

Simulink®

快速入门指南



MATLAB® & SIMULINK®

R2022a



如何联系 MathWorks



最新动态: www.mathworks.com
销售和服务: www.mathworks.com/sales_and_services
用户社区: www.mathworks.com/matlabcentral
技术支持: www.mathworks.com/support/contact_us



电话: 010-59827000



迈斯沃克软件 (北京) 有限公司
北京市朝阳区望京东园四区 6 号楼
北望金辉大厦 16 层 1604

Simulink® 快速入门指南

© COPYRIGHT 1990–2022 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

商标

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

专利

MathWorks products are protected by one or more U.S. patents. Please see www.mathworks.com/patents for more information.

修订历史记录

2005 年 9 月	仅限在线版本
2006 年 3 月	仅限在线版本
2006 年 9 月	仅限在线版本
2007 年 3 月	第一次印刷
2007 年 9 月	第二次印刷
2008 年 3 月	第三次印刷
2008 年 10 月	第四次印刷
2009 年 3 月	第五次印刷
2009 年 9 月	仅限在线版本
2010 年 3 月	仅限在线版本
2010 年 9 月	仅限在线版本
2011 年 4 月	仅限在线版本
2011 年 9 月	第六次印刷
2012 年 3 月	第七次印刷
2012 年 9 月	第八次印刷
2013 年 3 月	第九次印刷
2013 年 9 月	第十次印刷
2014 年 3 月	第十一次印刷
2014 年 10 月	第十二次印刷
2015 年 3 月	第十三次印刷
2015 年 9 月	第十四次印刷
2015 年 10 月	仅限在线版本
2016 年 3 月	第十五次印刷
2016 年 9 月	第十六次印刷
2017 年 3 月	第十七次印刷
2017 年 9 月	第十八次印刷
2018 年 3 月	第十九次印刷
2018 年 9 月	第二十次印刷
2019 年 3 月	仅限在线版本
2019 年 9 月	仅限在线版本
2020 年 3 月	仅限在线版本
2020 年 9 月	仅限在线版本
2021 年 3 月	仅限在线版本
2021 年 9 月	仅限在线版本
2022 年 3 月	仅限在线版本

版本 6.3 (版本 14SP3) 中的新增内容
Simulink 6.4 (版本 2006a) 中的修订内容
Simulink 6.5 (版本 2006b) 中的修订内容
Simulink 6.6 (版本 2007a) 中的修订内容
Simulink 7.0 (版本 2007b) 中的修订内容
Simulink 7.1 (版本 2008a) 中的修订内容
Simulink 7.2 (版本 2008b) 中的修订内容
Simulink 7.3 (版本 2009a) 中的修订内容
Simulink 7.4 (版本 2009b) 中的修订内容
Simulink 7.5 (版本 2010a) 中的修订内容
Simulink 7.6 (版本 2010b) 中的修订内容
Simulink 7.7 (版本 2011a) 中的修订内容
Simulink 7.8 (版本 2011b) 中的修订内容
Simulink 7.9 (版本 2012a) 中的修订内容
Simulink 8.0 (版本 2012b) 中的修订内容
Simulink 8.1 (版本 2013a) 中的修订内容
Simulink 8.2 (版本 2013b) 中的修订内容
Simulink 8.3 (版本 2014a) 中的修订内容
Simulink 8.4 (版本 2014b) 中的修订内容
Simulink 8.5 (版本 2015a) 中的修订内容
Simulink 8.6 (版本 2015b) 中的修订内容
Simulink 8.5.1 (版本 2015aSP1) 中的再发布内容
Simulink 8.7 (版本 2016a) 中的修订内容
Simulink 8.8 (版本 2016b) 中的修订内容
Simulink 8.9 (版本 2017a) 中的修订内容
Simulink 9.0 (版本 2017b) 中的修订内容
Simulink 9.1 (版本 2018a) 中的修订内容
Simulink 9.2 (版本 2018b) 中的修订内容
Simulink 9.3 (版本 2019a) 中的修订内容
Simulink 10.0 (版本 2019b) 中的修订内容
Simulink 10.1 (版本 2020a) 中的修订内容
Simulink 10.2 (版本 2020b) 中的修订内容
Simulink 10.3 (版本 2021a) 中的修订内容
Simulink 10.4 (版本 2021b) 中的修订内容
Simulink 10.5 (版本 2022a) 中的修订内容

简介

1

Simulink 产品说明	1-2
主要功能	1-2
使用 Simulink 进行基于模型的设计	1-3
Simulink 中基于模型的设计工作流示例	1-4
系统定义和布局	1-6
确定建模目的	1-6
确定系统组件和接口	1-7
建模并验证系统	1-12
打开系统布局	1-12
对组件进行建模	1-12
通过仿真来验证组件	1-16
验证模型	1-18
在 Simulink 中设计系统	1-22
打开系统模型	1-22
确定设计的组件和设计目的	1-22
通过仿真来分析系统行为	1-23
设计组件并验证设计	1-25

在 Simulink 中进行建模

2

Simulink 模块图	2-2
---------------------------	------------

简单的 Simulink 模型

3

创建简单模型	3-2
打开新模型	3-3
打开 Simulink 库浏览器	3-5
将模块添加到模型	3-7
连接模块	3-7
添加信号查看器	3-8
运行仿真	3-8
细化模型	3-9

探索模型层次结构	4-2
查看模型层次结构	4-2
查看信号属性	4-3
跟踪信号	4-5

简介

- “Simulink 产品说明” (第 1-2 页)
- “使用 Simulink 进行基于模型的设计” (第 1-3 页)
- “系统定义和布局” (第 1-6 页)
- “建模并验证系统” (第 1-12 页)
- “在 Simulink 中设计系统” (第 1-22 页)

Simulink 产品说明

仿真和基于模型的设计

Simulink 是一个模块图环境，用于多域仿真以及基于模型的设计。它支持系统级设计、仿真、自动代码生成以及嵌入式系统的连续测试和验证。Simulink 提供图形编辑器、可自定义的模块库以及求解器，能够进行动态系统建模和仿真。Simulink 与 MATLAB® 相集成，这样您不仅能够在 Simulink 中将 MATLAB 算法融入模型，还能将仿真结果导出至 MATLAB 做进一步分析。

主要功能

- 图形编辑器，用于构建和管理层次结构模块图
- 预定义模块库，用于构建连续时间和离散时间系统模型
- 仿真引擎，配有定步长和变步长 ODE 求解器
- 示波器和数据显示模块，用于查看仿真结果
- 工程和数据管理工具，用于管理模型文件和数据
- 模型分析工具，用于优化模型架构并提高仿真速度
- MATLAB Function 模块，用于将 MATLAB 算法导入到模型中
- 代码继承工具，用于将 C 和 C++ 代码导入到模型中

使用 Simulink 进行基于模型的设计

建模是为真实系统创建虚拟表示的一种方法。您可以在各种条件下仿真此虚拟表示，以观察其行为。

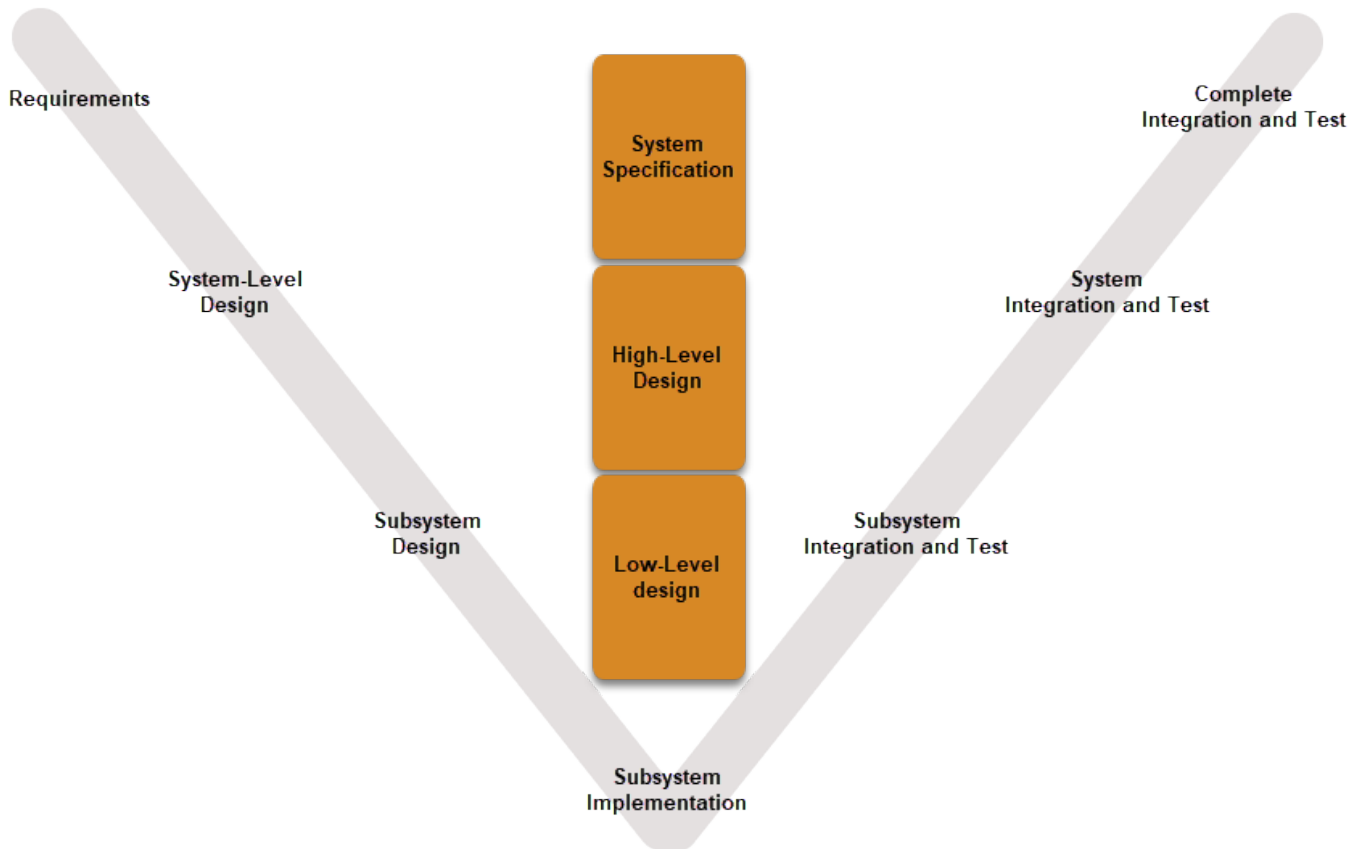
建模和仿真对于测试单独使用硬件原型时难以重现的情形非常有用。在尚无硬件可用的设计过程早期阶段尤其如此。交替进行建模和仿真可以减少设计过程后期发现的错误数量，从一开始就提高系统设计的质量。

您可以从模型中自动生成代码，当包含软件和硬件实现要求时，还可以创建测试平台进行系统验证。代码生成可以节省时间，并避免引入手动编码错误。

在基于模型的设计中，系统模型是工作流的核心。利用基于模型的设计，可以快速且经济高效地开发动态系统，包括控制系统、信号处理系统和通信系统。

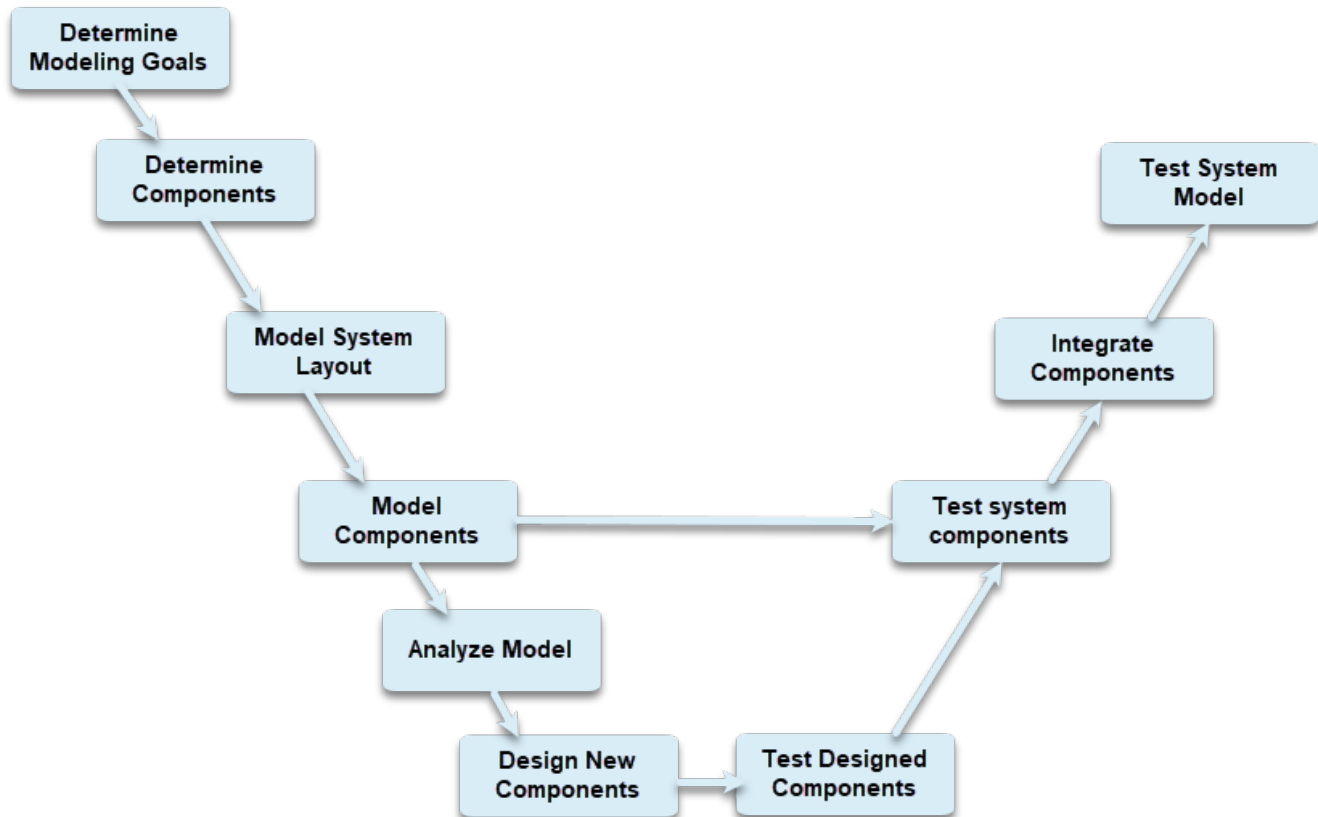
基于模型的设计允许您：

- 让各个工程团队使用同一个设计环境
- 将设计与要求直接挂钩
- 将测试与设计相结合，不断发现并更正错误
- 通过多域仿真对算法进行细化
- 自动生成嵌入式软件代码和文档
- 开发和重用测试套件



Simulink 中基于模型的设计工作流程示例

开始基于模型的设计任务之前，请考虑以下 workflow。



本教程中的 workflow 重点介绍与基于模型的设计相关的基本 Simulink 任务。

- “系统定义和布局”（第 1-6 页） - 确定建模目的、确定组件、建模系统布局。
- “建模并验证系统”（第 1-12 页） - 建模并测试组件、集成组件、测试系统。
- “在 Simulink 中设计系统”（第 1-22 页） - 设计并测试新组件。

此 workflow 中的前两个任务为现有系统建模，并建立设计组件的环境。此 workflow 的下一步是实现新组件。您可以使用快速原型和嵌入式代码生成产品（例如 Simulink Real-Time™ 和 Embedded Coder®）来生成代码，并将设计用于真实的物理系统。

另请参阅

相关示例

- “系统定义和布局”（第 1-6 页）
- “建模并验证系统”（第 1-12 页）
- “在 Simulink 中设计系统”（第 1-22 页）
- “组织大型建模工程”

外部网站

- [Simulink 概述](#)
- [利用 MATLAB 和 Simulink 进行基于模型的设计](#)

系统定义和布局

本节内容

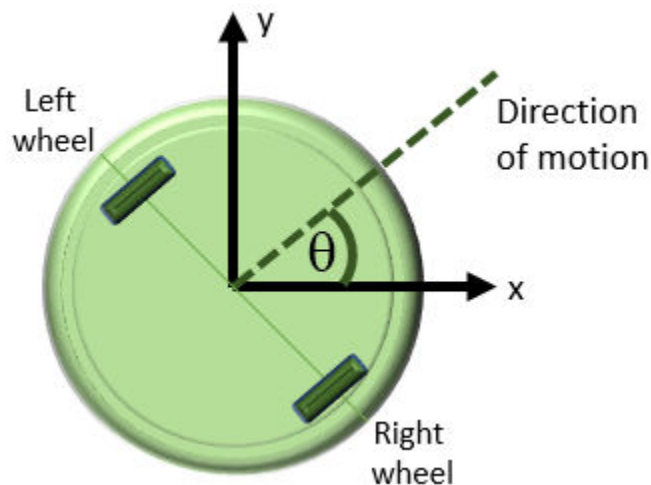
“确定建模目的” (第 1-6 页)

“确定系统组件和接口” (第 1-7 页)

Simulink 模型的顶层系统布局是许多工程团队可以使用的公共环境，是基于模型的设计范式：分析、设计、检验和实现。您可以通过确定模型的结构和各个组件来定义顶层系统。然后，您可以将模型按照层次结构进行组织，分别与系统的各个组件相对应。接下来，您可以定义每个组件的接口以及组件之间的连接。

本教程中讨论的模型是一个平板机器人，它可以借助两个轮子进行移动或旋转，类似于家用扫地机器人。此模型假设机器人以下列两种方式之一进行移动：

- 线性 - 两个轮子以相同的速度向同一方向转动，机器人线性移动。
- 旋转 - 两个轮子以相同的速度向相反方向转动，机器人原地旋转。



这两种运动都从静止状态开始，也就是说，旋转速度和线性速度均为零。按照这些假设，您可以对线性运动组件和旋转运动组件分别建模。

确定建模目的

在设计模型之前，请明确您的目的和需求。目的决定模型的结构和详细程度。如果目的只是弄清楚机器人可以走多快，那么只对线性运动进行建模就足够了。如果目的是为设备设计一组输入，使它按照给定的路径移动，那就会涉及到旋转组件。如果目的是避障，那么系统就需要一个传感器。本教程的模型构建目的是设计一些传感器参数，使机器人能够在检测到路径中的障碍物时及时停止。要实现此目的，模型必须：

- 确定电机停止后机器人多长时间能停止下来
- 提供一系列线性运动和旋转运动的命令，使机器人能够在二维空间内移动

第一个建模目的是让您能够分析运动，以便设计传感器。第二个目的是让您能够对设计进行测试。

确定系统组件和接口

明确建模需求之后，即可开始确定系统的组件。确定顶层结构的各个组件及其关系有助于您以系统化方式构建比较复杂的模型。这些步骤是在您构建模型之前在 Simulink 外部进行的。

执行此任务需要回答以下问题：

- 系统需要哪些结构组件和功能组件？当布局能够反映物理结构和功能结构时，将有助于理解、构建、进行通信和测试系统。当需要在设计过程中的不同阶段实现系统的各个组成部分时，这一点将变得更加重要。
- 每个组件的输入和输出是什么？绘制一个显示各组件连接情况的图形。此图可帮助您可视化模型中的信号流，确定每个信号的信源和信宿，并确定是否所有必要的组件都存在。
- 需要达到多高的详细程度？在图中包括主要系统参数。绘制系统图可以帮助您识别并建模对要观察的行为非常重要的部件。实现建模目的所需要的每个组件和参数在模型中都要有一个对应的表示，但需要在模型的复杂程度和可读性之间进行权衡。建模可以是一个反复的过程。您可以从具有很少细节的简要模型开始，然后根据需要逐渐提高复杂程度。

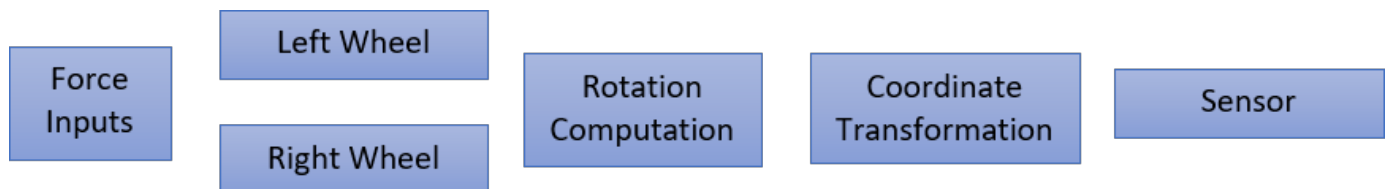
思考以下问题通常也很有帮助：

- 系统的哪些组成部分需要测试？
- 测试数据和成功的标准是什么？
- 哪些输出是分析和设计任务所必需的？

确定机器人运动组件

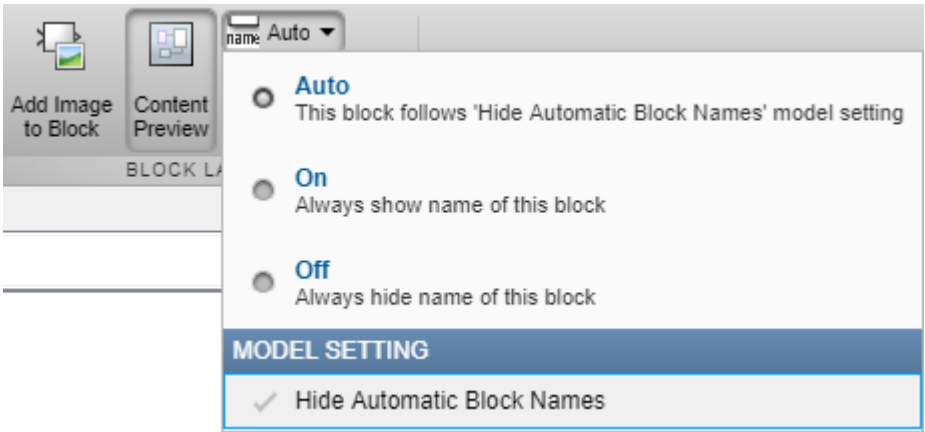
本教程中的系统定义了一个机器人，它通过两个电动轮子在两个维度上移动。其中包括：

- 线性运动特性
- 旋转运动特性
- 用于确定系统在两个维度上的位置的变换
- 用于测量机器人与障碍物之间的距离的传感器

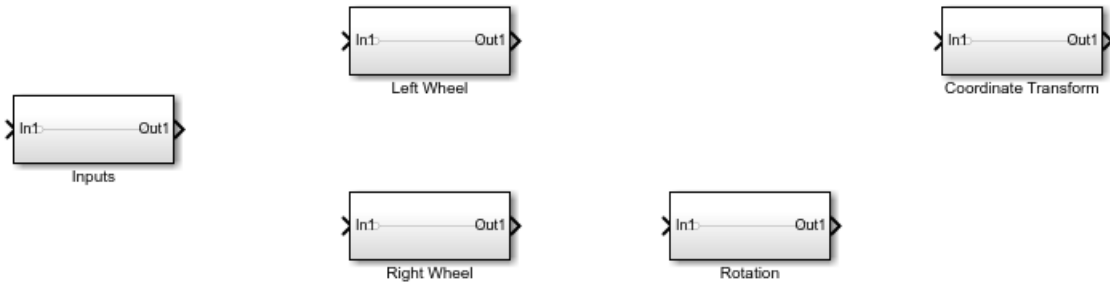


此系统的模型包括两个相同的轮子、施加在轮子上的输入作用力、旋转动态特性、坐标变换和一个传感器。此模型使用 Subsystem 表示每个组件：

- 1 打开一个新的 Simulink 模型。请参阅“打开新模型”（第 3-3 页）。
- 2 打开库浏览器。请参阅“打开 Simulink 库浏览器”（第 3-5 页）。
- 3 添加 Subsystem 模块。将五个 Subsystem 模块从 Ports & Subsystems 库拖放到新模型中。
- 4 点击一个子系统。在**格式**选项卡中，点击**自动名称**下拉列表。清除**隐藏自动模块名称**复选框。



5 按如下所示放置并重命名这五个 Subsystem 模块。要更改模块名称，请双击模块名称并编辑文本。



定义组件之间的接口

确定子系统之间的输入和输出连接。输入和输出值在仿真过程中动态变化。模块之间的连接线代表数据传输。下表显示每个组件的输入和输出。

模块	输入	输出	相关信息
输入	无	右轮作用力 左轮作用力	不适用
右轮	右轮作用力	右轮速度	有方向性，负值表示相反方向
左轮	左轮作用力	左轮速度	有方向性，负值表示相反方向
旋转	左右轮之间的速度差	旋转角度	逆时针测量
坐标变换	正常速度 旋转角度	X 速度 Y 速度	不适用
传感器	X 坐标 Y 坐标	无	建模不需要任何模块。

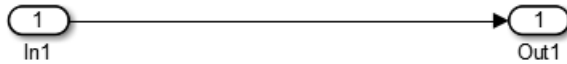
一些模块输入与模块输出并不完全匹配。因此，除了各个组件的动态特性之外，模型还必须计算以下内容：

- 旋转计算的输入 - 两个轮子的速度相减并除以 2。
- 坐标变换的输入 - 两个轮子的平均速度。
- 传感器的输入 - 对坐标变换的输出进行积分。

轮子的速度大小始终相等，并且计算在该假设内是精确的。

添加必要的组件并完成连接：

- 1 为每个子系统添加必要的输入端口和输出端口。双击 Subsystem 模块。



每个新的 Subsystem 模块都包含一个 Inport (In1) 和一个 Outport (Out1) 模块。这些模块定义信号与模型层次结构中下一个更高层级进行交互的接口。

每个 Inport 模块在 Subsystem 模块上创建一个输入端口，每个 Outport 模块创建一个输出端口。模型通过输入/输出端口名称反映这些模块的名称。为更多的输入和输出信号添加更多模块。在

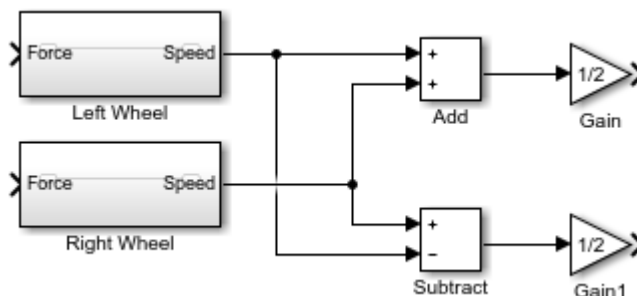
Simulink 编辑器工具栏上，点击[向上导航到父级按钮](#) 返回顶层。

对于每个模块，添加并重命名 Inport 和 Outport 模块。

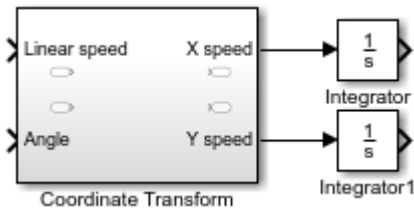


在复制 Inport 模块以创建新模块时，请使用[粘贴](#) (Ctrl+V) 选项。

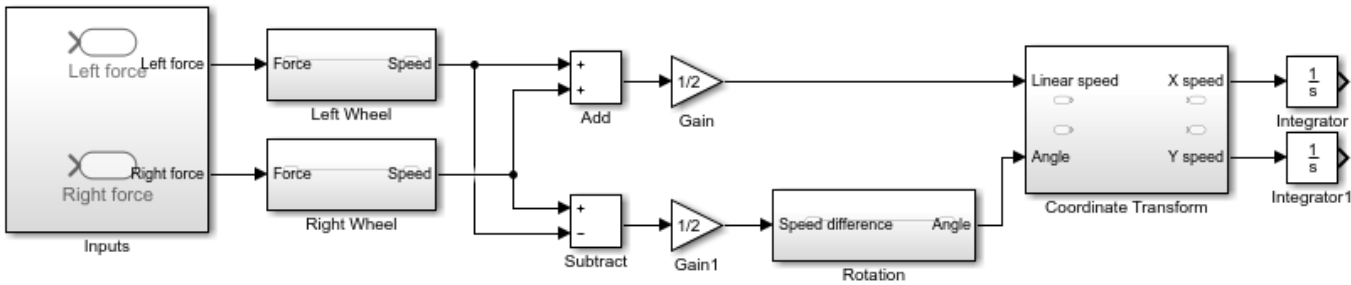
- 2 根据左轮和右轮速度计算 Coordinate Transform 和 Rotation 子系统所需的输入。
 - a 计算 Coordinate Transform 子系统的线性速度输入。从 Math Operations 库中添加一个 Add 模块，并将两个轮子组件的输出连接起来。添加一个 Gain 模块，并将 Gain 参数设置为 $1/2$ 。将 Add 模块的输出连接到此 Gain 模块。
 - b 计算 Rotation 子系统的速度差输入。从 Math Operations 库中添加一个 Subtract 模块。将右轮速度连接到 + 输入，将左轮速度连接到 - 输入。连接两个轮子组件的输出。添加一个 Gain 模块，并将 Gain 参数设置为 $1/2$ 。将 Subtract 模块的输出连接到此 Gain 模块。



- 3 根据 X 速度和 Y 速度计算 X 坐标和 Y 坐标。从 Continuous 库中添加两个 Integrator 模块，并连接 Coordinate Transform 模块的输出。将 Integrator 模块的初始条件设置保留为 0。



- 4 完成系统连接。



参数和数据

确定模型需要的参数以及参数的值。根据建模目的决定这些值是始终固定，还是针对不同的仿真而更改。实现建模目的所需要的参数必须在模型中显式表示。下表可以帮助您确定对每个组件进行建模时的详细程度。

参数	模块	符号	值	类型
质量	左轮	m	2.5 kg	可变
	右轮			
滚阻	左轮	k_drag	30 Ns ² /m	可变
	右轮			
机器人半径	旋转	r	0.15 m	可变
初始角度	旋转	无	0 rad	固定
初始速度	左轮	无	0 m/s	固定
	右轮		0 m/s	
初始 (X, Y) 坐标	积分器	无	(0, 0) m	固定

Simulink 使用 MATLAB 工作区来计算参数。可在 MATLAB 命令行窗口中设置这些参数：

```
m = 2.5;  
k_drag = 30;  
r = 0.15;
```


另请参阅

相关示例

- “建模并验证系统” (第 1-12 页)
- “在 Simulink 中设计系统” (第 1-22 页)

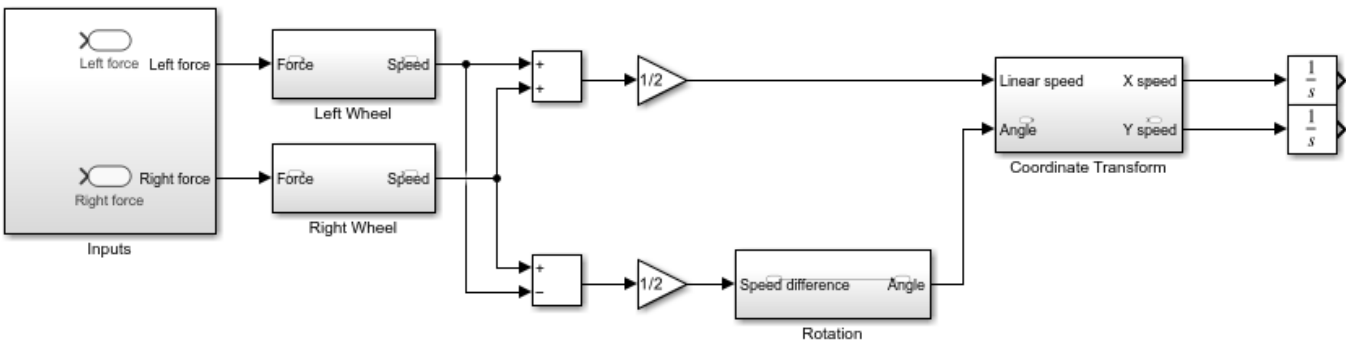
建模并验证系统

您可以对系统结构中的每个组件进行建模，以表示该组件的物理行为或功能行为。通过使用测试数据对组件进行仿真，以验证它们的基本行为。

打开系统布局

对各个组件进行建模时，需要从大局上把握整个系统布局。首先加载布局模型。在 MATLAB® 命令行中，输入：

```
open_system('system_layout.slx')
```



Copyright 2018 The MathWorks, Inc.

对组件进行建模

包含一个组件的 Simulink 模型基于以下几点：

- 物理组件的输出和输入之间的显式数学关系 - 您可以根据组件的输入通过代数计算和微分方程积分，直接或间接计算出组件的输出。例如，按照给定的进水速率计算水箱中的水位就是一种显式关系。每个 Simulink 模块基于从输入到输出的计算定义来执行。
- 物理组件的模型变量之间的隐式数学关系 - 由于变量之间相互依赖，因此为组件指定输入和输出并不容易。例如，电路中连接的电机的 + 极电压和 - 极电压之间就存在一种隐式关系。要在 Simulink 中对这种关系进行建模，您可以使用 Simscape™ 等物理建模工具，也可以将这些变量建模为允许定义输入/输出的更大组件的一部分。有时候，更仔细地审视建模目的和组件定义有助于定义输入/输出关系。
- 从实际系统获得的数据 - 您已经测得实际组件的输入/输出数据，但不存在完全定义的数学关系。许多设备具有符合此描述的未建模组件。例如，电视机散发的热量。您可以使用 System Identification Toolbox™ 来定义此类系统的输入/输出关系。
- 显式功能定义 - 您可以根据功能组件的输入通过代数计算和逻辑计算来定义功能组件的输出。例如，调温器的开关逻辑。您可以将大多数功能关系建模为 Simulink 模块和子系统。

本教程对具有显式输入/输出关系的物理组件和功能组件进行建模。在本教程中，您将：

- 1 使用系统方程创建一个 Simulink 模型。
- 2 在 Simulink 编辑器中添加并连接 Simulink 模块。模块代表方程中的系数和变量。
- 3 分别为每个组件构建模型。构建系统模型最有效的方法是首先独立地考虑各个组件。

- 4 首先，使用接近于系统的模型构建简单的模型。找出可能会影响模型准确性的假设条件。以迭代方式添加细节，直到复杂度满足建模和准确性要求。

对物理组件进行建模

描述组件之间的关系，例如数据、能量和力的传递。在 Simulink 中使用系统方程构建系统的图形化模型。

为组件建模之前，需要思考以下问题：

- 每个组件的常量是什么？什么值不会更改，除非您更改它们？
- 每个组件的变量是什么？什么值会随着时间而更改？
- 一个组件有多少个状态变量？

根据科学原理推导出每个组件的方程。许多系统方程不外乎以下三种类别：

- 对于连续系统，微分方程描述变量的变化率，为所有时间值都定义方程。例如，一阶微分方程给出车速：

$$\frac{dv(t)}{dt} = -\frac{b}{m}v(t) + u(t)$$

- 对于离散系统，差分方程描述变量的变化率，但只在特定时间定义方程。例如，来自离散比例微分控制器的控制信号：

$$pd[n] = (e[n] - e[n-1])K_d + e[n]K_p$$

- 没有导数的方程是代数方程。例如，用代数方程表示包含两个组件的并联电路中的总电流：

$$I_t = I_a + I_b$$

轮子和线性运动

作用在轮子上的力有两个：

- 电机施加的力 - 此力 F 作用在速度变化的方向上，是轮子子系统的输入。
- 阻力 - 此力 F_{drag} 作用在速度变化的相反方向上，是速度的函数。

$$F_{drag} = k_{drag}V|V|$$

加速度与这两个力之和成正比：

$$\begin{aligned}(m/2)\dot{V} &= F - F_{drag} \\ (m/2)\dot{V} &= F - k_{drag}V|V| \\ \dot{V} &= \frac{F - k_{drag}V|V|}{(m/2)}\end{aligned}$$

其中 k_{drag} 是阻力系数， m 是机器人的质量。每个轮子承载机器人一半的质量。

构建轮子模型：

- 1 在 `system_layout` 模型中，双击 Right Wheel 子系统以显示空子系统。
- 2 对速度和加速度进行建模。添加一个 Integrator 模块。将初始条件设置保留为 0。此模块的输入是加速度 \dot{V} ，输出是速度 V 。





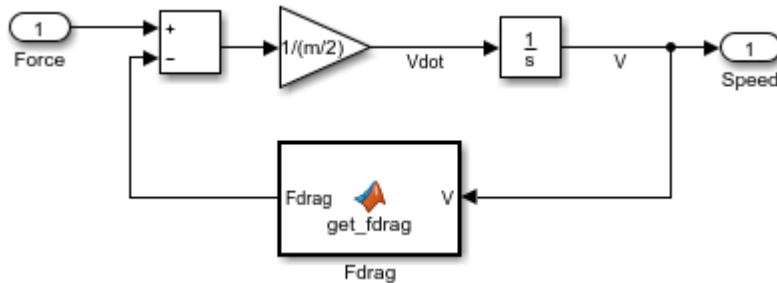
- 3 对阻力进行建模。从 User-Defined Functions 库中添加一个 MATLAB Function 模块。MATLAB Function 模块提供一种在模型中实现数学表达式的快速方法。要编辑函数，请双击该模块以打开 MATLAB Function 编辑器。

```

Right Wheel/Fdrag  ×  +
1  function Fdrag = get_fdrag(V,k_drag)
2
3  Fdrag = k_drag*V*abs(V);
4

```

- 4 定义 MATLAB Function 模块的参数。在 MATLAB Function 模块编辑器中，点击**编辑数据**  按钮。点击 k_drag，将**作用域**设置为**参数**并点击**应用**。
- 5 用 Subtract 模块从电机的作用力中减去阻力。使用带有参数 1/(m/2) 的 Gain 模块完成力-加速度方程。
- 6 要反转 MATLAB Function 模块的方向，请选择该模块。在工具条的**格式**选项卡上，点击**左右翻转** 。连接这些模块。



- 7 这两个轮子的动态特性相同。复制您刚刚建模的 Right Wheel 子系统，并将其粘贴到 Left Wheel 子系统中。
- 8 查看模型的顶层。点击**向上导航到父级**按钮 。

旋转运动

当两个轮子沿相反方向转动时，它们沿半径为 r 的圆周运动，从而产生机器人的旋转运动。当这些轮子向相同方向转动时，没有旋转。假设轮子的速度大小始终相等，则可将旋转运动视为两个轮子速度 V_R 与 V_L 之差的因变量来对其进行建模：

$$\dot{\theta} = \frac{V_R - V_L}{2r}$$

构建 Rotation Dynamics 模型：

- 1 在 **system_layout** 模型的顶层，双击 Rotation 子系统以显示空子系统。删除 Inport 和 Outport 模块之间的连接。
- 2 对角速度和角度进行建模。添加一个 Integrator 模块。将初始条件设置保留为 0。此模块的输出是角度 θ ，输入是角速度 $\theta_{\dot{}}$ 。

- 3 根据切向速度计算角速度。添加带有参数 $1/(2*r)$ 的 Gain 模块。
- 4 连接这些模块。



- 5 查看模型的顶层。点击[向上导航到父级按钮](#) 。

对功能组件进行建模

通过一个函数从输入到输出的整个过程来描述功能。此描述可以包含代数方程和逻辑构造，您可以使用它们在 Simulink 中构建系统的图形化模型。

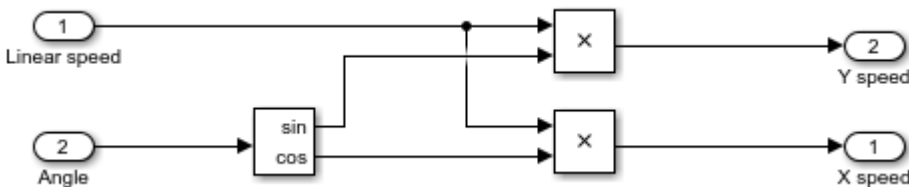
坐标变换

机器人在 X 坐标和 Y 坐标上的速度 V_X 和 V_Y 与线性速度 V_N 和角度 θ 相关：

$$V_X = V_N \cos(\theta)$$
$$V_Y = V_N \sin(\theta)$$

构建坐标变换模型：

- 1 在 `system_layout` 模型的顶层，双击 Coordinate Transform 子系统以显示空子系统。
- 2 对三角函数进行建模。从 Math Operations 库中添加一个 SinCos 模块。
- 3 对乘法进行建模。从 Math Operations 库中添加两个 Product 模块。
- 4 连接这些模块。



- 5 查看模型的顶层。点击[向上导航到父级按钮](#) 。

设置模型参数

模型参数值的来源可以是：

- 书面规范，如标准属性表或制造商的数据表
- 直接测量现有系统所得的测量值
- 基于系统输入/输出的估计值

此模型使用以下参数：

参数	符号	值
质量	m	2.5 kg

参数	符号	值
滚阻	<code>k_drag</code>	30 Ns ² /m
机器人半径	<code>r</code>	0.15 m

Simulink 使用 MATLAB 工作区来计算参数。可在 MATLAB 命令行窗口中设置这些参数：

```
m = 2.5;  
k_drag = 30;  
r = 0.15;
```

通过仿真来验证组件

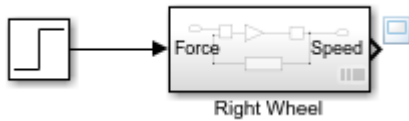
通过提供输入并观察输出来验证组件。即使这样简单的验证也能指出改进模型的直接方法。此示例验证以下行为：


- 当向轮子连续施加力时，速度会增加，直到达到稳定状态的速度为止。
- 当两个轮子向相反方向转动时，旋转角度以恒定速率增加。

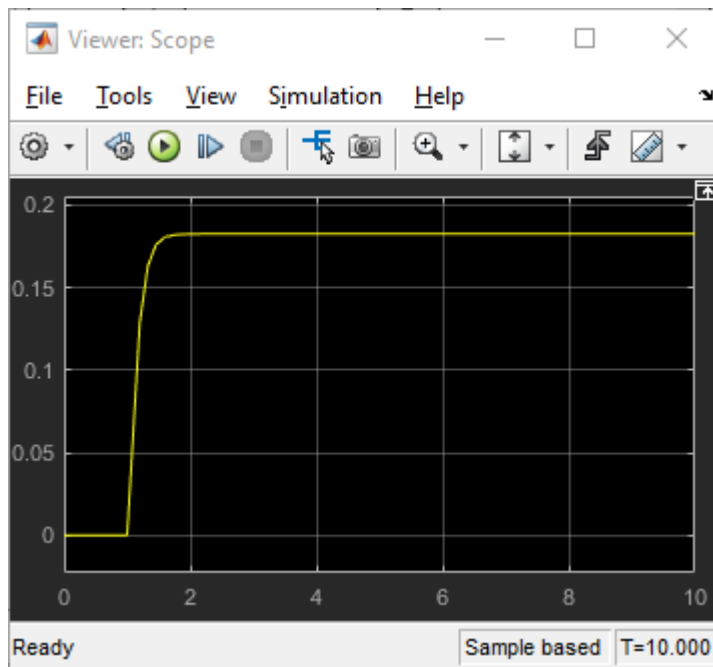
验证轮子组件

为轮子组件创建并运行测试模型：

- 1 创建一个新模型。在**仿真**选项卡中，点击**新建** 。将 Right Wheel 模块复制到新模型中。
- 2 创建一个测试输入。从 Sources 库中添加一个 Step 模块，并将其连接到 Right Wheel 模块的输入端口。将步长时间参数设置保留为 1。
- 3 在输出端口添加一个查看器。右键点击 Right Wheel 模块的输出端口，然后选择**创建并连接查看器** > **Simulink** > **Scope**。




- 4 运行仿真。在**仿真**选项卡中，点击**运行** 。



仿真结果表现出大致的预期行为。在步长时间处施加力之前，没有运动。施加力后，速度开始增加，当施加的力和阻力达到平衡后，速度将保持稳定。除验证外，此仿真还提供了在给定的作用力下轮子的最大速度信息。

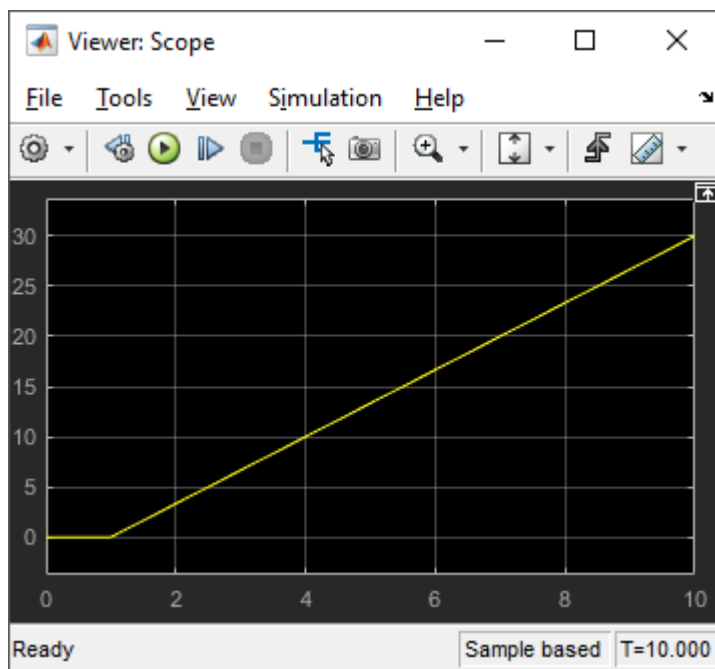
验证旋转组件

为旋转组件创建并运行测试模型：

- 1 创建一个新模型。点击 ，将 Rotation 模块复制到新模型中。
- 2 在新模型中创建测试输入。从 Sources 库中添加一个 Step 模块。将步长时间参数设置保留为 1。将其连接到 Rotation 模块的输入端口。此输入表示当两个轮子沿相反方向旋转时的轮子速度之差。
- 3 在输出端口添加一个查看器。右键点击 Rotation 模块的输出端口，然后选择**创建并连接查看器 > Simulink > Scope**。



- 4 运行仿真。在**仿真**选项卡中，点击**运行** 。



此仿真显示，当两个轮子以相同速度向相反方向转动时，角度会稳定增加。您可以对模型进行一些改进，以便更容易解释角度输出，例如：

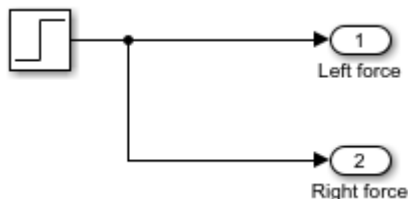
- 您可以将弧度输出转换为角度。添加增益为 $180/\pi$ 的 Gain 模块。
- 您可以按 360 度为一圈显示角度输出。添加带有函数 `mod` 的 Math Function 模块。

MATLAB 三角函数采用弧度输入。

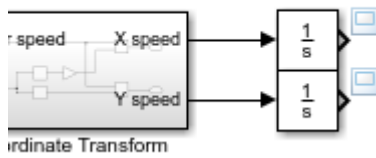
验证模型

验证单个组件后，您可以对整个模型进行类似的验证。此示例验证以下行为：

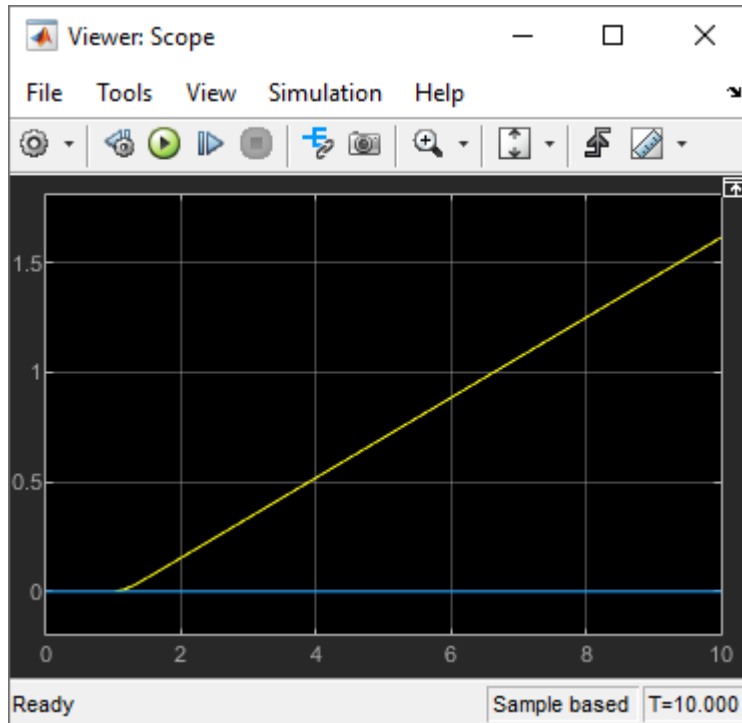
- 当沿相同方向对两个轮子施加相同的力时，机器人沿直线运动。
 - 当沿相反方向对两个轮子施加相同的力时，机器人原地旋转。
- 1 在 `system_layout` 模型中，双击 Inputs 子系统以显示空子系统。
 - 2 通过添加 Step 模块创建测试输入。将步长时间参数设置保留为 1。将其连接到两个 Output 模块上。



- 3 在模型的顶层，将两个输出信号连接到同一个波形查看器：

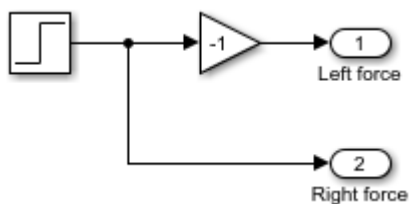


4 运行模型。



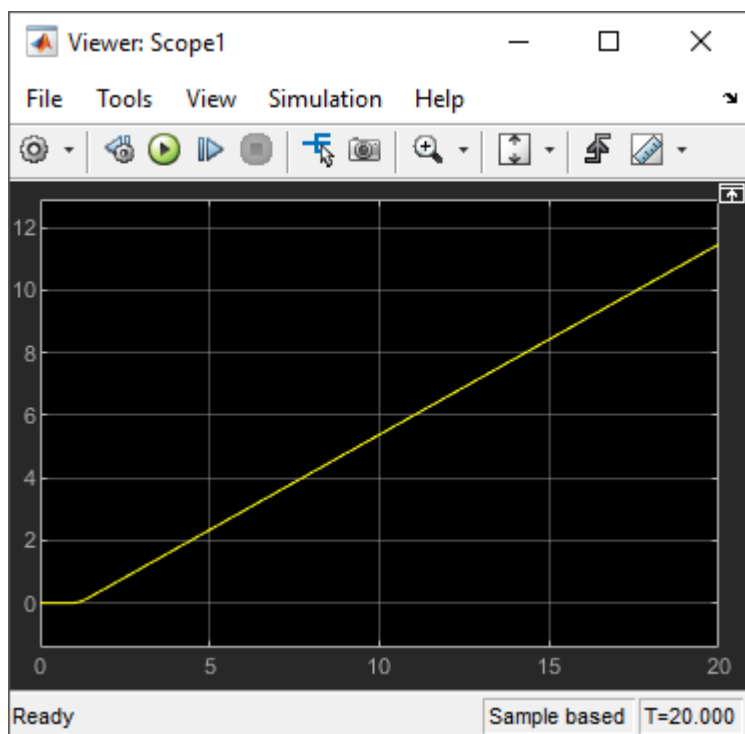
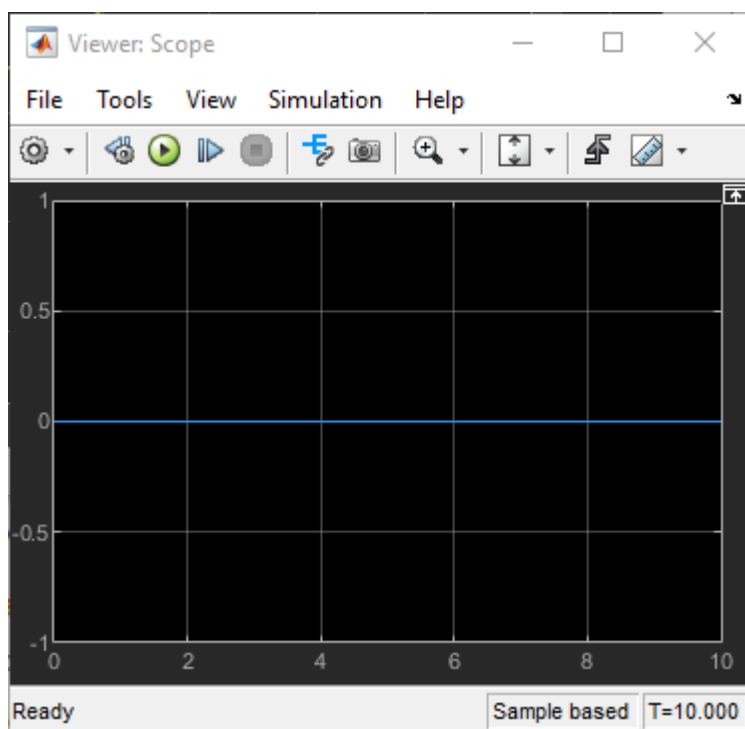
在上图中，黄线是 X 方向，蓝线是 Y 方向。由于角度为零并且保持不变，因此机器人只在 X 方向上移动，跟预期一样。

5 双击 Inputs 子系统，在信源和第二个输出之间添加具有参数 -1 的 Gain 模块。这将反转左轮的方向。



6 为角度输出添加一个示波器。

7 运行模型。



第一个视图显示 X-Y 平面上没有运动。第二个视图显示有稳定旋转。

通过更改输入，您可以使用这个最终模型来回答有关模型的许多问题。下面是一些示例：

- 当初始角度非零时会发生什么？

- 当作用力下降到零时，运动需要多长时间才能停止？
- 当机器人更重时会发生什么？
- 当机器人在更光滑的平面上移动时（即阻力系数更低时）会发生什么？

另请参阅

相关示例

- “系统定义和布局”（第 1-6 页）
- “在 Simulink 中设计系统”（第 1-22 页）

在 Simulink 中设计系统

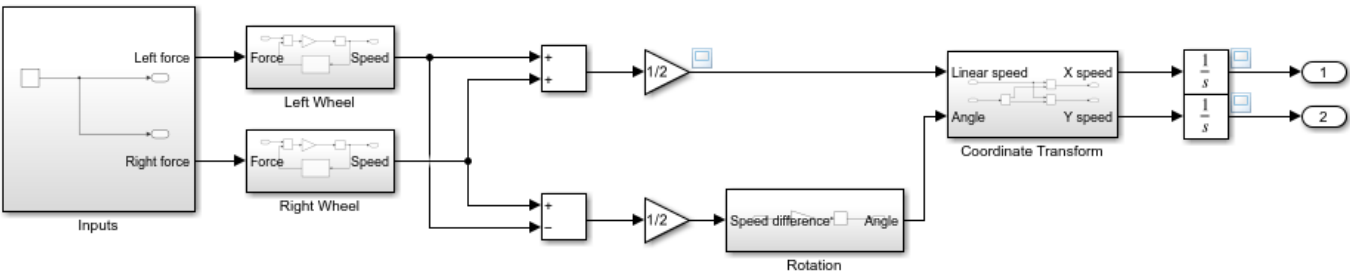
本节内容
“打开系统模型” （第 1-22 页）
“确定设计的组件和设计目的” （第 1-22 页）
“通过仿真来分析系统行为” （第 1-23 页）
“设计组件并验证设计” （第 1-25 页）

基于模型的设计范式以物理组件和系统的模型为中心，模型是设计、测试和实现活动的基础。本教程将在一个现有的系统模型中添加一个设计组件。

打开系统模型

此模型是一个平板机器人，它可以借助两个轮子进行移动或旋转，类似于家用扫地机器人。通过在 MATLAB® 命令行中输入以下命令打开模型：

```
open_system('system_model.slx')
```



Copyright 2018 The MathWorks, Inc.

本教程将分析此系统并为其添加功能。

确定设计的组件和设计目的

设定目标的说明是完成设计任务的关键第一步。即使是一个简单的系统，也可能存在多个甚至是相互矛盾的设计目的。对于示例模型，请考虑以下目标：

- 设计一个控制器，它可以改变作用力输入，使轮子按所需的速度转动。
- 设计输入，使设备沿预定的路径移动。
- 设计一个传感器和一个控制器，使设备沿直线移动。
- 设计一种规划算法，使设备沿可能的最短路径到达某个点并避开障碍物。
- 设计一个传感器和一种算法，使设备在某个区域内移动并避开障碍物。

本教程将设计一个警报系统。您需要确定用于测量障碍物距离的传感器的参数。完美的传感器精确地测量它与障碍物的距离。警报系统按固定时间间隔对这些测量值进行采样，使输出与测量值相差始终不超过 0.05 米。系统会及时生成警报，让机器人在碰到障碍物前停下来。

通过仿真来分析系统行为

设计新组件需要分析机器人的线性运动以确定：

- 如果切断轮子的电源，机器人能够以最高速度行驶多远
- 机器人的最高速度

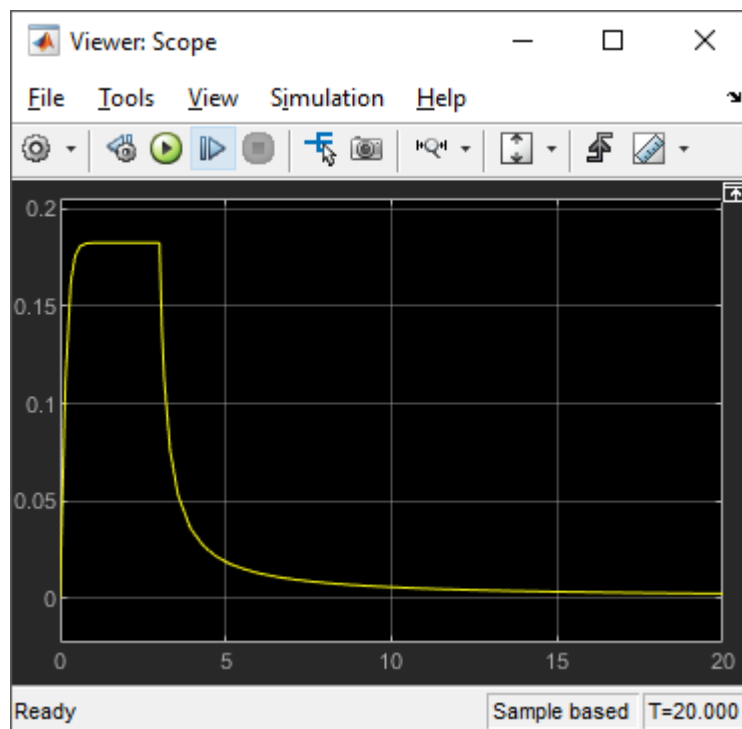
使用能够使机器人开始运动的作用力输入来运行模型，等到机器人达到稳定速度后，将输入作用力设置为零：

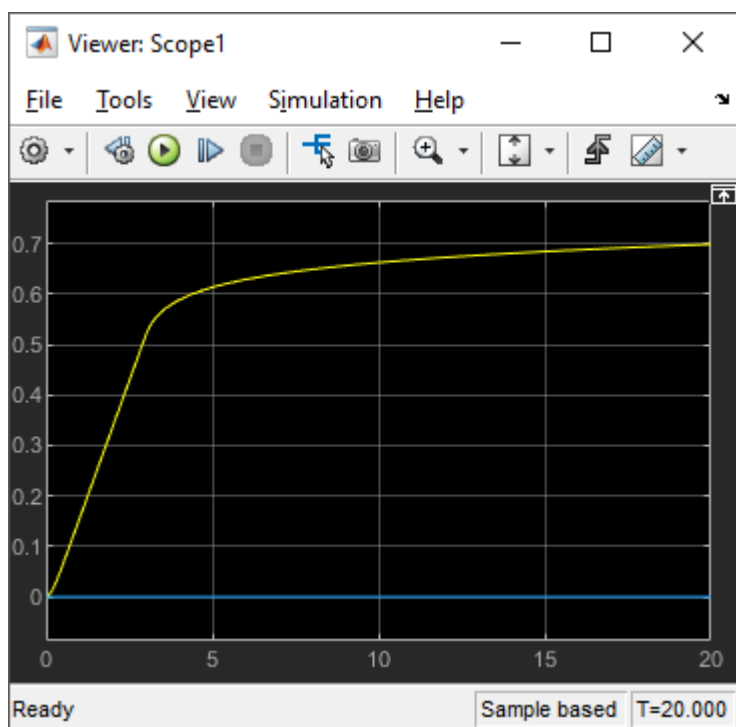
- 1 在模型中，双击 Inputs 子系统。
- 2 删除现有步长输入，并添加一个 Pulse Generator 模块。
- 3 设置 Pulse Generator 模块的参数：

- 幅值：1
- 周期：20
- 脉冲宽度：15

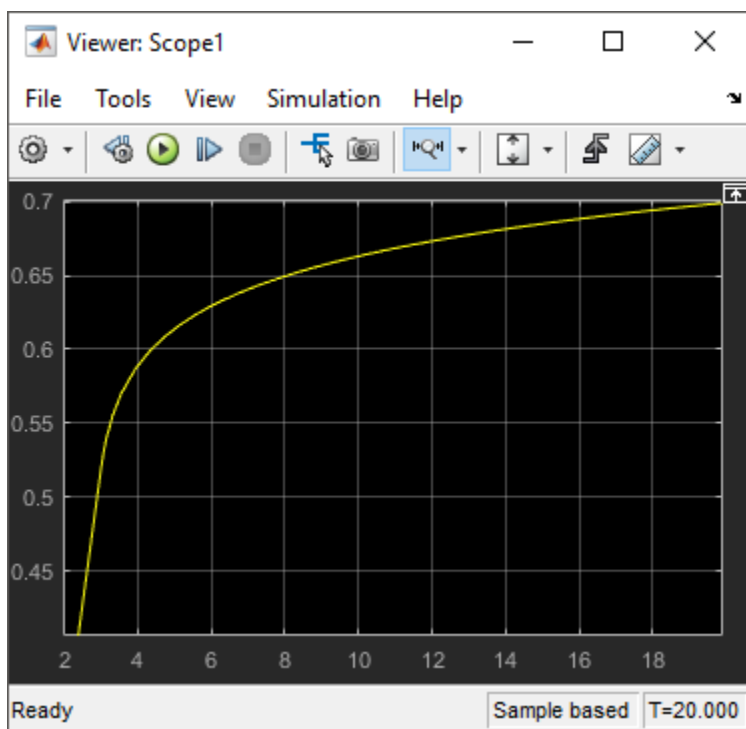
采用以上参数值是为了确保达到最高速度。您可以更改参数以查看其效果。

- 4 运行模型 20 秒。






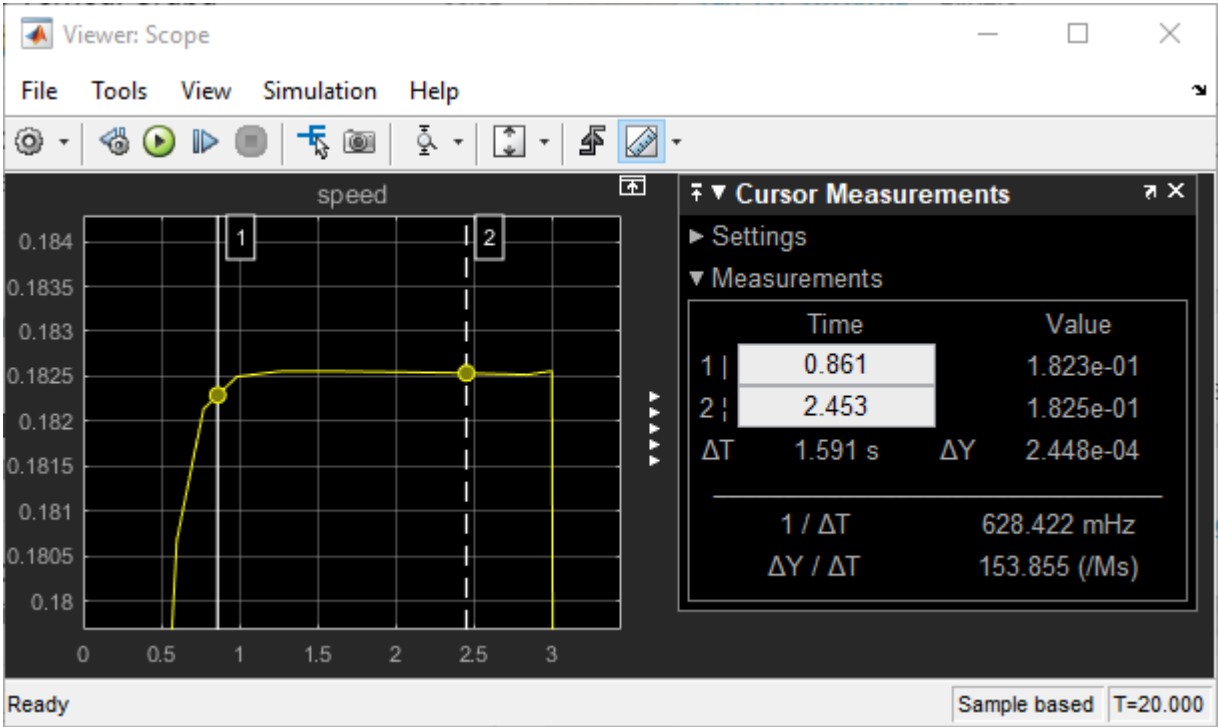
第一个示波器显示，在时间值为 3 时切断电源，此时速度迅速开始下降。然后速度逐渐接近零，但不会完全达到零。这是建模的局限性；无外力作用下的低速动态特性需要更复杂的表示。但针对本示例的目的，我们可以取近似值。放大位置信号。



在时间值为 3 时，机器人的位置大约在 0.55 米处。当仿真结束时，位置值小于 0.71 米。可以肯定地说，机器人在切断电源后行驶距离不到 0.16 米。

要求出最高速度，请执行以下操作：

- 1 放大速度输出在时间上的稳定区域，即从 1 秒到 3 秒。
- 2 再次点击缩放按钮，退出缩放模式。点击游标测量按钮 .
- 3 将第二个游标设置在速度曲线平坦的区域。



游标测量面板中的值列表明机器人的最高速度为 0.183 米/秒。要计算机器人行驶 0.05 米所需的时间，用 0.05 米除以 0.183 米/秒。您即可得到 0.27 秒。

设计组件并验证设计

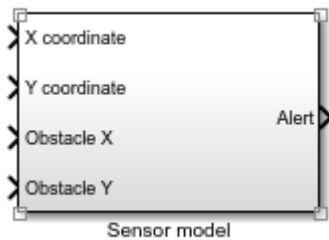
传感器设计包括以下组件：

- 机器人与障碍物之间的距离测量值 - 此示例假定测量值非常精确。
- 警报系统测量距离的时间间隔 - 要使测量误差低于 0.05 米，采样间隔必须小于 0.27 秒。采用 0.25 秒。
- 传感器发出警报时的距离 - 分析显示必须从距离障碍物 0.16 米处开始减速。但实际警报距离还必须将离散测量误差 0.05 米考虑在内。

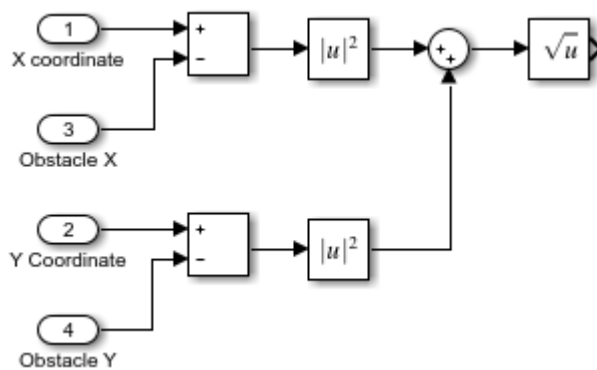
添加设计的组件

构建传感器：

- 1 创建具有如下端口的子系统。



- 构建距离测量值子系统。在 Sensor 模块中，按如下所示使用 Subtract、具有函数 magnitude^2 的 Math Function、Sum 和 Sqrt 模块。请注意输入端口的重新排序。

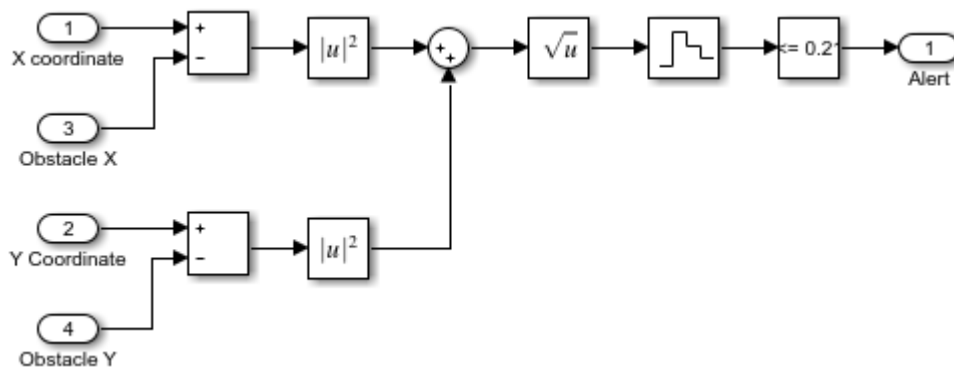


- 对采样进行建模。从 Discrete 库中将一个 Zero-Order Hold 模块添加到子系统中，并将其采样时间参数设置为 0.25。
- 对警报逻辑进行建模。从 Logic and Bit Operations 库中添加 Compare to Constant 模块，并设置参数：

- 运算符：<=
- 常量值：0.21
- 输出数据类型：boolean

此逻辑模块在输入小于或等于 0.21 时输出 1。

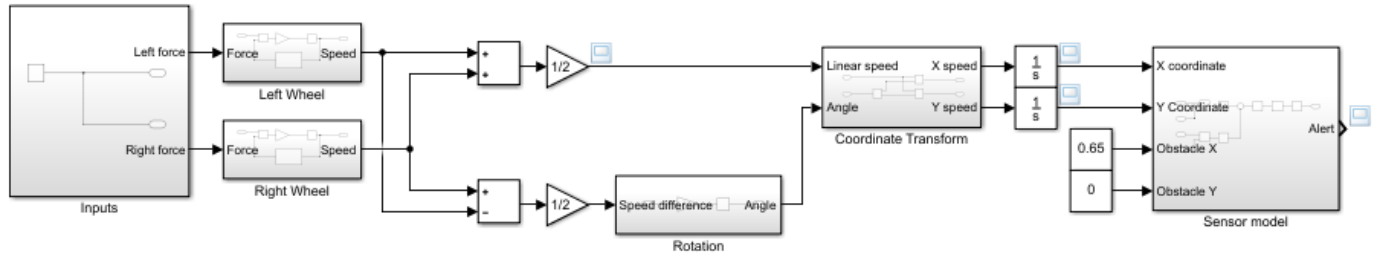
- 完成模块的连接。



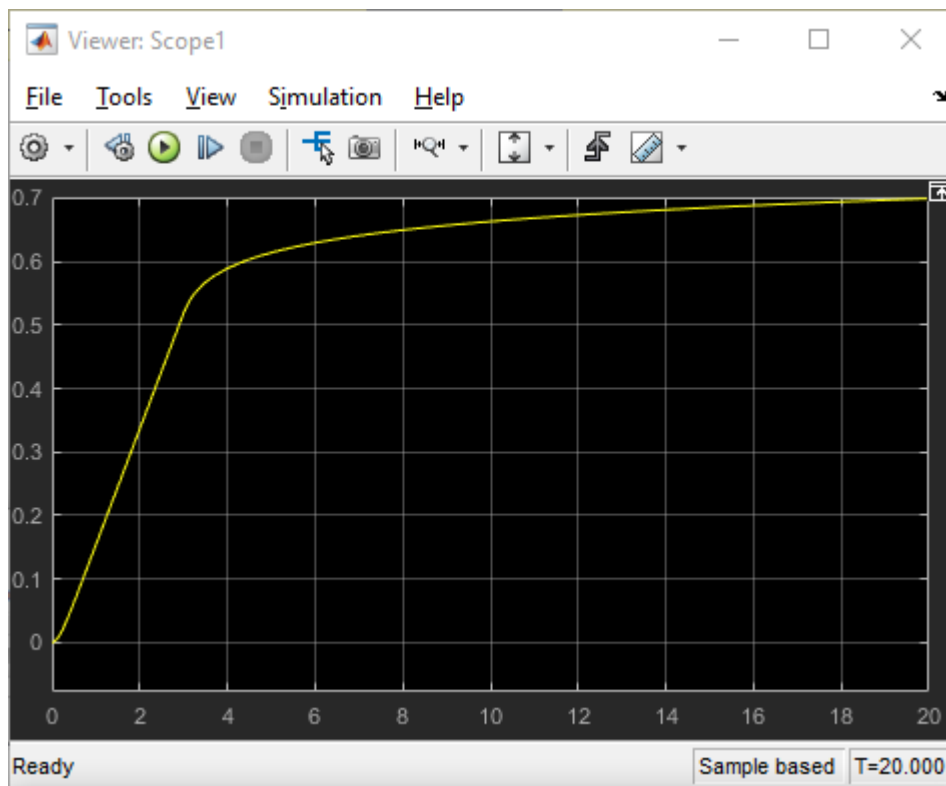
验证设计

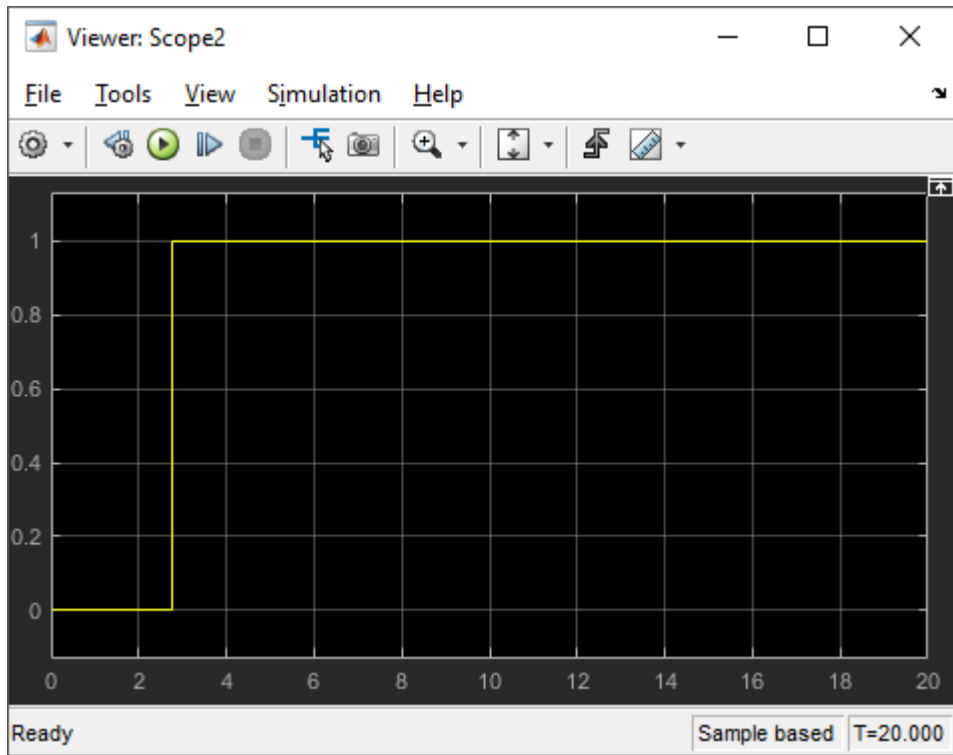
使用 Constant 模块作为 Sensor model 子系统的输入，以 $X = 0.65$, $Y = 0$ 的障碍物位置来测试设计。此测试验证 X 方向的设计功能。您可以针对不同路径创建类似的测试。此模型仅生成警报，不控制机器人。

- 1 设置障碍物位置。从 Sources 库中添加两个 Constant 模块，并将常量值设置为 **0.65** 和 **0**。将机器人的位置输出连接到传感器的输入端口。
- 2 为 Alert 输出添加一个示波器。



- 3 运行模型。





注意观察，一旦机器人位置与障碍物位置之间的距离在 0.21 米范围内并且满足此组件的设计要求，警报状态就变为 1。

对于具有复杂组件和严格要求的真实系统，Simulink 产品系列提供了一些附加工具，可以细化和自动完成设计过程。Requirements Toolbox™ 可以对要求进行形式化定义，并将它们与模型组件联系起来。如果您要为此机器人构建一个控制器，Simulink Control Design™ 可以帮助您完成设计。Simulink Verification and Validation™ 产品则建立了一套严格的框架，可用于测试组件和系统。

另请参阅

相关示例

- “使用 Simulink 进行基于模型的设计” (第 1-3 页)
- “系统定义和布局” (第 1-6 页)
- “建模并验证系统” (第 1-12 页)

在 Simulink 中进行建模

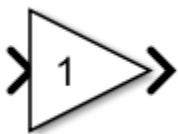
Simulink 模块图

Simulink 是动态系统的图形建模和仿真环境。您可以创建模块图，用模块表示系统的各个组成部分。模块可以表示物理组件、小型系统或函数。输入/输出关系则完整描述了模块特征。请思考下面这些示例：

- 一个水龙头往一个水桶里注入水 - 水以一定的流速进入水桶，水桶变重。模块可以表示水桶，水的流速为输入，水桶的重量为输出。
- 您用扩音器传递声音 - 扩音器一端产生的声音在另一端被放大。扩音器是模块，输入是声源的声波，输出是您听到的声波。
- 您推动购物车使它移动 - 购物车是模块，您施加的力是输入，购物车的位置是输出。

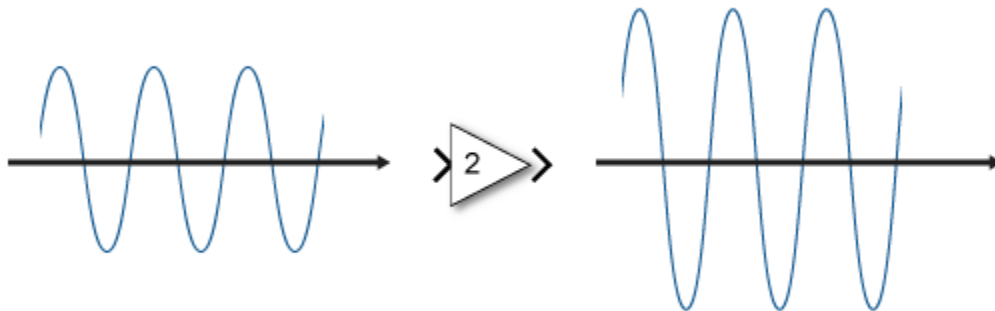
只有定义了输入和输出，模块的定义才算完成，并且此模型定义任务需与建模目的相关。例如，如果建模目的不涉及购物车的位置，则会自然选择购物车的速度作为输出。

Simulink 提供了一些模块库，它们是按功能分组的模块集合。例如，要对以常量倍数放大输入的扩音器进行建模，可以使用 Math Operations 库中的 Gain 模块。



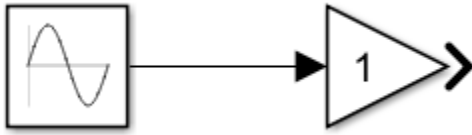
进入扩音器的声波作为输入，出来的同一声波的更大版本作为输出。

> 符号表示模块的输入和输出，可以连接到其他模块。

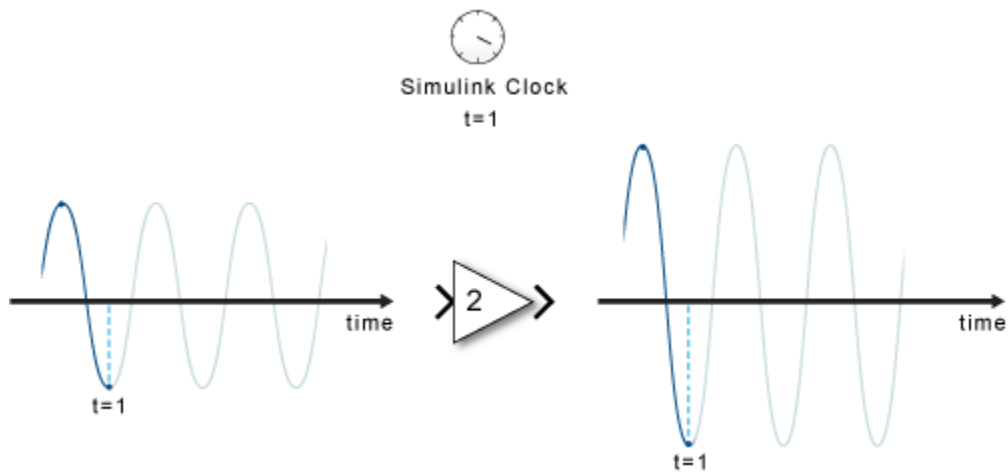


您可以将模块连接到其他模块以构成系统，从而表示更复杂的功能。例如，音频播放器可将数字文件转换为声音。软件从存储中读取数字表示，以数学方式对其进行解释，然后将其变为物理声音。处理数字文件以计算声音波形的软件可以是一个模块，接收波形并将其转换为声音的扬声器可以是另一个模块。生成输入的组件又是另一个模块。

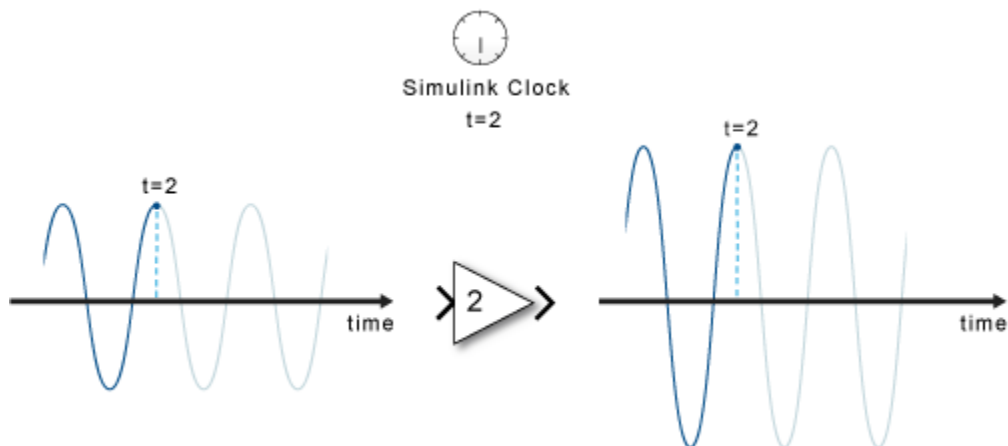
要在 Simulink 中对扩音器的正弦波输入进行建模，需要包含 Sine Wave 源。



Simulink 的主要功能是对系统各个组件随时间流逝的行为变化进行仿真。简单来讲就是：采用一个时钟，按时间确定各个模块的仿真顺序，并在仿真过程中依次将在上一个模块图中计算得出的输出传播到下一个模块，直至最后一个模块。以扩音器为例。在每个时间步，Simulink 都必须计算正弦波的值，将其传播给扩音器，然后计算输出值。



在每个时间步，每个模块都要根据输入计算输出。当在一个给定时间步计算完图中的所有信号后，Simulink 将基于模型配置和数值求解器算法确定下一个时间步，并向前移动仿真时钟。接下来，每个模块将为这个新的时间步计算输出。



在仿真中，时间的移动与真实时钟不同。完成每个时间步的计算需要多长时间，该时间步就会花费多长时间，而不管它代表几分之一秒还是几年。

通常，组件的输入对其输出的影响不是瞬时的。例如，打开加热器不会导致温度立即发生变化。该动作作为微分方程提供输入。历史温度（一个状态）也是一个输入因子。当仿真需要求解微分方程或差分方程时，Simulink 使用内存和数值求解器来计算时间步的状态值。

Simulink 处理三类数据：

- 信号 - 在仿真期间计算的模块输入和输出
- 状态 - 在仿真期间中计算的模块动态的内部值
- 参数 - 影响模块行为的值，由用户控制

在每个时间步，Simulink 都计算信号和状态的新值。相比之下，您可以在编译模型时指定参数，并且可以在仿真运行时偶尔更改它们。

另请参阅

相关示例

- “创建简单模型” （第 3-2 页）
- “使用 Simulink 进行基于模型的设计” （第 1-3 页）

简单的 Simulink 模型

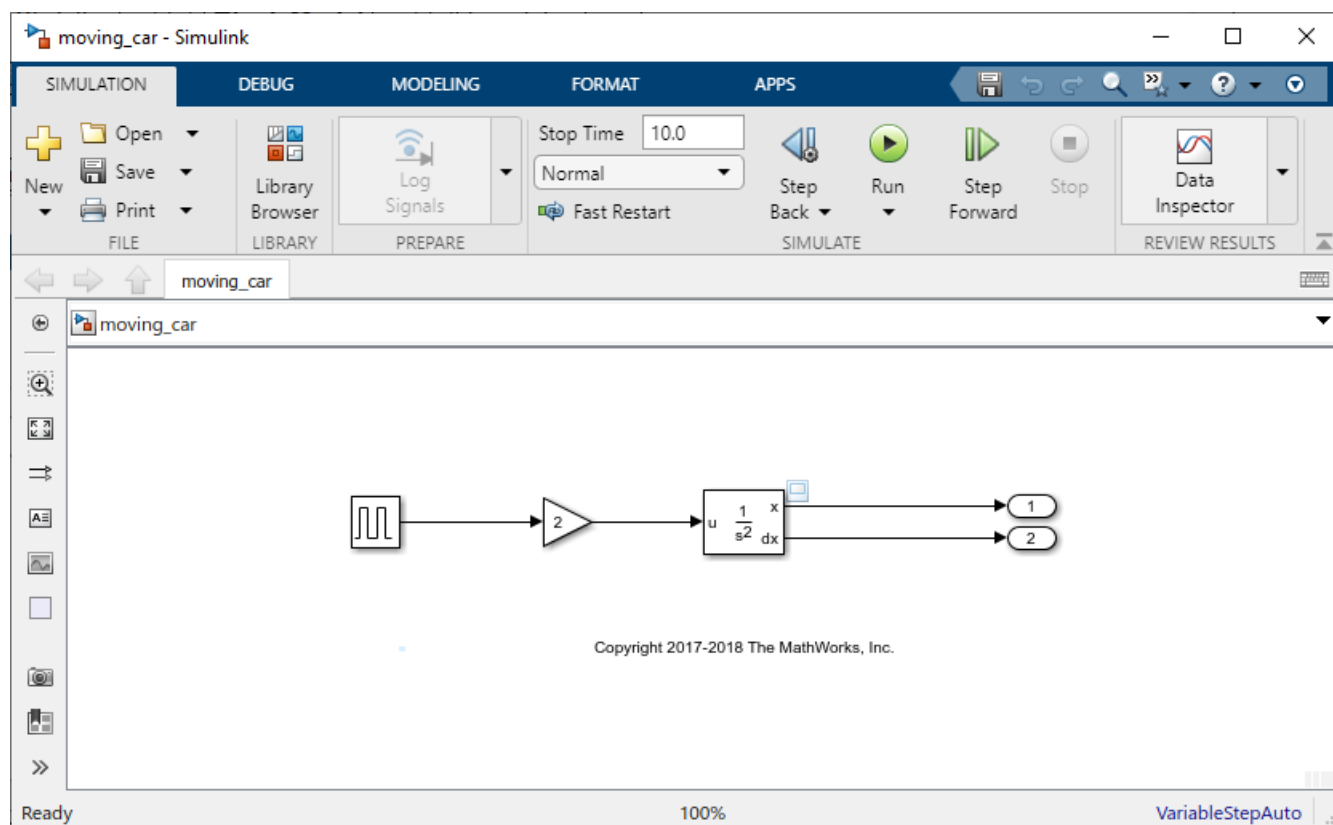
创建简单模型

本节内容
“打开新模型” （第 3-3 页）
“打开 Simulink 库浏览器” （第 3-5 页）
“将模块添加到模型” （第 3-7 页）
“连接模块” （第 3-7 页）
“添加信号查看器” （第 3-8 页）
“运行仿真” （第 3-8 页）
“细化模型” （第 3-9 页）

您可以使用 Simulink 来对系统建模，然后仿真该系统的动态行为。本教程中创建简单模型所使用的基本方法也适用于创建更复杂的模型。此示例对简化的汽车运动进行仿真。当踩下油门踏板时，汽车通常处于行进状态。松开踏板后，汽车怠速并停下来。

Simulink 模块是定义模块输入和模块输出之间数学关系的模型元素。要创建这个简单模型，您需要四个 Simulink 模块。


模块名称	模块目的	模型目的
Pulse Generator	为模型生成输入信号	表示加速踏板
Gain	将输入信号乘以常量值	计算踩下加速踏板后如何影响汽车的加速度
Integrator, Second-Order	将输入信号积分两次	根据加速度计算汽车位置
Outport	指定一个信号作为模型的输出	指定汽车位置做为模型的输出



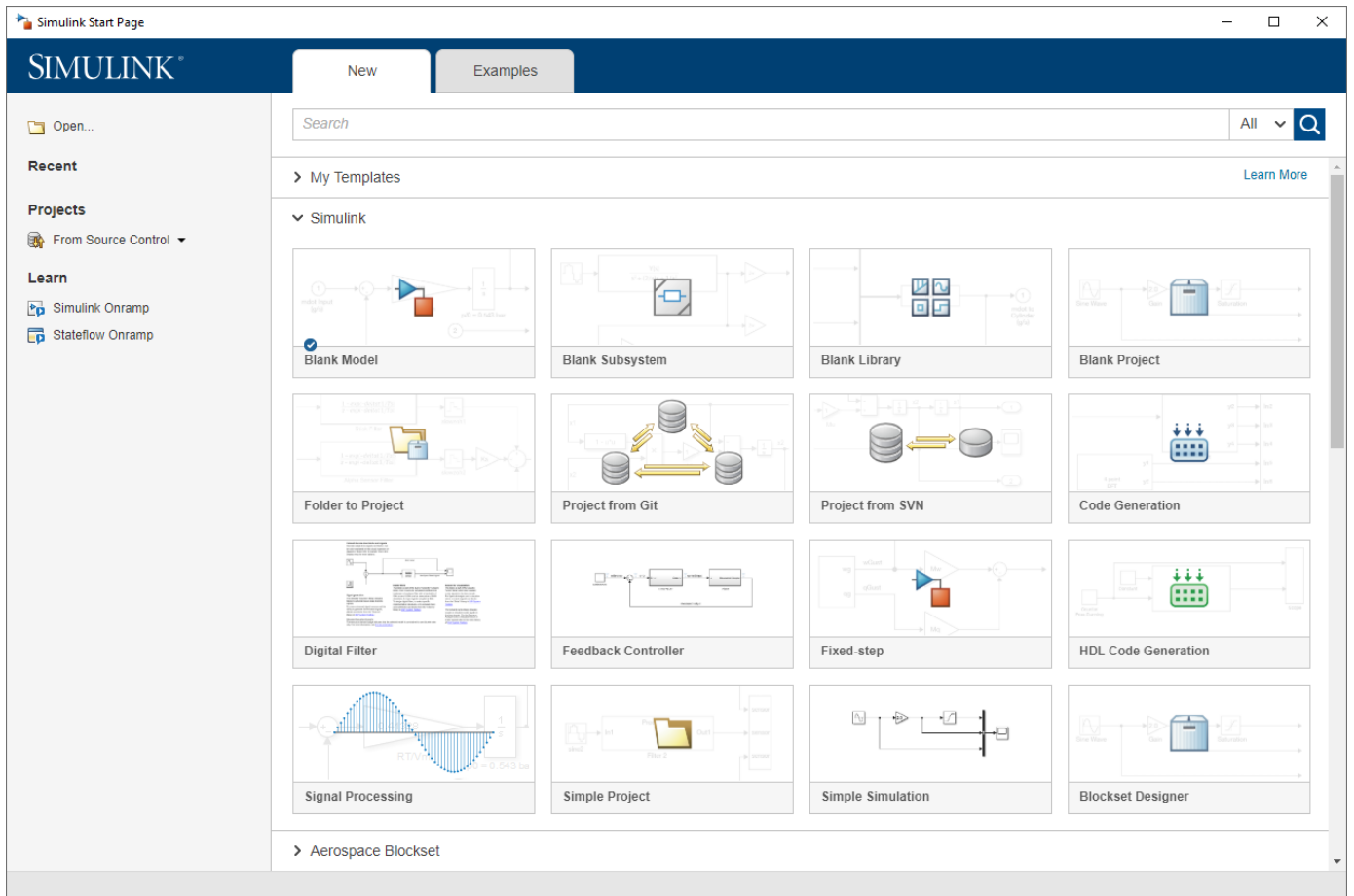
此模型的仿真过程是将一个简短的脉冲信号积分两次，形成一个斜坡。结果显示在一个示波器窗口中。输入脉冲表示是否踩下油门踏板 - 1 表示踩下，0 表示未踩下。输出斜坡表示与起点之间的距离增加。

打开新模型

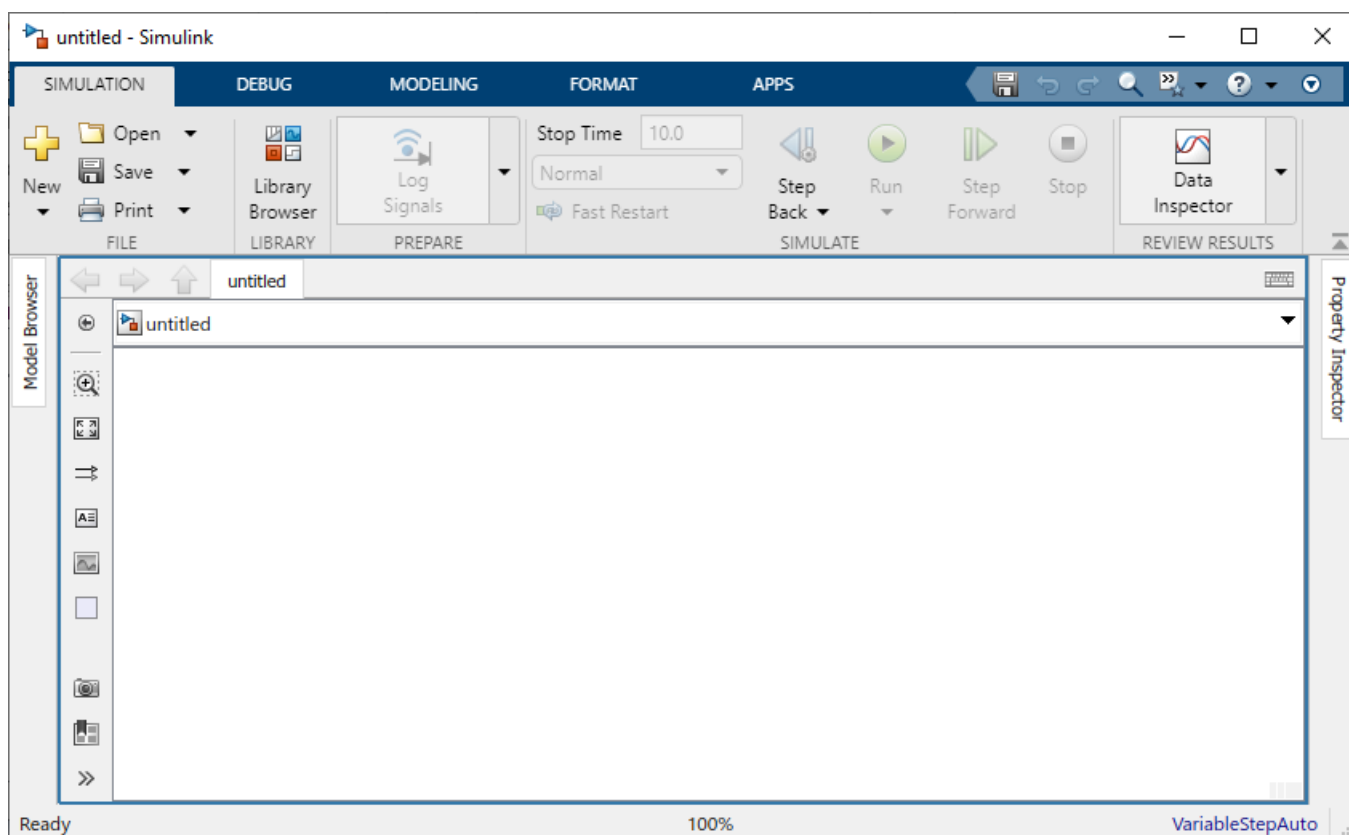
使用 Simulink 编辑器构建模型。

- 1 启动 MATLAB。在 MATLAB 工具条上，点击 **Simulink** 按钮 。

3 简单的 Simulink 模型



- 2 点击空白模型模板。
Simulink 编辑器打开。




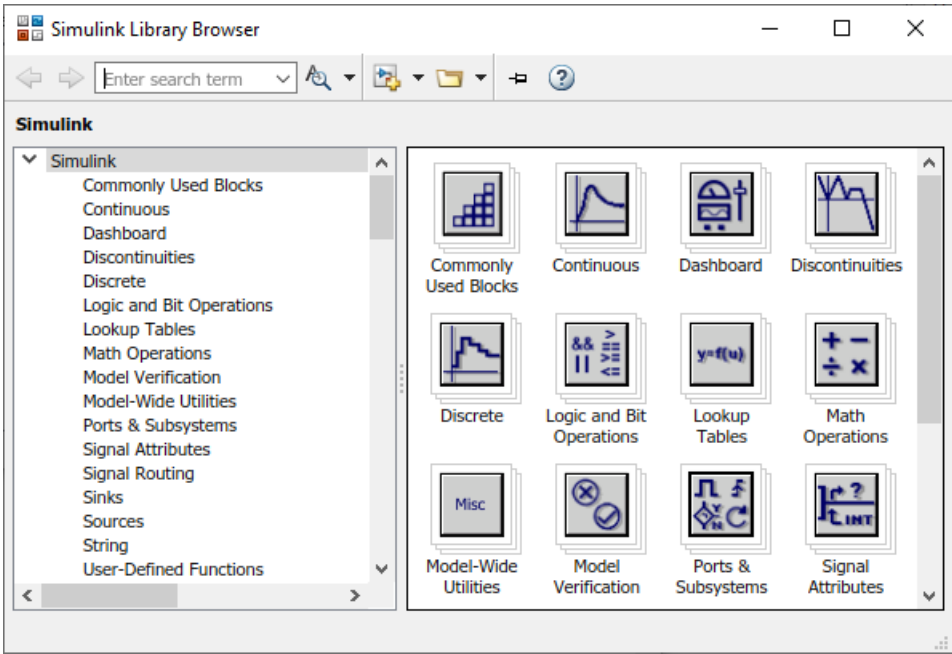
- 3 从**仿真**选项卡中，选择**保存 > 另存为**。在**文件名**文本框中，输入您的模型的名称。例如，`simple_model`。点击**保存**。模型使用文件扩展名 `.slx` 进行保存。

打开 Simulink 库浏览器

Simulink 在库浏览器中提供了一系列按功能分类的模块库。下面是大多数工作流常用的一些模块库：

- Continuous - 表示具有连续状态的系统的模块
- Discrete - 表示具有离散状态的系统的模块
- Math Operations - 实现代数和逻辑方程的模块
- Sinks - 存储并显示所连接信号的模块
- Sources - 生成模型的驱动信号值的模块

- 1 在**仿真**选项卡中，点击**库浏览器**按钮 。

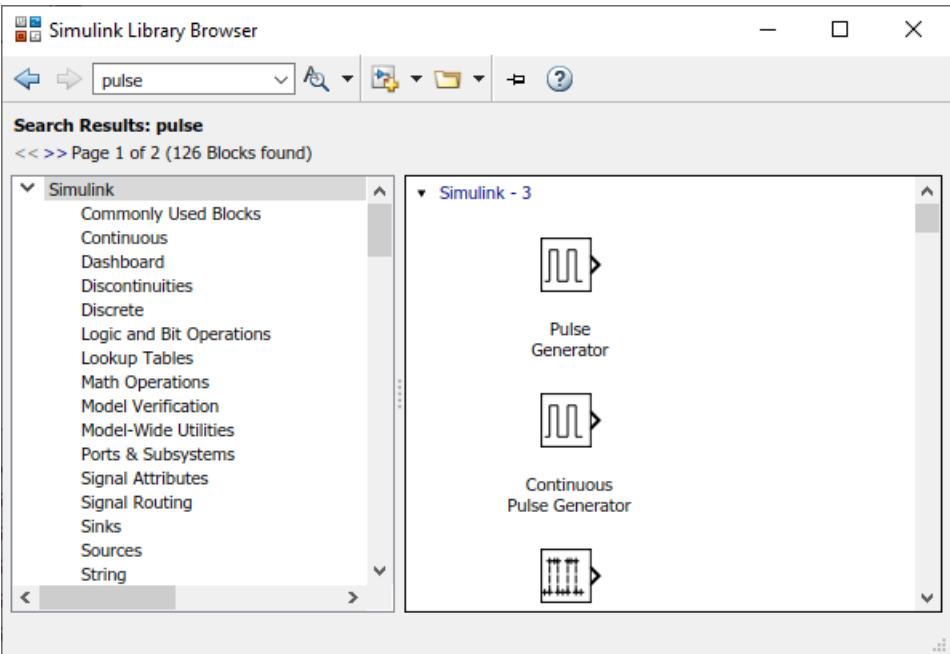


2 将库浏览器设置为始终在其他桌面窗口前端。在 Simulink 库浏览器工具栏上，选择**保持在前端**按钮



要浏览模块库，请从左窗格中选择一个类别，然后选择一个功能区。要搜索所有可用的模块库，请输入搜索词。

例如，查找 Pulse Generator 模块。在浏览器工具栏的搜索框中输入 **pulse**，然后按 Enter。Simulink 将在模块库中搜索名称或说明中包含 **pulse** 的模块，然后显示这些模块。



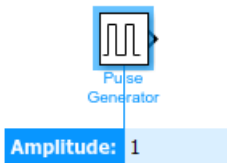
获取模块的详细信息。右键单击 Pulse Generator 模块，然后选择 **Pulse Generator 模块的帮助**。帮助浏览器随即打开并显示该模块的参考页。

模块通常有几个参数。您可以通过双击该模块来访问所有模块参数。

将模块添加到模型

要开始构建模型，请浏览库并添加模块。

- 1 从 “Sources” 库中，将 Pulse Generator 模块拖到 Simulink 编辑器中。您的模型中将出现 Pulse Generator 模块的副本，还有一个文本框用于输入**振幅**参数的值。输入 1。



参数值在整个仿真过程中保持不变。

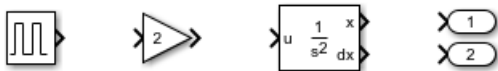
- 2 使用相同的方法将以下模块添加到模型中。

模块	库	参数
Gain	“Simulink/Math Operations”	增益: 2
Integrator, Second-Order	“Simulink/Continuous”	初始条件: 0
Outport	“Simulink/Sinks”	端口号: 1

复制现有 Outport 模块，然后使用键盘快捷方式将其粘贴到另一个点，从而添加第二个 Outport 模块。

您的模型现在已经包含您需要的模块。

- 3 通过点击并拖动每个模块来排列模块。要调整模块大小，请拖动一个角。



连接模块

通过在输出端口和输入端口之间创建线条来连接模块。

- 1 点击 Pulse Generator 模块右侧的输出端口。

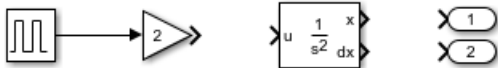
该输出端口和所有适合连接的输入端口由蓝色向前符号 ➤ 指示。



- 2 指向 ➤ 以查看连接提示。



点击此提示。Simulink 用线条连接模块，并用箭头表示信号流的方向。



- 3 将 Gain block 的输出端口连接到 Integrator, Second-Order 模块的输入端口。
- 4 将 Integrator, Second-Order 模块的两个输出连接到两个 Outport 模块。
- 5 保存模型。在**仿真**选项卡中，点击**保存**。



添加信号查看器

要查看仿真结果，请将第一个输出连接到一个 Signal Viewer。

点击信号。在**仿真**选项卡中的**准备**下，点击**添加查看器**。选择 **Scope**。信号上会出现查看器图标，并打开一个示波器窗口。

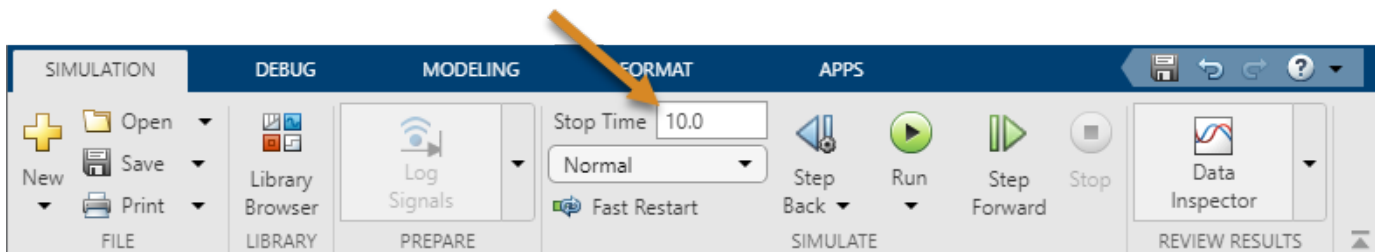


您可以随时通过双击该图标打开示波器。


运行仿真

定义配置参数后，即可进行模型仿真。

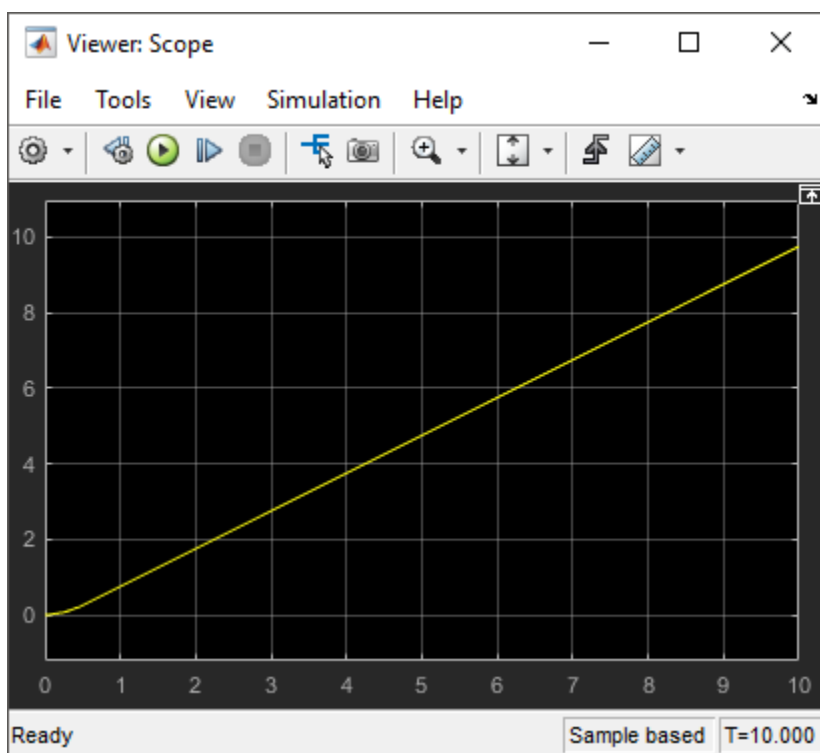
- 1 在**仿真**选项卡中，通过更改工具条中的值来设置仿真停止时间。



默认停止时间 10.0 适合此模型。此时间值没有单位。Simulink 中的时间单位取决于方程的构造方式。此示例对简化的汽车运动进行 10 秒的仿真 - 其他模型的时间单位可以是毫秒或年。

- 2 要运行仿真，请点击**运行**按钮 .

仿真开始运行并在查看器中生成输出。



细化模型

此示例使用现有模型 `moving_car.slx`，并基于此运动模型对接近传感器建模。在这种情况下，数字传感器用于测量汽车与 10 米（30 英尺）外的障碍物之间的距离。模型基于下列条件来输出传感器的测量值和汽车的位置值：

- 汽车在到达障碍物时会紧急刹车。
- 在现实世界中，传感器对距离的测量不够精确，从而导致随机数值误差。
- 数字传感器以固定时间间隔运行。

更改模块参数

要开始，请打开 `moving_car` 模型。在 MATLAB 命令行中，输入：

```
open_system('moving_car.slx')
```



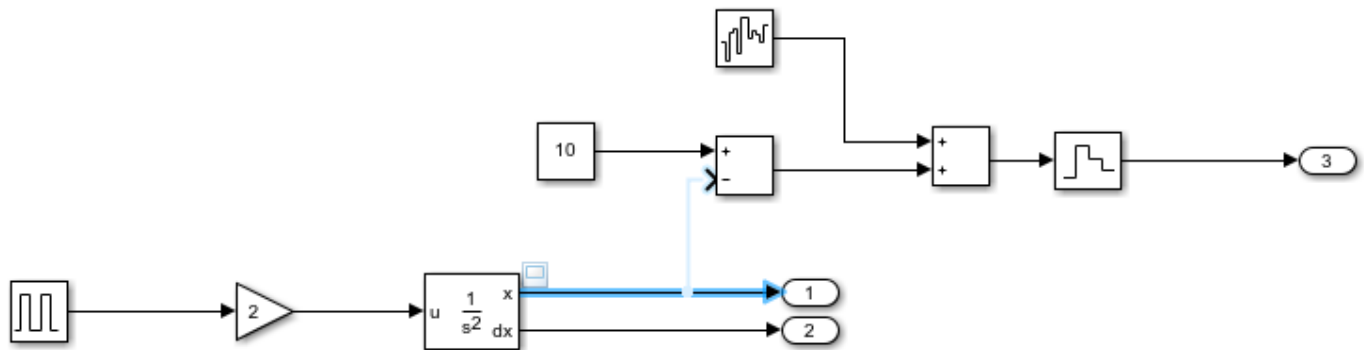
您首先需要对在汽车位置到达 10 时的紧急刹车进行建模。Integrator, Second-Order 模块有用于此目的的参数。

- 1 双击 Integrator, Second-Order 模块。将出现“模块参数”对话框。
- 2 选择 **x 限制**，然后为 **x 上限**输入 10。参数的背景色发生变化以指示模型存在未应用的修改。点击**确定**应用更改并关闭对话框。

添加新模块和连接

添加一个用来测量与障碍物之间距离的传感器。

- 1 修改模型。根据需要展开模型窗口，以容纳新模块。
 - 求实际距离。要想求出障碍物位置和车辆位置之间的距离，需要从“**Math Operations**”库中添加 Subtract 模块。还要从“**Sources**”库中添加 Constant 模块来为障碍物的位置设置常量值 10。
 - 对真实传感器中常见的不完美测量进行建模。使用“**Sources**”库中的 Band-Limited White Noise 模块产生噪声。将**噪声功率**参数设置为 0.001。通过使用“**Math Operations**”库中的 Add 模块将噪声添加到测量中。
 - 对每 0.1 秒触发一次的数字传感器进行建模。在 Simulink 中，以给定时间间隔对信号进行采样需要一个样本和保持器。从“**Discrete**”库中添加 Zero-Order Hold 模块。将该模块添加到模型后，将**采样时间**参数更改为 0.1。
 - 添加另一个 Output，用来连接传感器输出。保留**端口号**参数的默认值。
- 2 连接新模块。Integrator, Second-Order 模块的输出已连接到另一个端口。要在该信号中创建分支，请左键点击该信号以突出显示可供连接的端口，然后点击适当的端口。



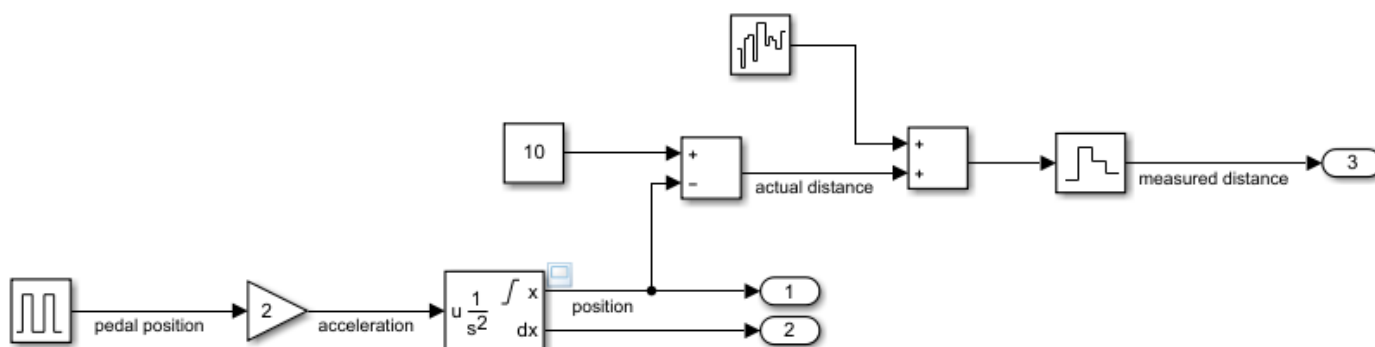
为信号添加注释

将信号名称添加到模型中。

- 1 双击信号并键入信号名称。



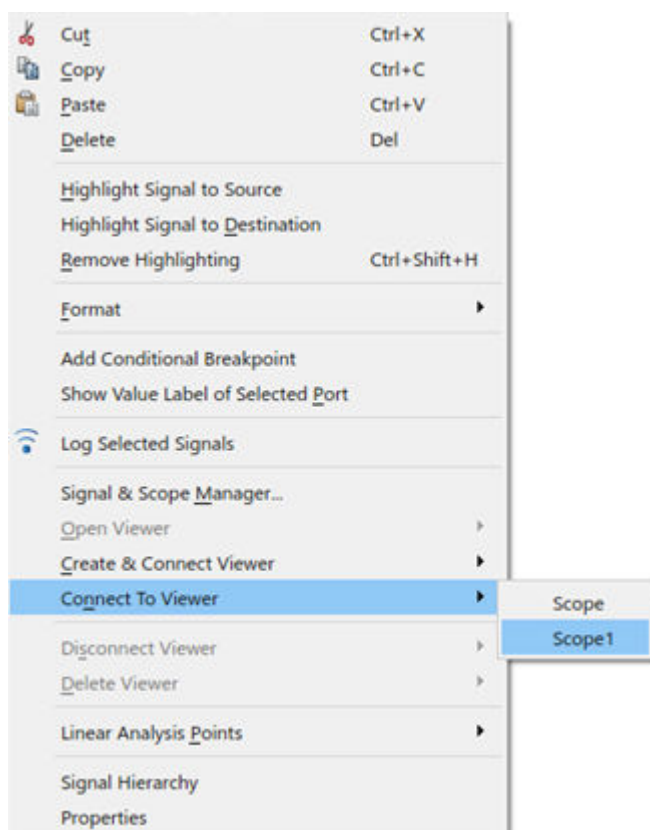
- 2 要完成输入，请点击文本框外部。
- 3 重复上述步骤以添加下图中所示的名称。



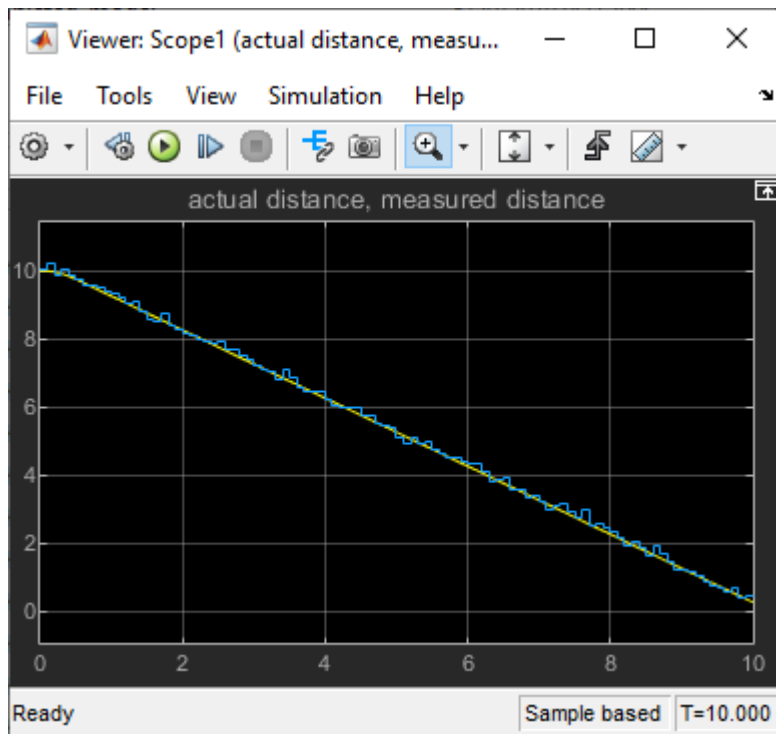
比较多个信号

将 **actual distance** 信号与 **measured distance** 信号进行比较。

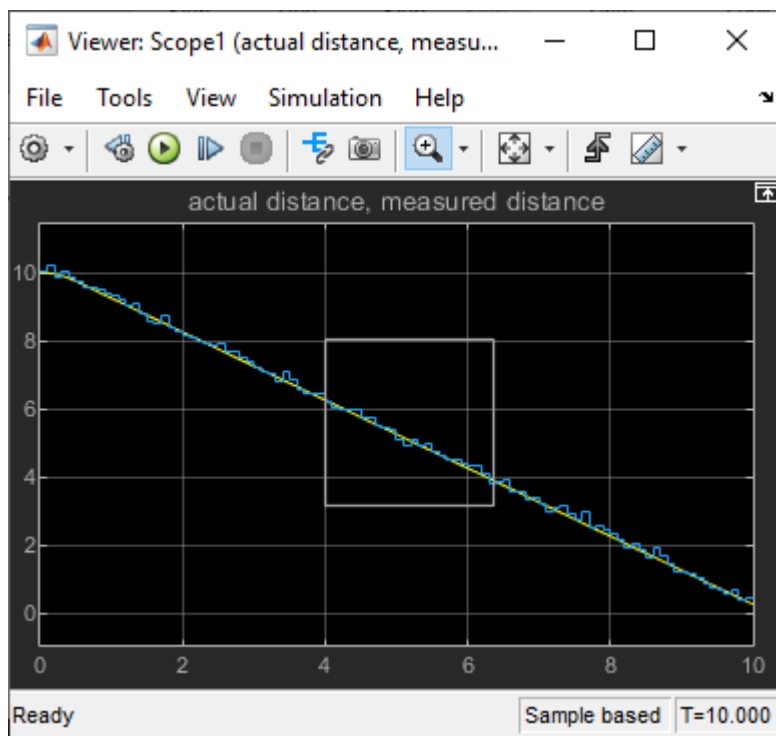
- 1 创建 Scope Viewer 并将其连接到 **actual distance** 信号。右键点击信号并选择**创建并连接查看器** > **Simulink** > **Scope**。信号的名称显示在查看器标题中。
- 2 将 **measured distance** 信号添加到同一个查看器中。右键点击信号，然后选择**连接到查看器** > **Scope1**。确保您连接到在上一步中创建的查看器。



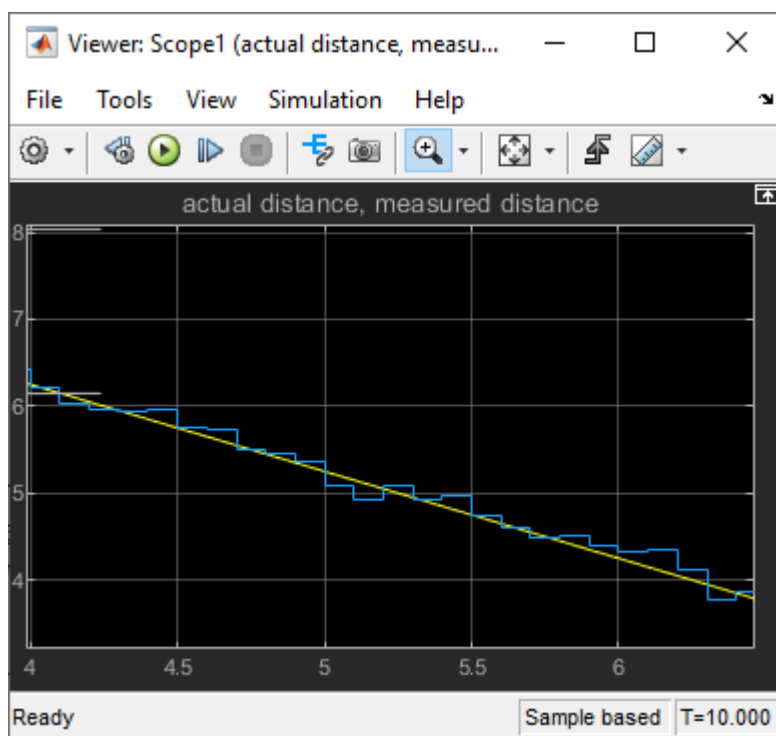
- 3 运行模型。查看器显示两个信号：**actual distance**（黄色）和 **measured distance**（蓝色）。



- 4 放大图形以观察噪声和采样的影响。点击**缩放**按钮 。左键点击并拖动鼠标框住您想放大查看的区域。



您可以反复放大以观察细节。



从图中可以看到，测量值可偏离实际值达 0.3 米之多。此信息在设计安全功能（例如碰撞警告）时非常有用。

另请参阅

模块

Pulse Generator | Gain | Second-Order Integrator | Add | Constant | Zero-Order Hold | Band-Limited White Noise

相关示例

- “建模并验证系统”（第 1-12 页）

导航 Simulink 模型

探索模型层次结构

本节内容
“查看模型层次结构” （第 4-2 页）
“查看信号属性” （第 4-3 页）
“跟踪信号” （第 4-5 页）

Simulink 模型可以组织成具有层次结构的组件。在分层模型中，您可以选择查看整体系统，或沿模型层次结构向下导航以逐级深入模型细节。

查看模型层次结构

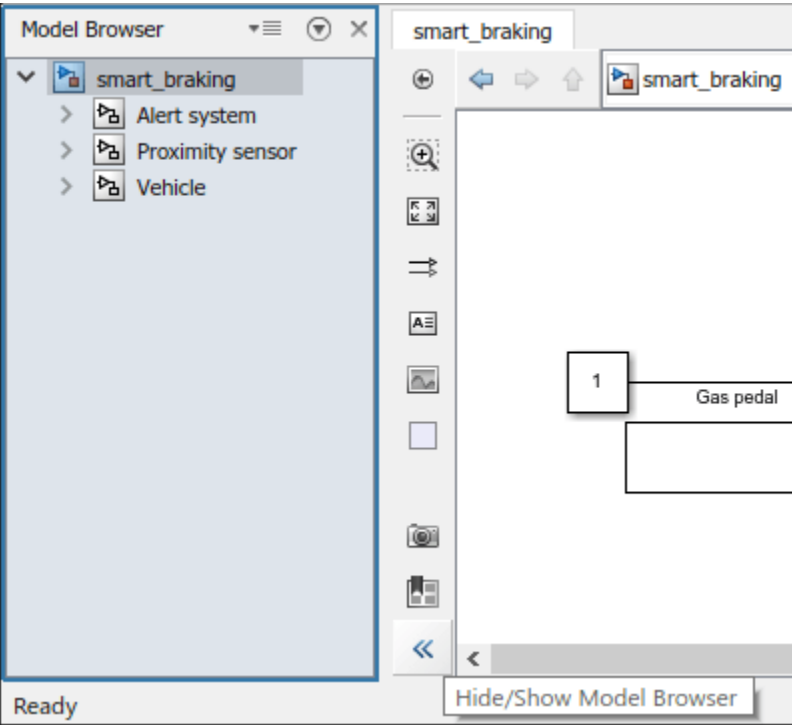
要开始导航，请打开 `smart_braking` 模型。

在模型中：

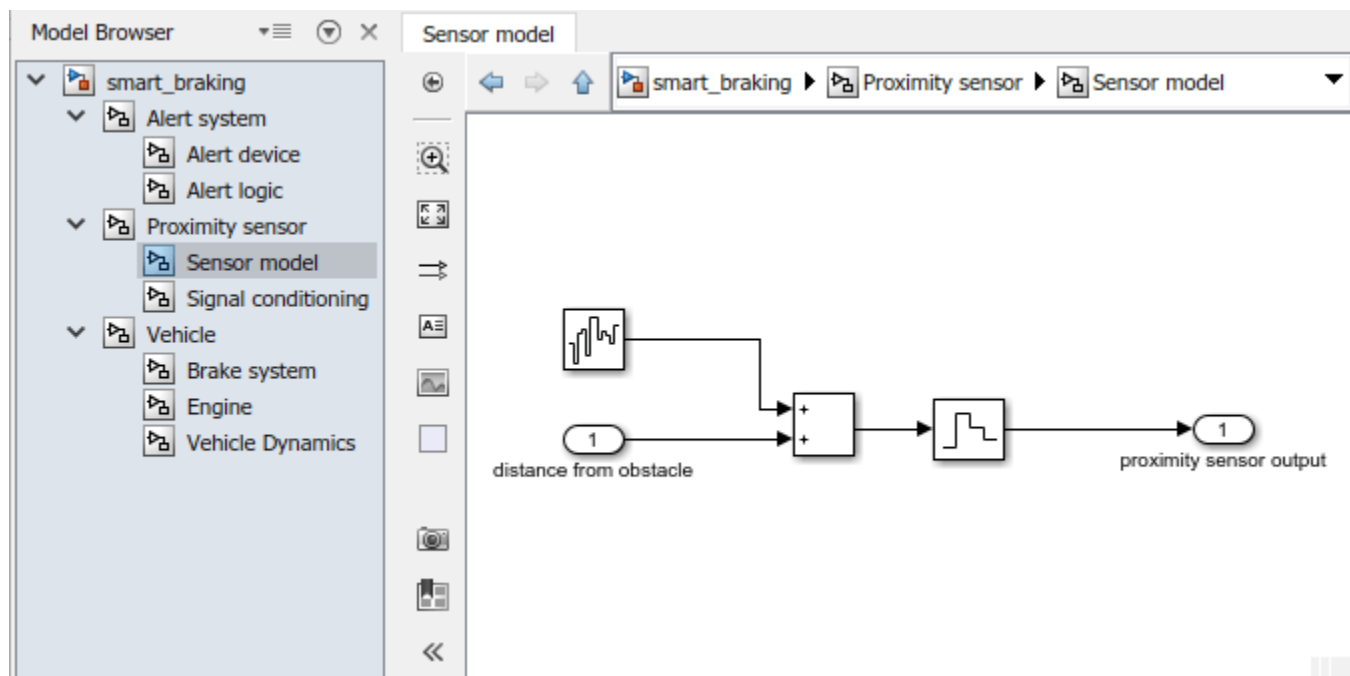
- 当踩下油门踏板时，汽车移动。
- 接近传感器将测量车辆和障碍物之间的距离。
- 警报系统基于该接近度生成警报。
- 警报系统自动控制刹车以防止碰撞。

当您构建模型时，您可以将模块连接起来，以对表示系统动态的复杂组件进行建模。在此模型中，Vehicle、Proximity sensor 和 Alert system 都是包含多个模块的复杂组件，它们都存在于一个子系统层次结构中。要查看子系统的内容，请双击该子系统。

要查看完整模型层次结构的表示，请点击模型窗口左下角的**隐藏/显示模型浏览器** 按钮。



模型浏览器显示，您在顶层查看的所有子系统都有自己的子系统。展开每个子系统节点以查看它包含的子系统。您可以在模型浏览器中导航层次结构。例如，展开 Proximity sensor 节点，然后选择 Sensor model 子系统。



地址栏会显示您正在查看哪个子系统。要在单独的窗口中打开子系统，请右键单击该子系统，然后选择**在新窗口中打开**。

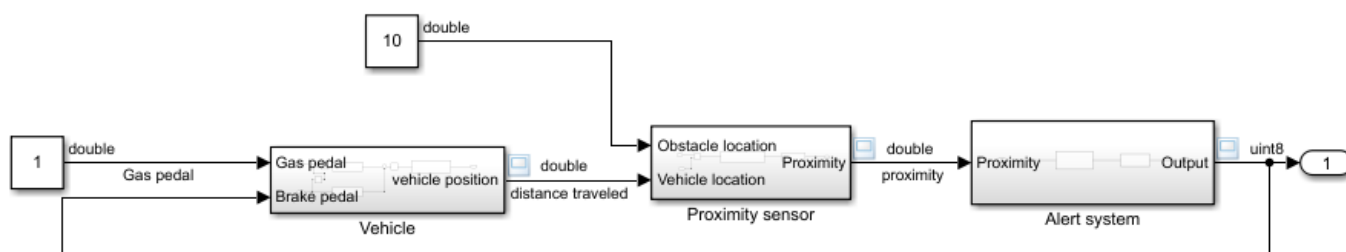
子系统上的每个输入端口或输出端口在子系统内都有对应的 Inport 或 Outport 模块。这些模块表示子系统与父系统之间的数据传输。当一个系统包含多个输入或输出端口时，Inport 或 Outport 模块上的数字表示该端口在子系统接口上的位置。

查看信号属性

Simulink 中的信号线表示从模块到模块的数据传输。信号具有与其在模型中的函数对应的属性：

- 维度 - 标量、向量或矩阵
- 数据类型 - 字符串、双精度、无符号整数等
- 采样时间 - 信号产生更新值的固定时间间隔（对于连续采样，则为 0）

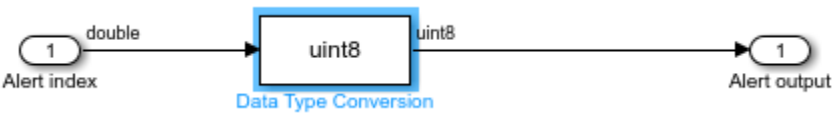
要显示一个模型中所有信号的数据类型，请在**调试**选项卡的**叠加信息**下，点击**基本数据类型**。



模型将在信号线旁显示数据类型。大多数信号均为双精度，只有 Alert 系统的输出例外。双击子系统进行调查。

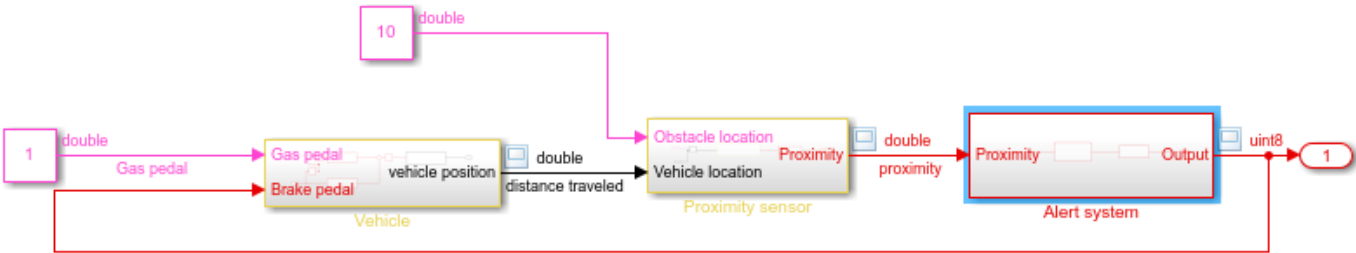


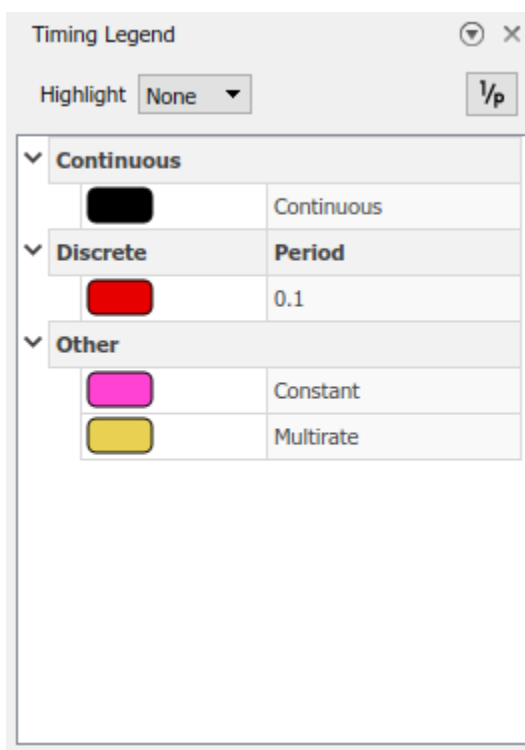
此系统中的数据类型标签显示，数据类型在 Alert device 子系统中发生改变。双击子系统进行调查。



Alert device 组件将 Alert index 信号从双精度转换为整数。您可以在源头设置数据类型，也可以使用 Signal Attributes 库中的 Data Type Conversion 模块。双精度（默认数据类型）提供最佳数值精度，是所有模块都支持的数据类型。双精度数据类型占用的内存和计算能力也最多。要对内存和计算能力有限的嵌入式系统建模，可使用其他数值数据类型。

要显示采样时间，请在调试选项卡的叠加信息下，点击“采样时间”部分的颜色。模型会更新以将模型中的各个采样时间显示为不同颜色，并显示图例。





- 具有连续动态的模块或信号显示为黑色。具有连续采样时间的信号根据 Simulink 的需要进行更新，以使计算尽可能接近真实情况。
- 保持不变的模块或信号显示为品红色。它们在仿真过程中保持不变。
- 以最低固定时间间隔更新的离散模块或信号显示为红色。具有离散采样时间的信号会以固定间隔更新。如果模型中包含具有不同固定采样时间的组件，则每个离散采样时间显示为不同的颜色。
- 同时包含离散和连续信号的多速率子系统显示为黄色。

跟踪信号

此模型有常量输入和离散输出。要确定采样方案从哪里发生改变，可以跟踪各个模块的输出信号。

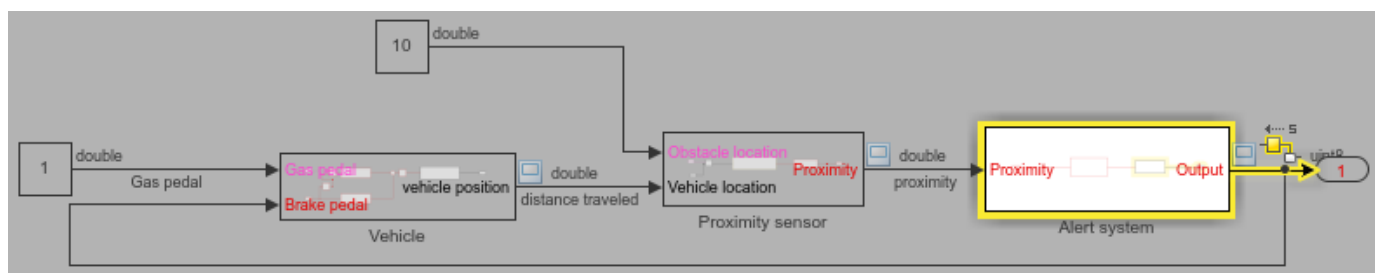
1

要打开模型浏览器，请点击**隐藏/显示模型浏览器**按钮 。

2

要突出显示输出信号，请选择信号，并在**信号**选项卡中，点击**跟踪到源**按钮 。

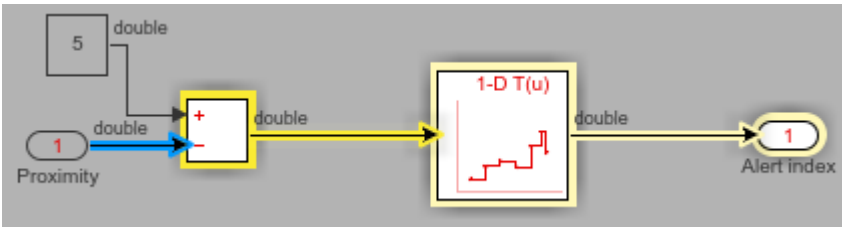
编辑器现在处于突出显示模式。点击编辑器以继续。



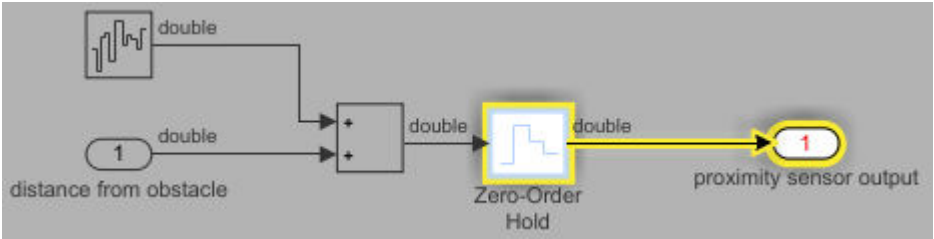
3 要继续跟踪信号以追溯其信源，请按向左箭头键。



4 继续跟踪信号以追溯其信源，直到到达 Alert logic 子系统。您会看到 Subtract 模块有两个输入。按向下箭头键，从 Inport 上选择信号路径。



5 要查找离散信源，请一直按左箭头，并留意反映采样时间的端口名称的颜色。



Sensor model 子系统中的 Zero-Order Hold 模块将信号从连续转换为离散。

另请参阅

相关示例

- “创建简单模型” （第 3-2 页）
- “使用 Simulink 进行基于模型的设计” （第 1-3 页）