simulink>配置管理器>在 块"组>名称列>块元素上下文菜单>创建仿真数据集

数据集管理

任务	行动
创造	要创建新数据集，请右键单击块名称并选择 创建仿真数据集"选项。数据集被添加到块名称下方的组件列表的末尾。单击 ... 按钮在 配置仿真数据"对话框中设置数据集（参见配置仿真数据表）。
复制	要复制现有数据集作为创建新数据集的基础，请右键单击数据集名称并选择 复制"选项。重复的数据集将添加到块名称下方的组件列表的末尾。单击 ... 按钮以编辑 配置仿真数据"对话框中的数据集（请参阅配置仿真数据表）。
删除	要删除不再需要的数据集，请右键单击数据集并选择 删除数据集"选项。
默认设置	要设置 SysMLSimClass 在用作属性类型或继承（以及当有多个数据集时）使用的默认数据集，请右键单击数据集并选择 设置为默认值"选项。默认数据集的名称以粗体突出显示。模型使用的属性将使用此默认配置，除非模型明确覆盖它们。

配置仿真数据

此对话框主要用于提供信息。您可以直接添加或更改数据的唯一列是 值"列。



柱子	描述
属性	属性”列提供了正在编辑的块中所有属性的树视图。
构造型	对于每个属性， 构造型”列标识它是否已在模拟或变量期间配置为常量，以便预期值会随时间变化。
类型	类型”列描述了用 模拟此属性的类型。它可以是原始类型（例如 Real”）或对模型中包含的块的引用。属性引用块将显示由它们下面的引用块指定的子属性。
默认值	如果未提供覆盖， 默认值”列显示将在模拟中使用的值。这可以来自SysML模型中的 初始值”字段或来自父类型的默认数据集。
价值	值”列允许您覆盖每个原始值的默认值。
导出/导入	单击这些按钮可使用电子表格等外部应用程序修改当前数据集中的值，然后将它们重新导入列表。

SysML仿真实例

本节为每个阶段提供了一个工作示例：为域创建 SysML模型，对其进行仿真，并评估仿真结果。这些示例应用了前面主题中讨论的信息。

例子

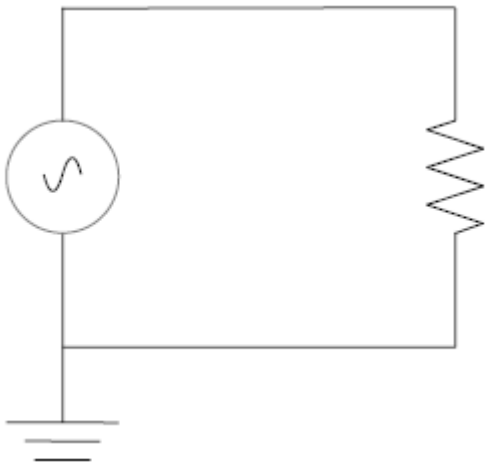
模型	描述
电路仿真示例	第一个例子是加载电路的模拟。该示例从电路图开始，并将其转换为参数模型。然后对模型进行仿真，评估电阻器源端和目标端的电压并将其与预期值进行比较。
质量弹簧阻尼振荡器仿真示例	第二个例子使用一个简单的物理模型来演示质量-弹簧-阻尼系统的振荡行为。
水箱压力调节器	最后一个示例显示了两个水箱的水位，其中水在它们之间分配。我们首先模拟一个平衡良好的系统，然后我们模拟一个水将从第二个水箱溢出的系统。

电路仿真示例

在本例中，我们为一个简单的电路创建 SysML 参数模型，然后使用参数仿真来预测和绘制该电路的行为。

图表

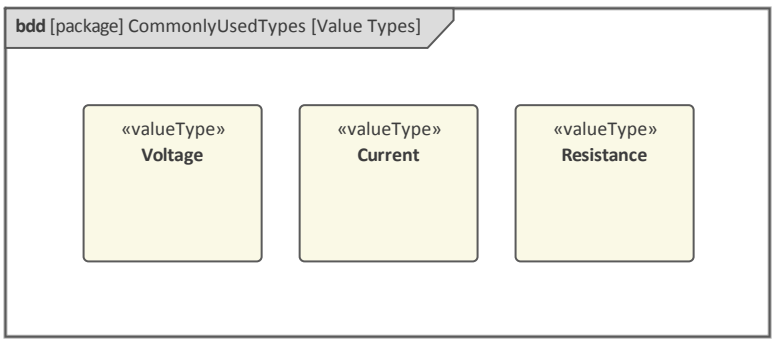
我们将要学习的电路模型（此处显示）使用标准电路符号。

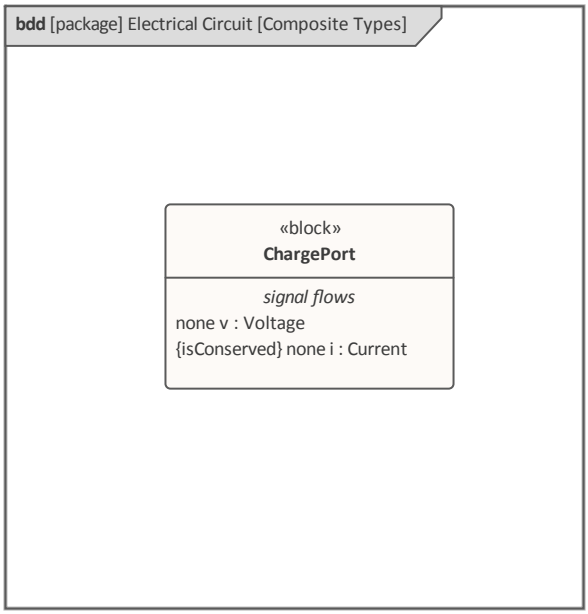


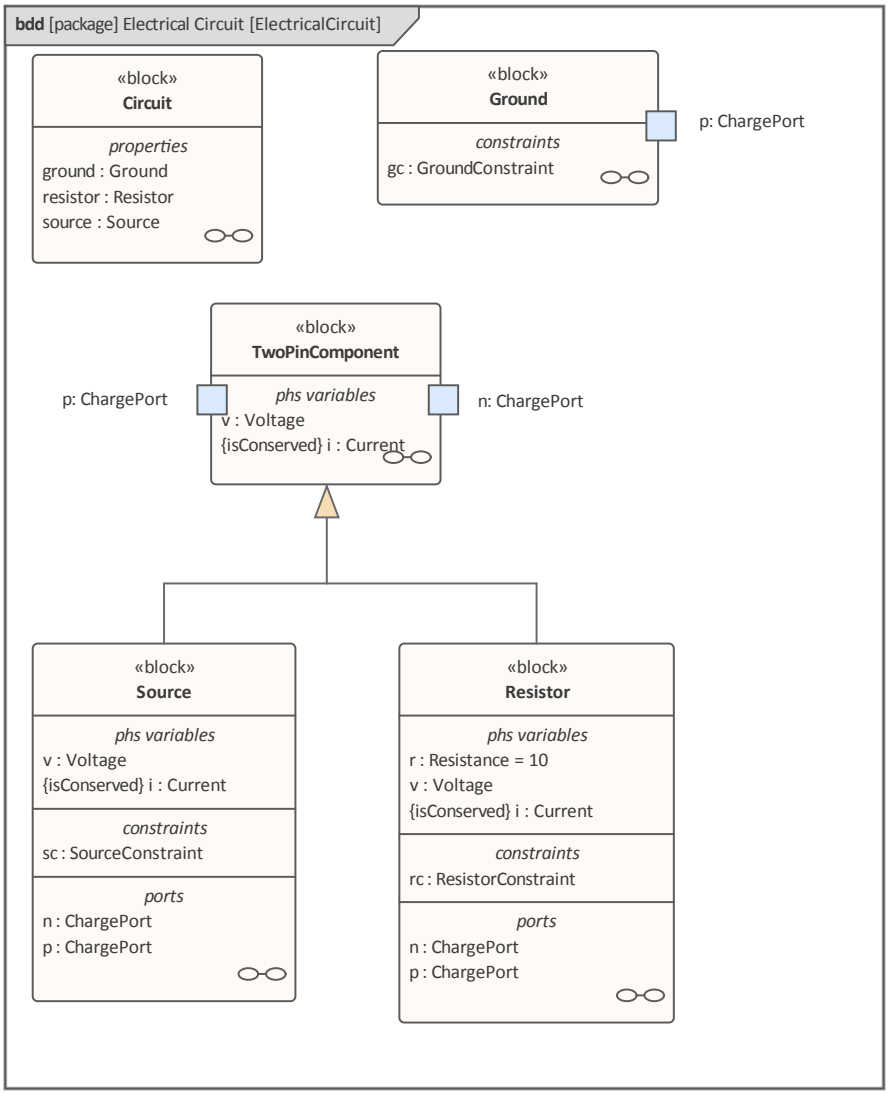
该电路包括交流电源、源和电阻器，通过电线相互连接。

创建 SysML模型

这张表显示了我们如何构建一个完整的完成模型来表示电路，从最低级别的类型开始，一步一步地构建模型。

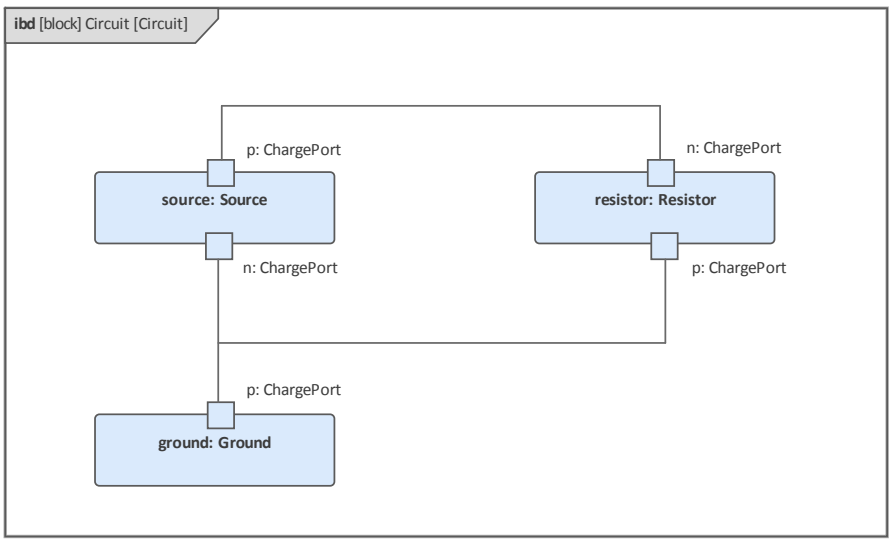
部件	行动
类型	定义电压、电流和电阻的值类型。单位和数量种类对于模拟而言并不重要，但如果定义一个完整的完成模型则需要设置。这些类型将从原始类型 “Real”泛化。在其他模型中，您可以选择将值类型映射到与模型分开的相应仿真类型。  此外，定义一个名为 ChargePort 的复合类型 (块)，其中包括电流和电压属性。这种类型允许我们表示组件之间连接器处的电能。

	
块	在 SysML 中，电路和每个组件都将表示为块。 在块定义图 (图表) 中，创建一个电路块。该电路由三部分组成：源、地和电阻。这些部分属于不同的类型，具有不同的行为。 为每个零件类型创建一个块。电路块的三个部分通过端口连接，代表电气引脚。源和电阻有一个正极和一个负极引脚。地只有一个引脚，它是正极。电流（电荷）通过引脚传输。创建一个带有两个端口（引脚）的抽象块 “TwoPinComponent″。这两个端口分别命名为 ṕ”（正）和 ń”（负），它们的类型为 ChargePort. 此图显示了 BDD，包括电路块、接地、TwoPinComponent、源和电阻器。

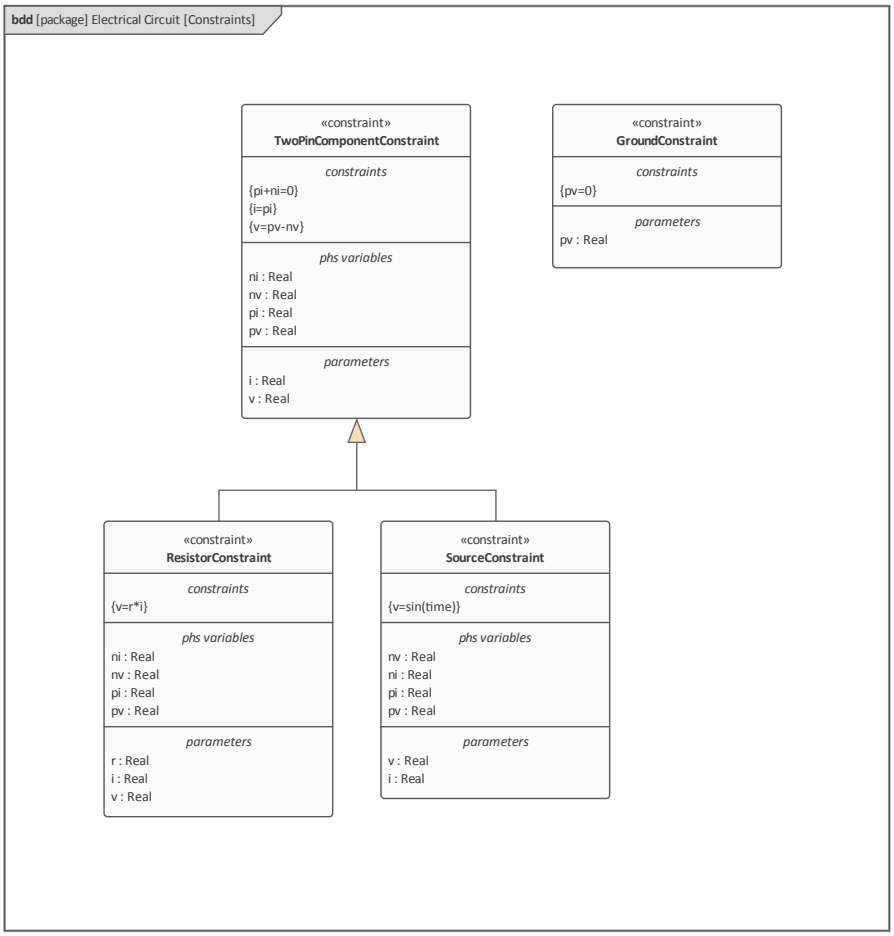


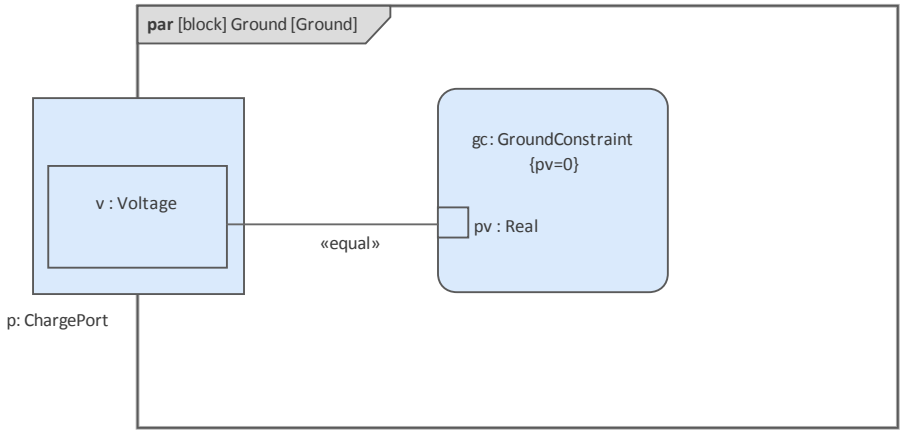
内部结构

为电路创建一个内部块图（图表）。为源、电阻和接地添加属性，由相应的块键入。用连接器连接端口。源的正极引脚连接到电阻器的负极引脚。Resistor 的正极引脚连接到源的负极引脚。地也连接到源的负极引脚。



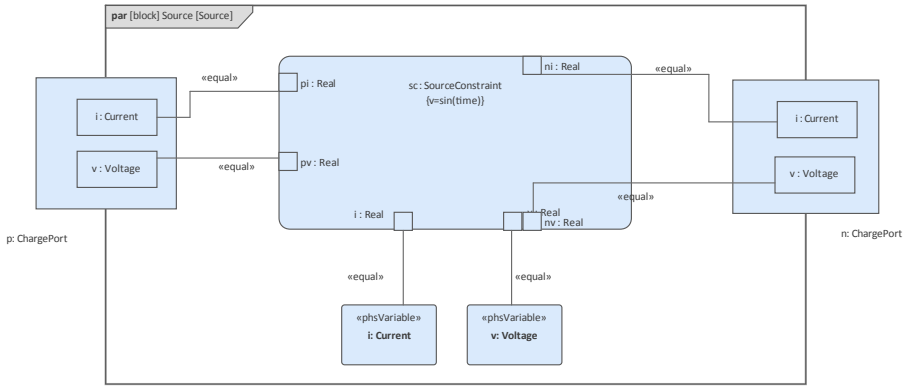
请注意，这遵循与原始电路图相同的结构，但每个组件的符号已替换为由我

	们定义的块键入的属性。
约束	方程定义数值属性之间的数学关系。在 SysML 中，方程表示为 ConstraintBlocks 中的约束。 ConstraintBlocks 的参数对应于块的 PhSVariables 和 PhSConstants（本例中为'i'、'v'、'r'），以及端口类型中存在的端口（'pv'、'pi'、'nv '，在本例中为 'ni'）。 创建一个 ConstraintBlock 'TwoPinComponentConstraint' 来定义源和电阻器共有的参数和方程。该等式应状态组件的电压等于正负引脚电压之差。组件的电流等于通过正极引脚的电流。通过两个引脚的电流之和必须加起来为零（一个是另一个的负数）。接地约束表明接地引脚上的电压为零。源约束将电压定义为sine ，以当前仿真时间为参数。此图显示了这些约束如何在 BDD 中呈现。 
绑定	约束参数的值等同于具有绑定连接器的可变值和常数值。在每个块上创建约束属性（属性类型为 ConstraintBlocks），并将块变量和常量绑定到参数，以将约束约束块。这些图分别显示了接地、源和电阻的绑定。 对于地面约束，将 gc.pv 绑定到 pv



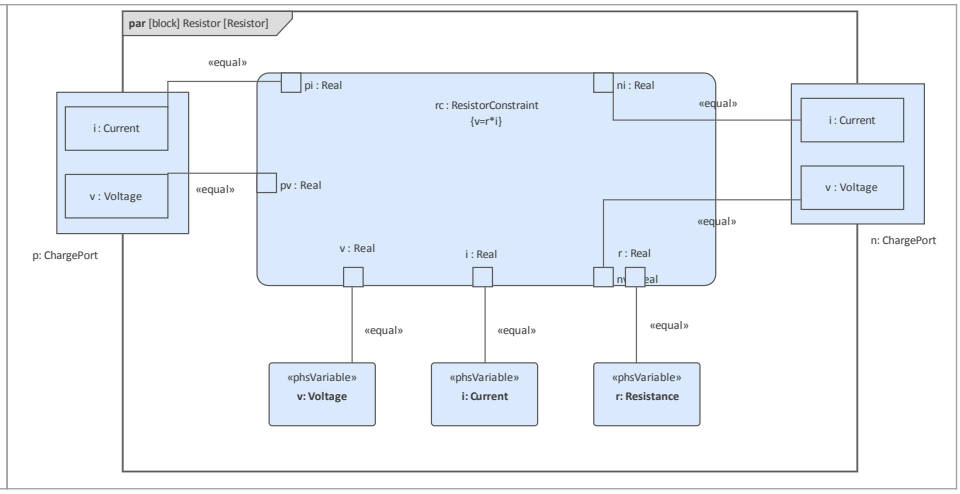
对于源约束，绑定：

- sc.pi 到 pi
- sc.pv 到 pv
- sc.v 到 v
- sc.i到我
- sc.ni 到 ni 和
- sc.nv 到 nv



对于电阻器约束，绑定：

- rc.pi 到 pi
- rc.pv 到 pv
- rc.v 到 v
- rc.i 到我
- rc.ni 到 ni
- rc.nv 到 nv 和
- rc.r 到 r



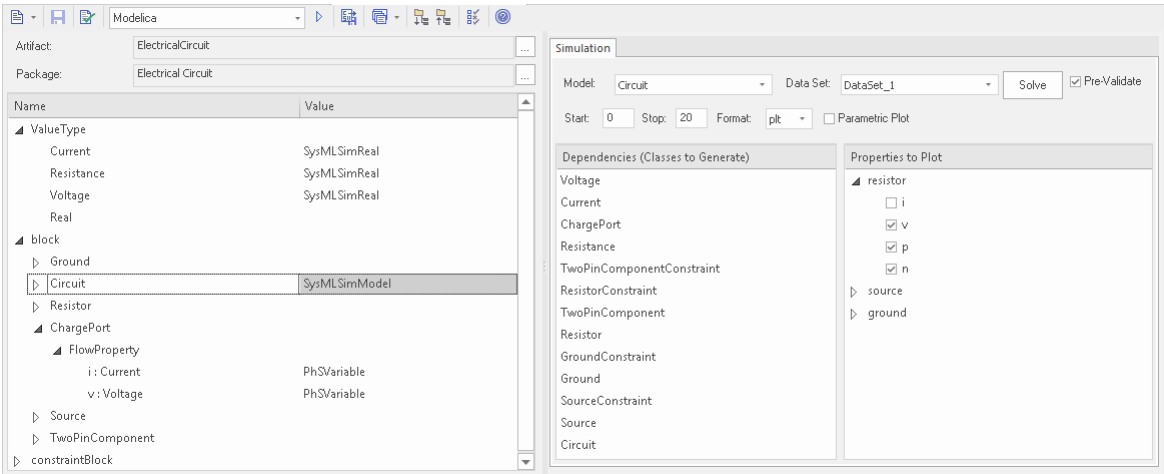
配置仿真行为

此表显示了 SysMLSim 配置的详细步骤。

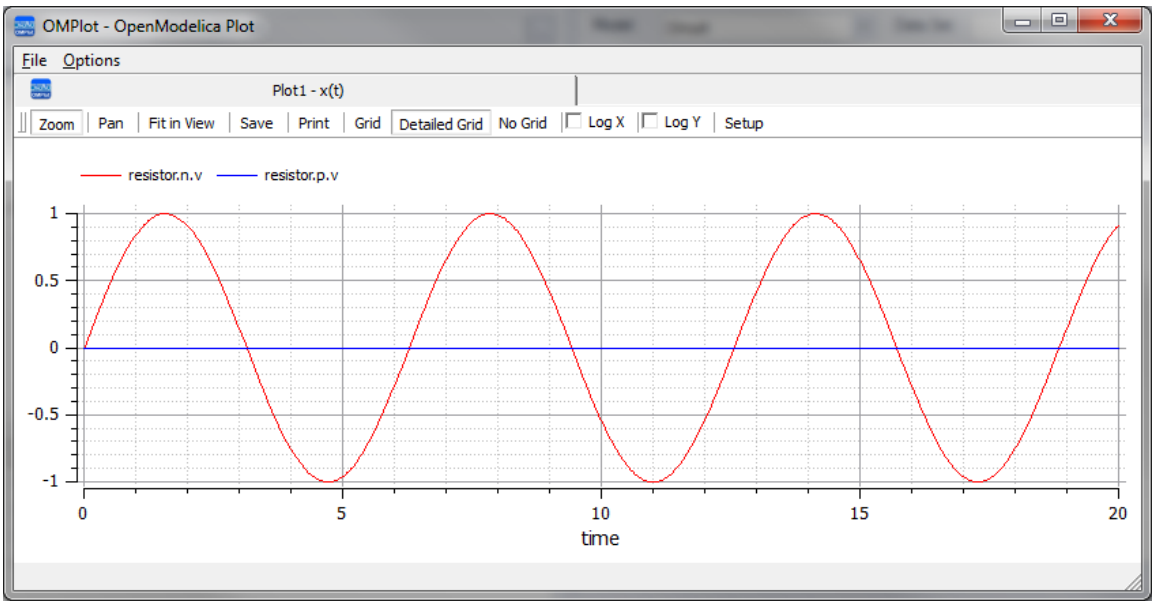
节	行动
工件适合	• 选择 "仿真>系统行为> Modelica/Simulink >配置管理器" • 从第一个下拉菜单中，选择工具栏创建工具工件元素创建工件 • 选择拥有此 SysML模型的包
在配置管理器中创建根元素	• 值类型 • 块 • 约束块
值类型替换	展开 ValueType 并为 Current、Resistance 和 Voltage 中的每一个从 "Value"组合框中选择 "SysMLSimReal"。
将属性设置为流	• 将 "块"扩展到 ChargePort |流属性	i ：当前并从 "值"组合框中选择 "SimVariable" • 对于 "元素"，单击 ... 按钮以打开 "ElementConfigurations"对话框 • 将 "isConserved"设置为 "True"
SysMLSimModel	这就是我们要模拟的模型：将块'Circuit'设置为'SysMLSimModel'。

运行仿真

在 "仿真"页面中，选中 "resistor.n"和 "resistor.p"复选框进行绘图，然后单击 "求解"按钮。



如图所示，绘制了两个图例 “resistor.n”和 “resistor.p”。

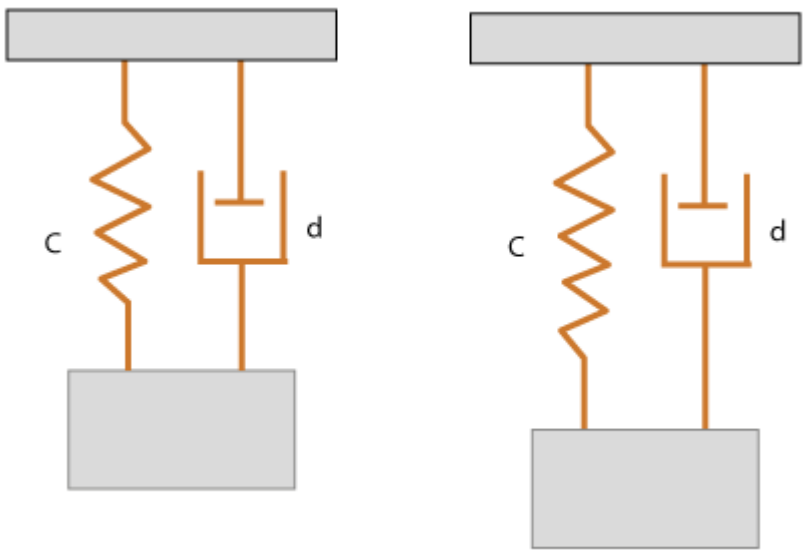


质量弹簧阻尼振荡器仿真示例

在本节中，我们将为由质量、弹簧和阻尼器组成的简单振荡器创建 SysML 参数模型，然后使用参数模拟来预测和绘制该机械系统的行为。最后，我们通过比较通过数据集提供不同参数值的两个振荡器来执行假设分析。

正在建模的系统

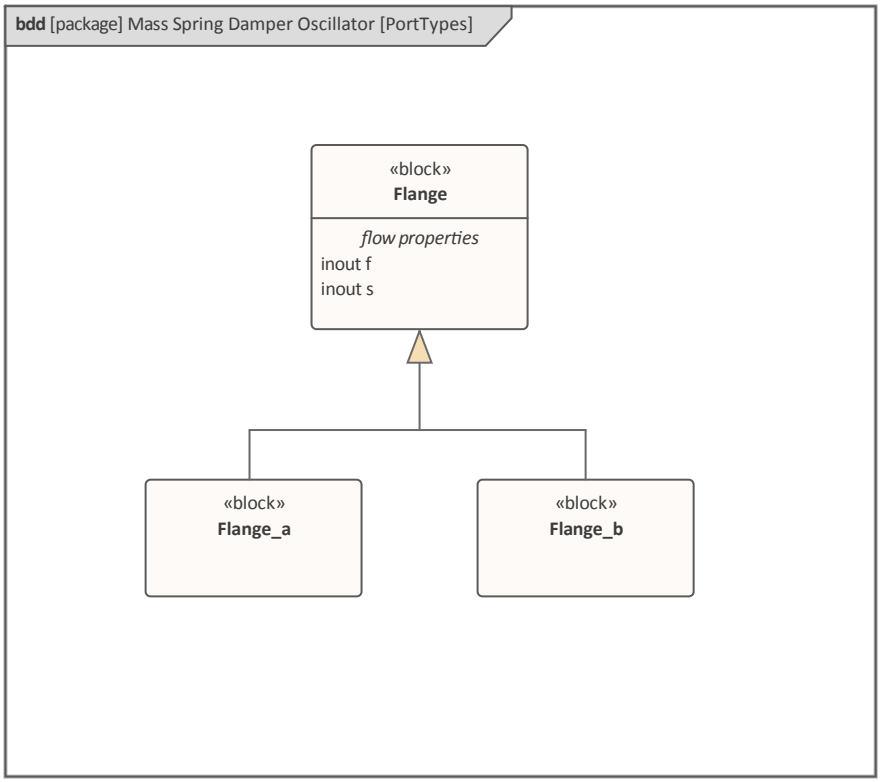
质量悬挂A弹簧和阻尼器上。此处显示的第一个状态表示时间=0 时的初始点，就在质量释放时。第二个状态代表当身体静止并且弹簧力与重力平衡时的最终点。

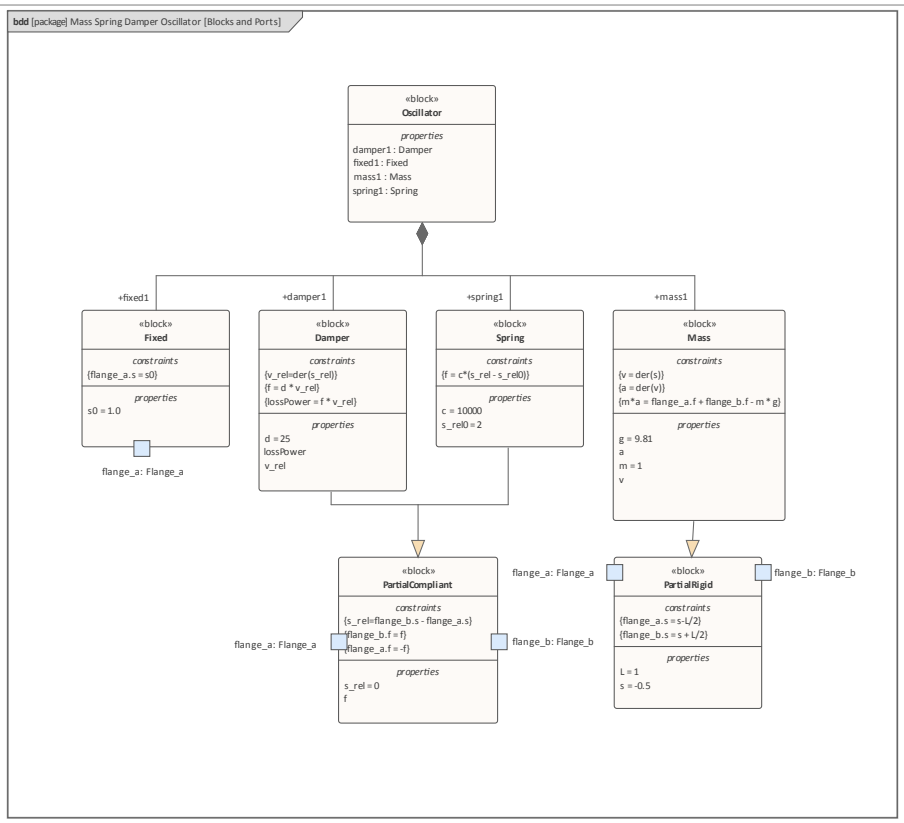


创建 SysML模型

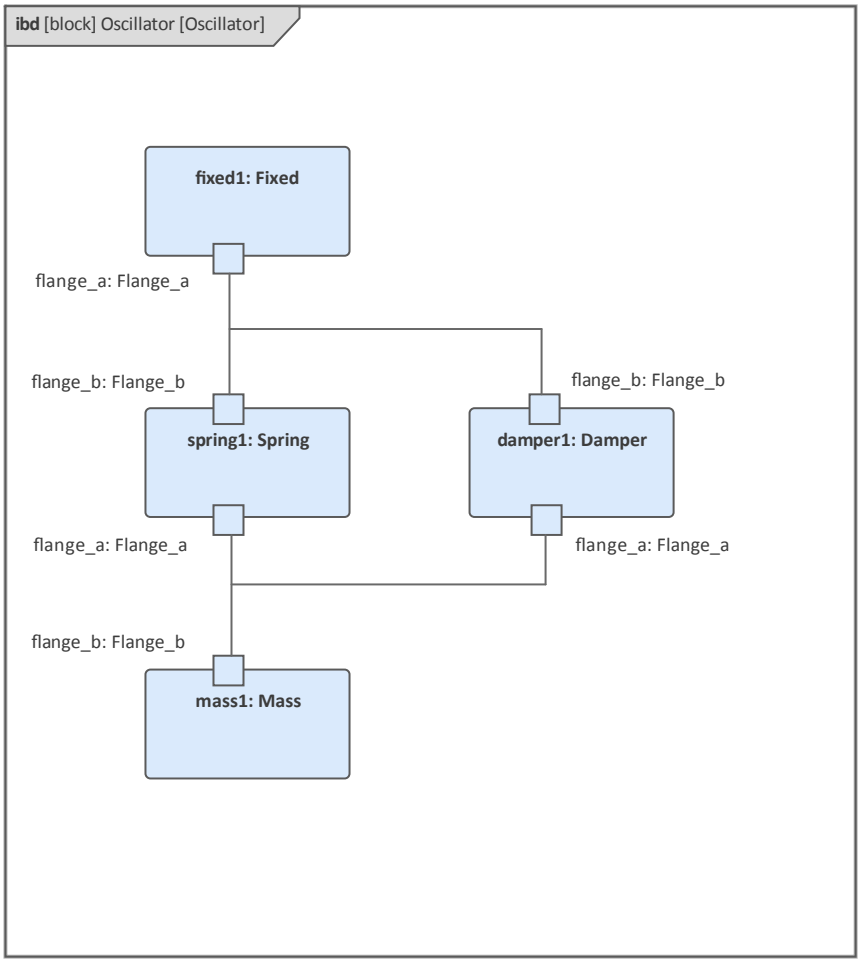
SysML 中的块模型有一个主要模块，即振荡器。振荡器有四个部分：固定顶板、弹簧、阻尼器和质量体。为这些部分中的每一个创建一个块。 Oscillator块的四个部分通过端口连接，代表机械法兰。

成分	描述
端口类型	用于 1D 过渡机械域中法兰的块 'Flange_a' 和 'Flange_b' 相同，但作用略有不同，有点类似于电气域中 PositivePin 和 NegativePin 的作用。力通过法兰传递。所以 flow 属性*Flange.f*的属性*isConserved*应该设置为True 。

	
块和端口	• 创建块 弹簧″、 阻尼器″、 质量″和 固定″以分别表示弹簧、阻尼器、质量体和天花板 • 创建一个块'PartialCompliant'，有两个端口（法兰），命名为'flange_a'和'flange_b'——它们分别是Flange_a和Flange_b类型； 'Spring' 和 'Damper' 块从 'PartialCompliant' 概括 • 创建一个具有两个端口（法兰）的块 “PartialRigid”，命名为 “flange_a″和 “flange_b″——它们分别是 Flange_a 和 Flange_b 类型； 'Mass'块从 'PartialRigid'概括 • 为天花板创建一个只有一个法兰的块'Fixed'，它的端口只有'flange_a'类型为 Flange_a



内部结构	为 “振荡器”创建内部块图 (IBD)。为固定天花板、弹簧、阻尼器和质量体添加属性，由相应的 Blocks 键入。用连接器连接端口。 • 将 “fixed1”的 “flange_a”连接到 “spring1”的 “flange_b” • 将 “damper1”的 “flange_b”连接到 “spring1”的 “flange_b” • 将 “damper1”的 “flange_a”连接到 “spring1”的 “flange_a” • 将 “spring1”的 “flange_a”连接到 “mass1”的 “flange_b”

	
约束	为简单起见，我们直接在块元素中定义约束；您可以选择定义约束块，在块中使用属性，并将它们的参数绑定到块的属性。

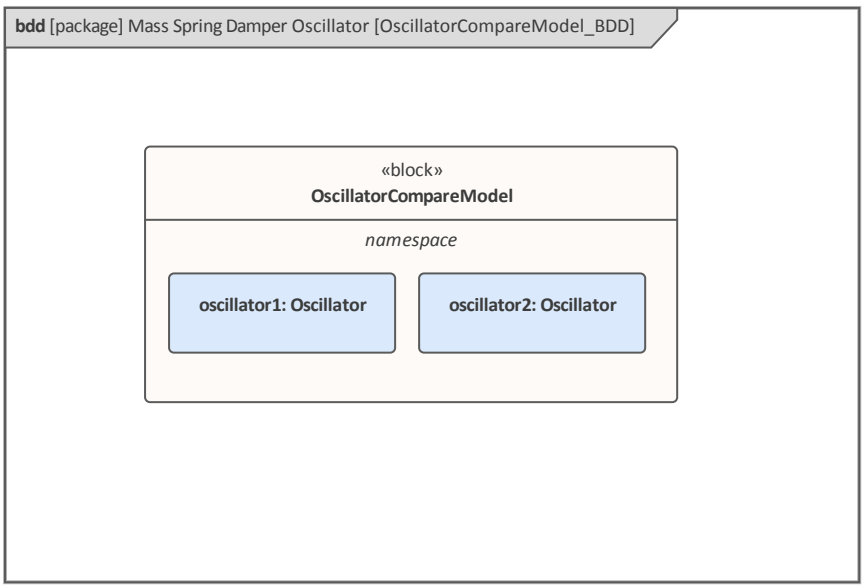
两个振荡器比较计划

在我们对模型进行建模后，我们想做一些假设分析。例如：

- 具有不同阻尼器的两个振荡器有什么区别？
- 如果没有阻尼器怎么办？
- 两个不同弹簧的振荡器有什么区别？
- 两个不同质量的振荡器有什么区别？

以下是创建比较模型的步骤：

- 创建一个名为 "OscillatorCompareModel" 的块
- 为 "OscillatorCompareModel" 创建两个属性，称为振荡器1 和振荡器2，并使用块振荡器键入它们



设置数据集并运行仿真

创建一个配置工件其分配给这个包。然后创建这些数据集：

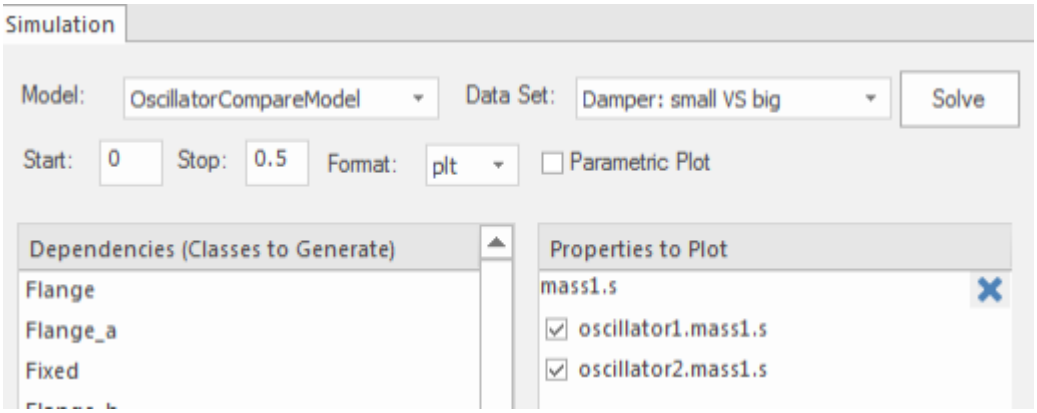
- 阻尼器：小与大
  为 'oscillator1.damper1.d' 提供值 10，为 'oscillator2.damper1.d' 提供较大的值 20
- 阻尼器：否 vs 是
  为 'oscillator1.damper1.d 提供值 0;（'oscillator2.damper1.d' 将使用默认值 25）
- 弹簧：小与大
  提供 'oscillator1.spring1.c' 的值 6000 和 'oscillator2.spring1.c' 的较大值 12000
- 质量：轻与重
  为 "oscillator1.mass1.m" 提供值 0.5，为 "oscillator2.mass1.m" 提供较大的值 2

配置的页面如下所示：

OscillatorCompareModel	SysMLSimModel
Part	
Damper: small VS big	Click button to configure...
oscillator2.damper1.d	20
oscillator1.damper1.d	10
Spring: small VS big	Click button to configure...
oscillator2.spring1.c	12000
oscillator1.spring1.c	6000
Damper: no VS yes	Click button to configure...
oscillator1.damper1.d	0
Mass: light VS Heavy	Click button to configure...
oscillator2.mass1.m	2
oscillator1.mass1.m	0.5

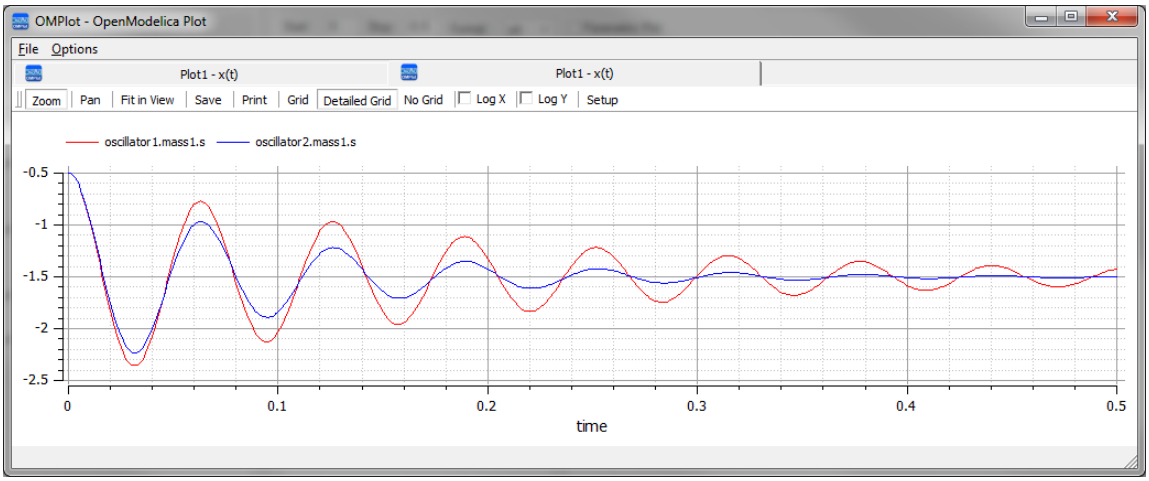
在 "仿真" 页面上，选择 "OscillatorCompareModel"，绘制 "oscillator1.mass1.s" 和 "oscillator2.mass1.s"，然后选择一个创建的数据集并运行仿真。

提示：如果绘图列表中的属性过多，您可以使用列表标题上的时间菜单切换过滤器栏，然后在本例中键入 “上下文”。

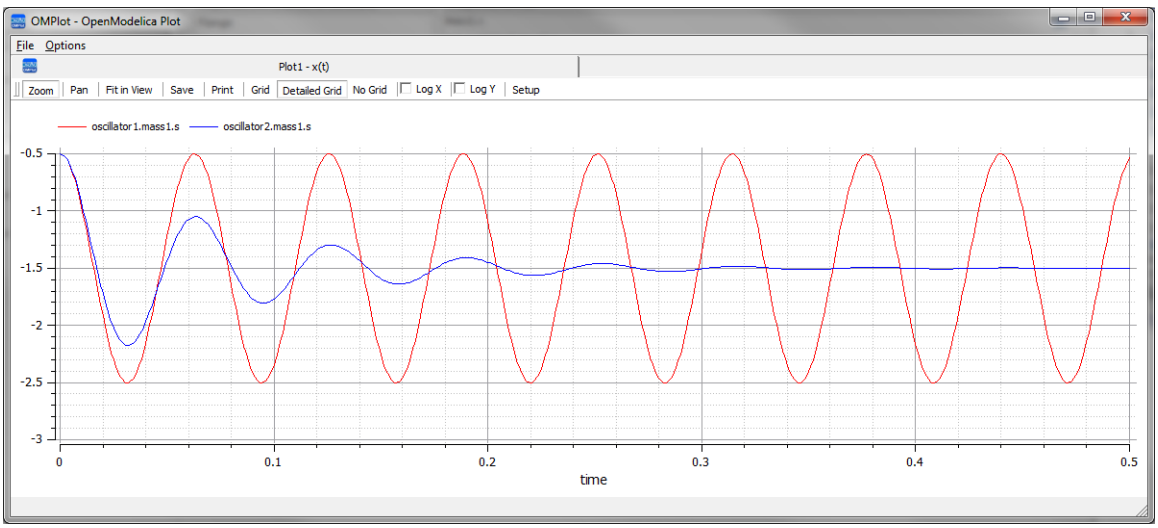


这些是模拟结果：

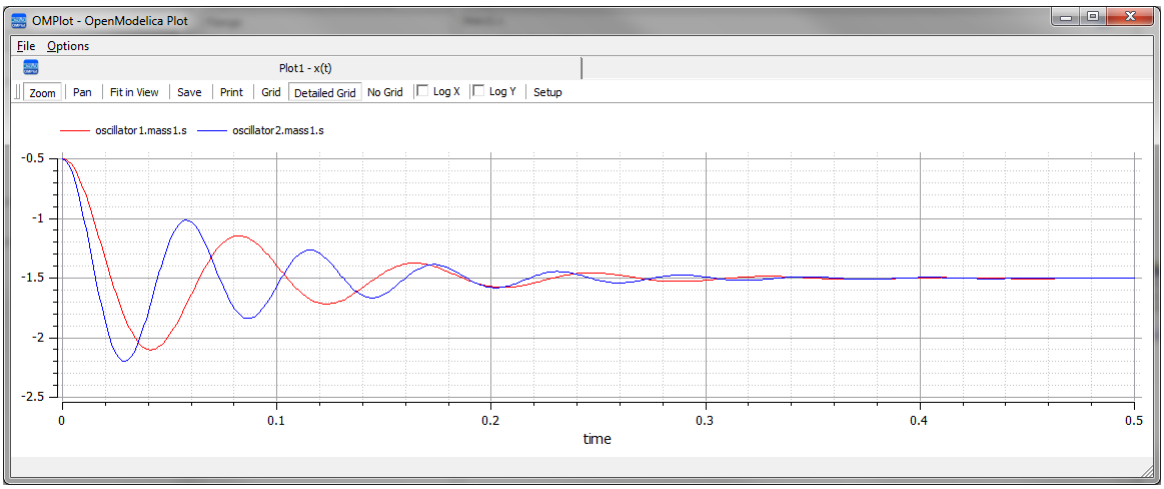
- 阻尼器，小与大：阻尼器越小，车身振动越大



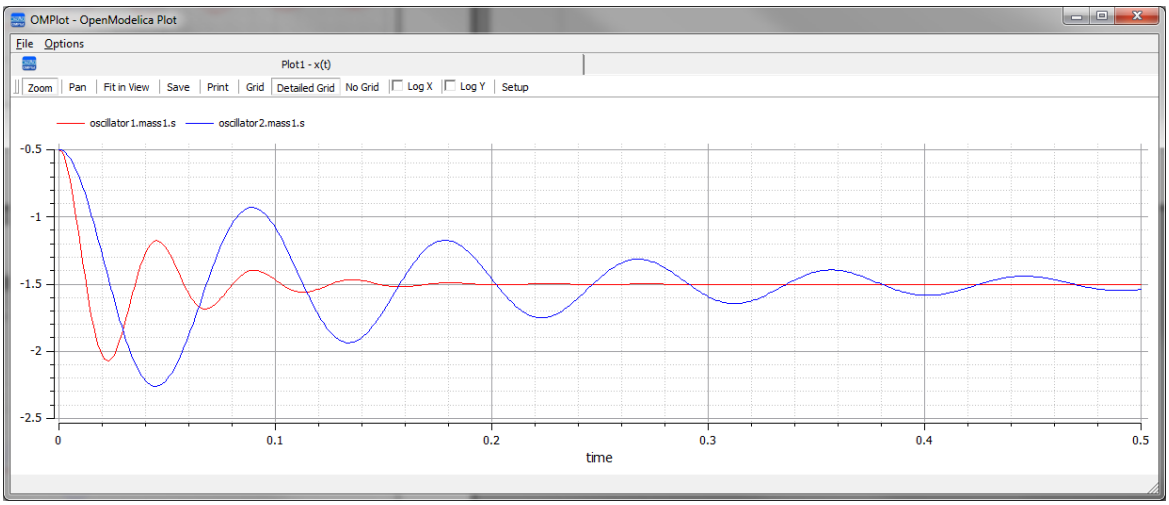
- 阻尼器，否 vs 是：振荡器在没有阻尼器的情况下永远不会停止



- 弹簧，小与大：“c”较　的弹簧会摆动得更慢



- 质量，轻与重：质量较小的object会更快地振动并更快地调节



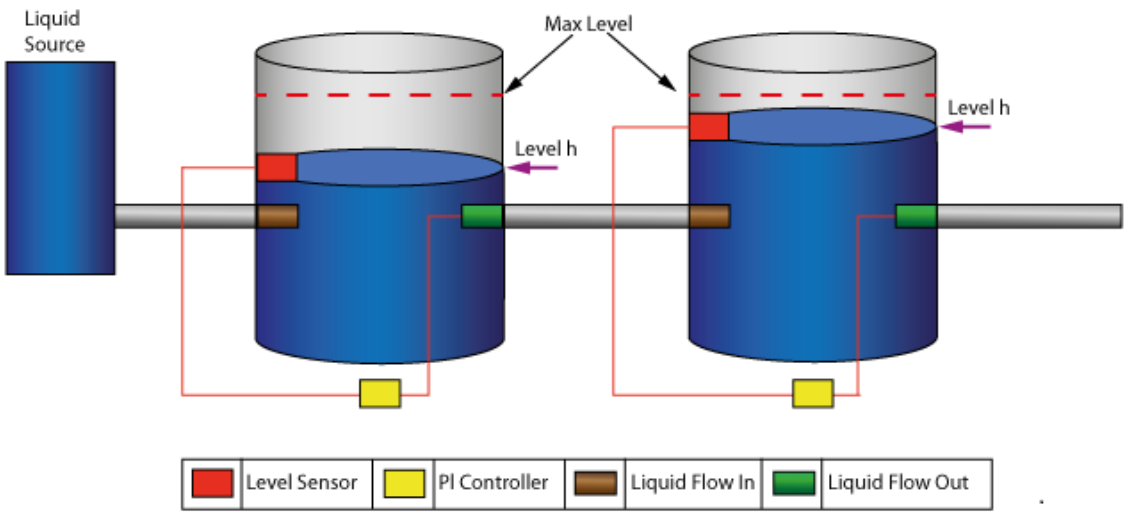
水箱压力调节器

在本节中，我们将介绍为水箱压力调节器创建 SysML 参数模型，该模型由两个连接的水箱、一个源和两个控制器组成，每个控制器监控水位并控制阀门以调节系统.

我们将解释 SysML模型，创建它并设置 SysMLSim 配置。然后我们将使用运行仿真。

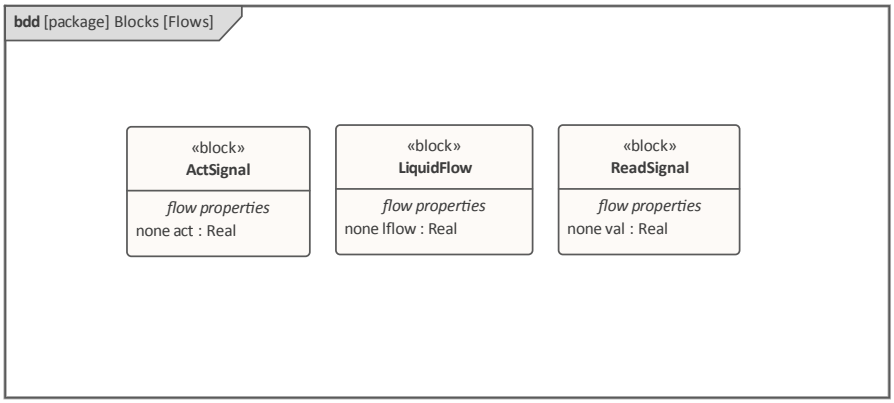
正在建模的系统

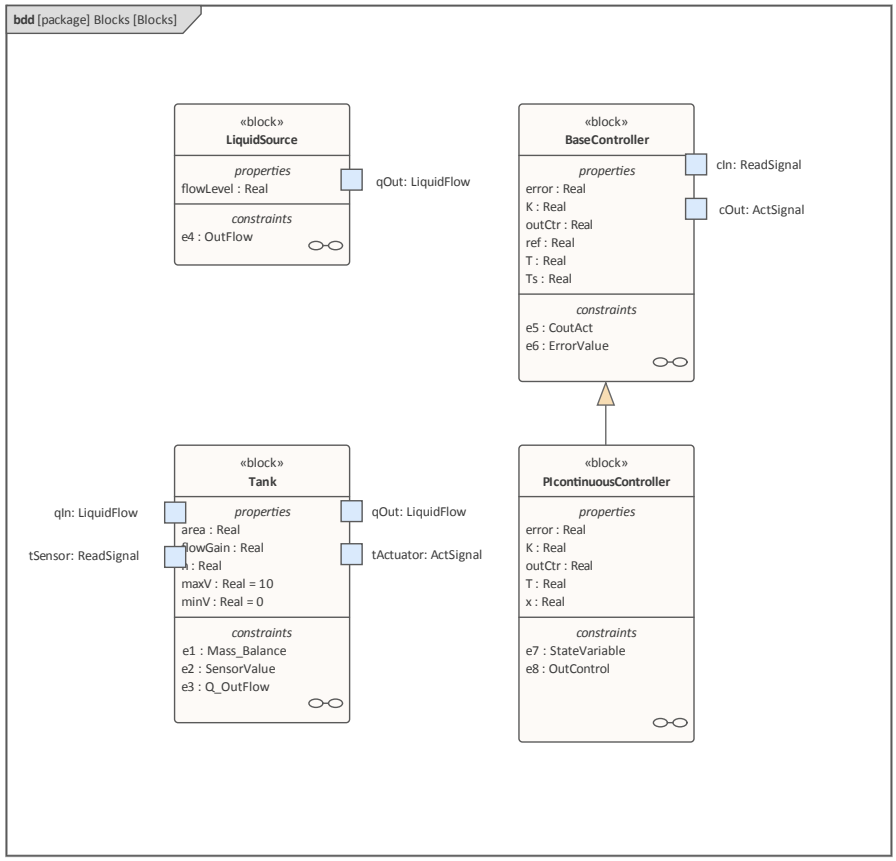
此图描绘了连接在一起的两个源，以及填充第一个水箱的水源。每个水箱都有一个与其相连的比例积分 (PI) 连续控制器，它将水箱中的水位调节到参考水位。当源向第一个水箱注水时，PI 连续控制器根据水箱的实际水位调节水箱的流出量。第一个水箱中的水流入第二个水箱，PI 连续控制器也试图对其进行调节。这是一个自然的、非特定领域的物理问题。



创建 SysML模型

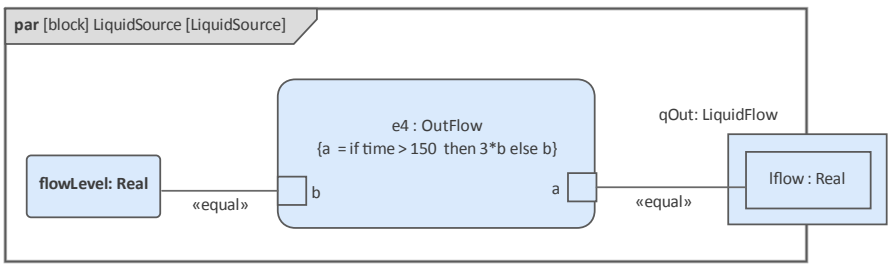
部件	讨论
端口类型	坦克有四个端口，它们分别输入到这三个块中： • ReadSignal：读取液位；这有一个单位为 "m" 的属性 "val" • ActSignal：用于设置阀门位置的执行器信号 • LiquidFlow：入口或出口处的液体流量；这有一个属性 "flow"，单位为 "m^3/s"

<table>
<tr><td></td><td> 
</td></tr>
<tr><td>块定义图表</td><td>LiquidSource：进入水箱的水一定来自某个地方，因此我们在水箱系统中有一个液体源组件，属性flowLevel的单位为 m³/s。 A 端口”键入为 端口”。
水箱：水箱通过端口连接到控制器和液体源。
● 每个 Tank 有四个端口：
- qIn：用于输入流
- qOut：用于输出流
- tSensor：用于提供液位测量
- tActuator：用于设置阀门在出口处的位置
坦克
● 属性：
- 体积（单位='m³'）：罐的容量，参与质量平衡
方程
- h（单位='m'）：水位，参与质量平衡
方程;它的值由传感器读取
- flowGain（单位='m³/s'）：输出流量与阀门有关
按流量增益定位
- minV, maxV：输出阀流量的限制
BaseController：这个块可以是 PI Continuous控制器和 PI Discrete控制器的父级或祖先。
● 端口：
- cIn：输入传感器电平
-控件：控制执行器
● 属性：
- Ts（单位='s'）：离散样本之间的时间周期（未使用
在这个例子中）
- K ：增益因子
- T (unit = 's'): 控制器的时间常数
- 参考：参考水平
- 误差：参考水平与实际水平之间的差异
水位，从传感器获得
- outCtr：到执行器的控制信号，用于控制阀门
位置
PIcontinuousController：从 BaseController 专门化
● 属性：
-x：控制器状态变量</td></tr>
</table>



约束块

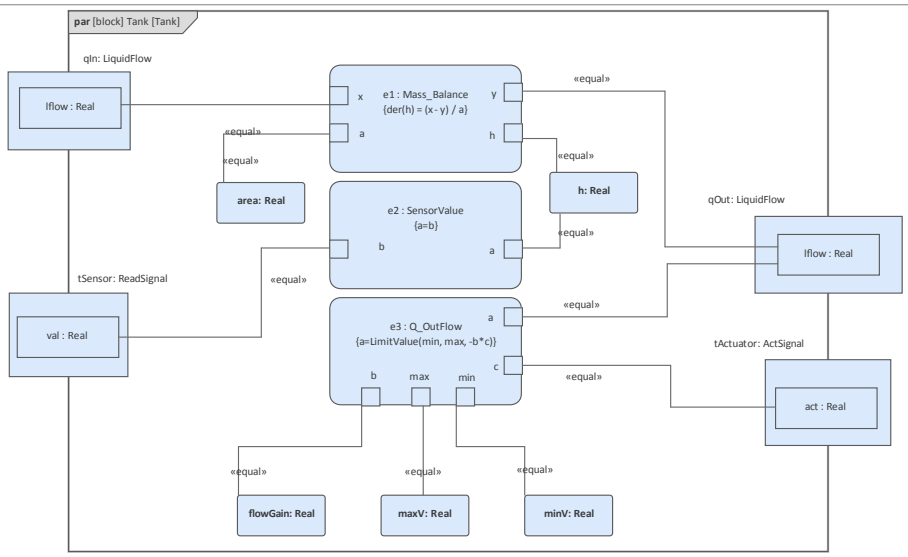
流量在时间=150 时急剧增加至先前流量水平的三倍，这产生了一个有趣的控制问题，罐的控制器必须处理。



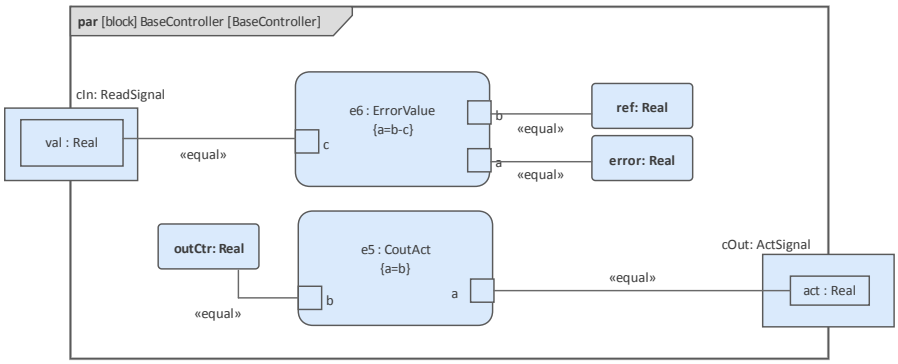
调节罐性能的中心方程是质量平衡方程。

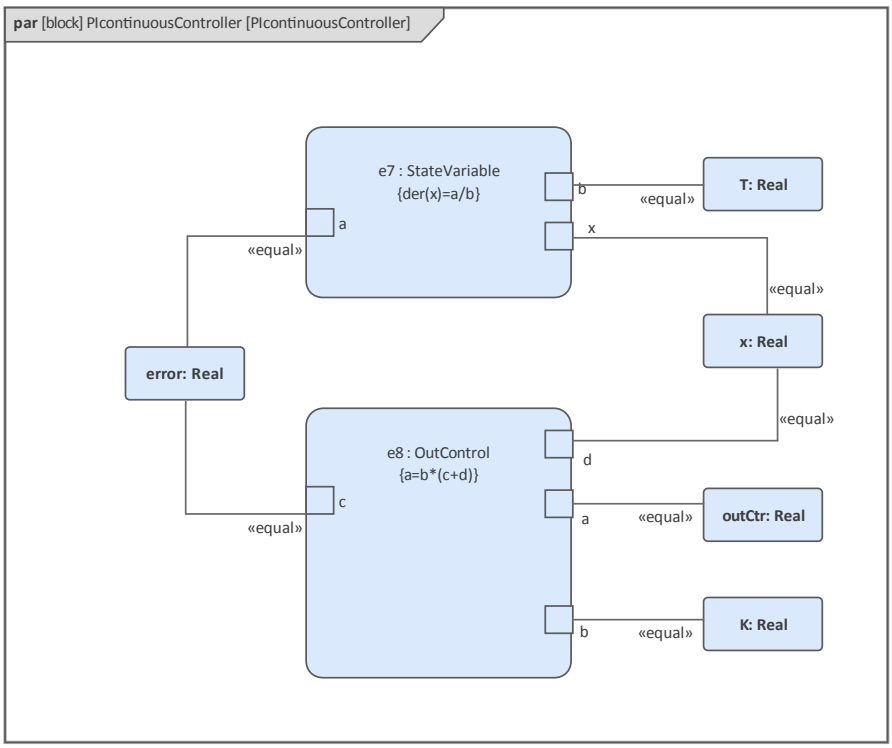
输出流量通过“flowGain”参数 阀门位置相关。

传感器只是读取水箱的液位。

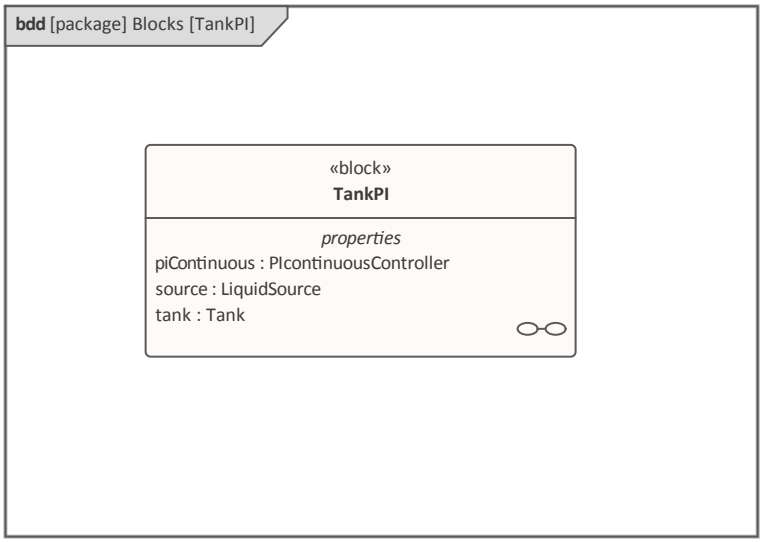


这些图中说明了为“BaseController”和“PIcontinuousController”定义的约束。

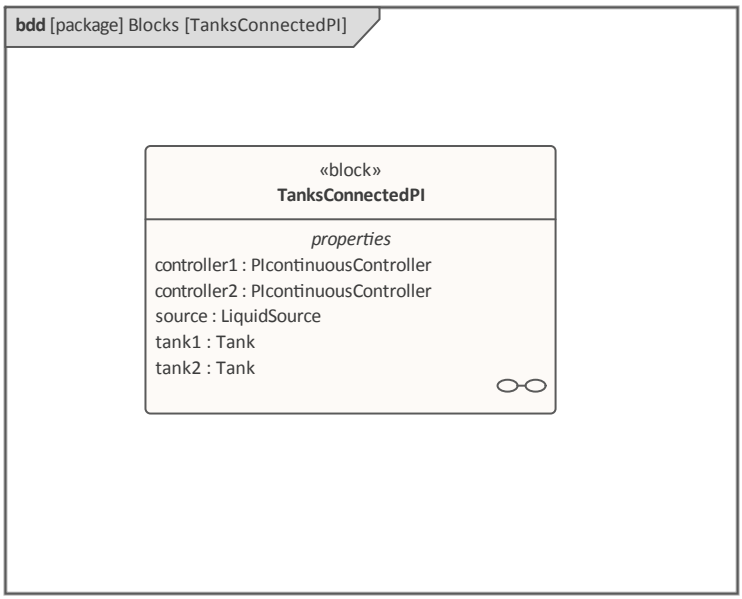




内部块图表	这是具有单个水箱的系统的内部块图。



这是具有两个连接罐的系统的内部块图。



运行仿真

由于*TankPI*和*TanksConnectedPI*被定义为 "sysMLSimModel"，因此它们将列在 "仿真"页面的 "模型"组合框中。

选择*TanksConnectedPI* ，并观察这些 GUI 发生的变化：

- "数据集"组合框：将填充*TanksConnectedPI*中定义的所有数据集
- 'Dependencies' 列表：将自动收集 TanksConnectedPI 直接或间接引用的所有 Blocks、约束、 *SimFunctions*和 ValueTypes（这些元素将生成为 OpenModelica 代码）
- '属性Plot'：将收集一长串'叶子'变量属性（即它们没有属性）；您可以选择一个或几个进行模拟，属性将显示在图例中

工件和配置

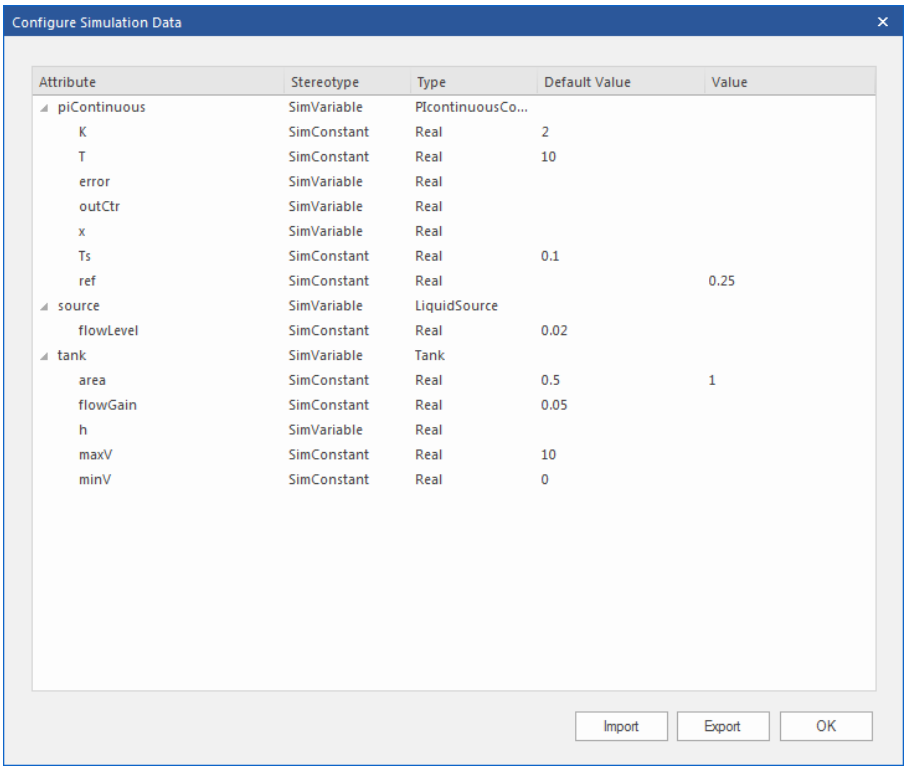
选择 "仿真>系统行为> Modelica/Simulink >配置管理器"
包中的元素将被加载到配置管理器中。
配置这些块及其属性，如本表所示。
注记：未配置为 "属性"的属性默认为 "SimVariable"。

块	属性
液体源	配置为 "配置"。 属性配置： • flowLevel：设置为 "SimConstant"
坦克	配置为 "配置"。 属性配置： • 区域：设置为 "SimConstant" • flowGain：设置为 "SimConstant" • maxV：设置为 "SimConstant" • minV：设置为 "SimConstant"
基本控制器	配置为 "配置"。 属性配置： • K ：设置为 "SimConstant" • T ：设置为 "SimConstant" • Ts：设置为 "SimConstant" • 参考：设置为 "SimConstant"
PI连续控制器	配置为 "配置"。
坦克PI	配置为 "配置"。
TanksConnectedPI	配置为 "配置"。

设置数据集

右键单击每个元素，选择 "创建仿真数据集"选项，然后配置数据集，如本表所示。

元素	数据集
液体源	流量等级：0.02
坦克	h.开始：0 流量增益：0.05 面积：0.5 最大V：10 最小伏特：0
基本控制器	T ：10 K : 2

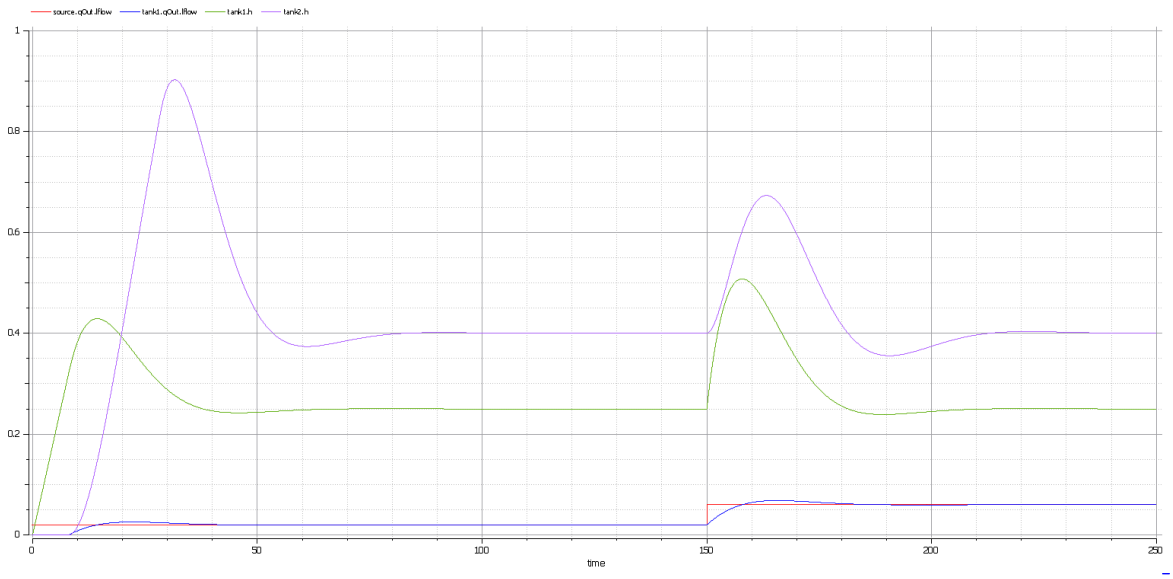
<table>
<tr><td></td><td>TS: 0. 1</td></tr>
<tr><td>PI连续控制器</td><td>无需配置。
默认情况下，特定块将使用超级块的默认数据集中配置的值。</td></tr>
<tr><td>坦克PI</td><td>这里有趣的是可以在 配置仿真数据”对话框中加载默认值。例如，我们在每个块元素上配置为默认数据集的值被加载为属性的默认值。单击每行上的图标可将属性的内部结构扩展到任意深度。
 

<table>
<tr><th>Attribute</th><th>Stereotype</th><th>Type</th><th>Default Value</th><th>Value</th></tr>
<tr><td>piContinuous</td><td>SimVariable</td><td>PIcontinuousCo...</td><td></td><td></td></tr>
<tr><td>K</td><td>SimConstant</td><td>Real</td><td>2</td><td></td></tr>
<tr><td>T</td><td>SimConstant</td><td>Real</td><td>10</td><td></td></tr>
<tr><td>error</td><td>SimVariable</td><td>Real</td><td></td><td></td></tr>
<tr><td>outCtr</td><td>SimVariable</td><td>Real</td><td></td><td></td></tr>
<tr><td>x</td><td>SimVariable</td><td>Real</td><td></td><td></td></tr>
<tr><td>Ts</td><td>SimConstant</td><td>Real</td><td>0.1</td><td></td></tr>
<tr><td>ref</td><td>SimConstant</td><td>Real</td><td></td><td>0.25</td></tr>
<tr><td>source</td><td>SimVariable</td><td>LiquidSource</td><td></td><td></td></tr>
<tr><td>flowLevel</td><td>SimConstant</td><td>Real</td><td>0.02</td><td></td></tr>
<tr><td>tank</td><td>SimVariable</td><td>Tank</td><td></td><td></td></tr>
<tr><td>area</td><td>SimConstant</td><td>Real</td><td>0.5</td><td>1</td></tr>
<tr><td>flowGain</td><td>SimConstant</td><td>Real</td><td>0.05</td><td></td></tr>
<tr><td>h</td><td>SimVariable</td><td>Real</td><td></td><td></td></tr>
<tr><td>maxV</td><td>SimConstant</td><td>Real</td><td>10</td><td></td></tr>
<tr><td>minV</td><td>SimConstant</td><td>Real</td><td>0</td><td></td></tr>
</table>
点击确定按钮，返回配置管理器。然后配置这些值：
• tank.area: 1这会覆盖 Tank块数据集中定义的默认值 0.5
• piContinuous.ref：0.25</td></tr>
<tr><td>TanksConnectedPI</td><td>• 控制器1.ref：0.25
• 控制器2.ref：0.4</td></tr>
</table>

仿真分析1

选择这些变量并单击求解按钮。这个情节应该提示：

- 源.qOut.lflow
- tank1.qOut.lflow
- 坦克1.h
- 坦克2.h



以下是对结果的分析：

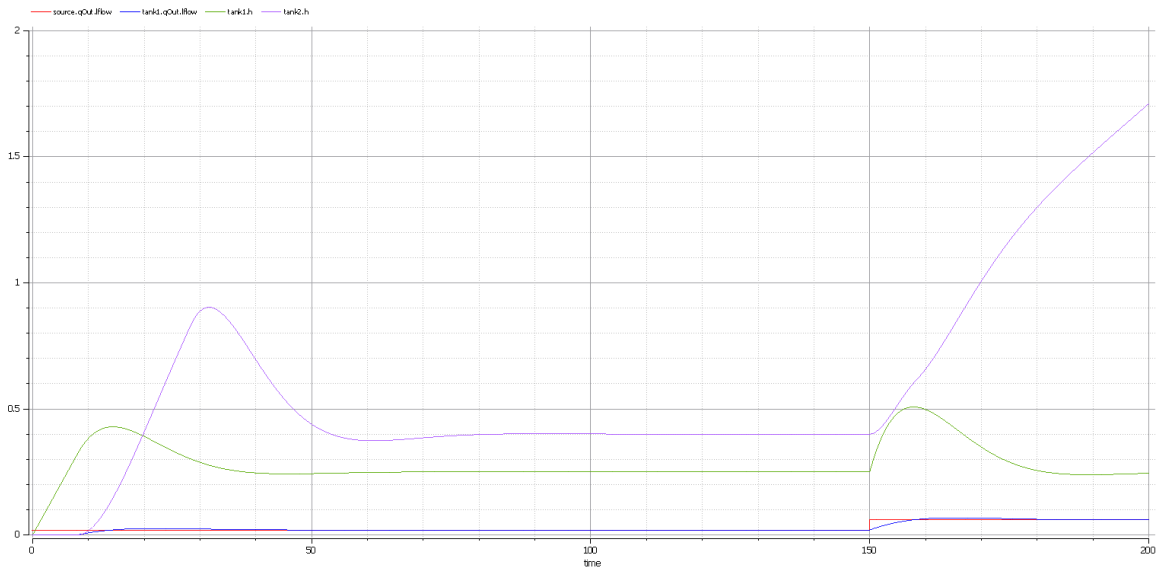
- 液体流量在时间=150 时急剧增加，达到 0.06 m^3/s，是之前流量 (0.02 m^3/s) 的三倍
- Tank1 调节高度 0.25，tank2 调节高度 0.4 符合预期（我们通过数据集设置参数值）
- 在模拟过程中，tank1 和 tank2 都进行了两次调节；首次调节流量0.02 m^3/s；第二次调节流量 0.06 m^3/s
- 在 tank1 有任何流量之前 Tank2 是空的

仿真分析2

在示例中，我们将坦克的属性 "minV" 和 "maxV" 分别设置为值 0 和 10。在现实世界中，10 m^3/s 的流量需要在水箱上安装一个非常大的阀门。

如果我们将 "maxV" 的值更改为 0.05 m^3/s 会发生什么？基于之前的模型，我们可能会做出这些改变：

- 在 TanksConnectedPI 的现有 "DataSet_1" 上，右键单击并选择 "复制数据集"，然后重命名为 "Tank2WithLimitValveSize"
- 单击按钮进行配置，展开 "tank2" 并在 "属性" 的 "值" 列中键入 "0.05"
- 在 "仿真" 页面上选择 "Tank2WithLimitValveSize" 并绘制属性
- 单击求解按钮执行模拟



以下是对结果的分析：

- 我们的更改仅适用于 tank2； tank1 可以像以前一样在 0.02 m ³ /s 和 0.06 m ³ /s 上调节
- 当源流量为0.02 m ³ /s时，tank2可以像以前一样调节
- 然而，当源流量增加到 0.06 m ³ /s 时，阀门太小，无法让流出流量与流入流量匹配；唯一的结果是tank2的水位升高
- 然后由用户来解决这个问题；例如，换一个更大的阀门，减少源流量或制作一个额外的阀门

总之，此示例显示了如何通过复制现有 DataSet 来调整参数值。