

无人系统设计

课 程：软件工程专业-专业实践类课程

学 分：3

总课时：48

课程参考教材：

《认识飞行（第二版）》 / 《Understanding Flight, 2nd》

作者：David F. Anderson, Scott Eberhardt

译者：周尧明（2019年） / 韩莲（2011年）

北京联合出版公司2019.07 / 航空工业出版社2011.01

授课教师：王赓

课程助教：李旭辉、蒋李康、方俊杰、张源娣、范文婷、曹恺洋、杨道

课程主要内容

(1) 认识飞行

- ▣ 牛顿力学（作用力与反作用力）
- ▣ 刚体转动（转矩、陀螺、进动） / 大学物理基础

(2) 认识多种多样的无人飞行系统

- ▣ 飞行原理
- ▣ 动力技术（螺旋桨、喷气式）

(3) 控制技术

- ▣ 飞行操纵原理（机翼、襟翼、旋翼、尾桨、自动倾斜器）
- ▣ 作动器（电动机、舵机（PWM调制））
- ▣ 传感器（电子指南针、加速度计、陀螺仪、GPS、高度计、高速相机、全景相机、……）
- ▣ 电子控制器（PID算法、飞行控制原理与算法）

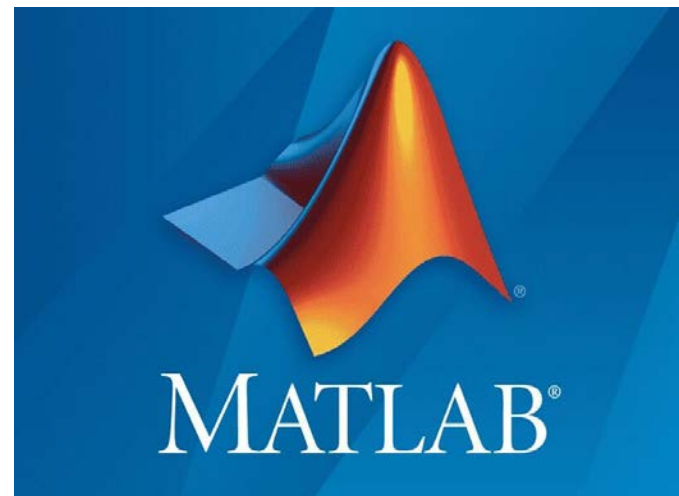
(4) 飞行性能（飞行性能指标体系、稳定性、可靠性、易操作性）

课程主要内容

- (5) 基于4旋翼、固定翼模型机的认知验证实验
(含早期自由组合发现学习过程)
- (6) 仿真技术
 - ▣ 飞行器建模 (动力学、运动学) / 大学物理基础、高等数学
 - ▣ 软件技术 (Unity3D、MATLAB/Simulink)
- (7) 仿真技术实践 (半实物)
 - ▣ 无人AI战机模拟格斗对抗系统
 - ▣ 软件技术 (Unity3D、MATLAB/Simulink、图像处理技术、人工智能AI技术、计算加速技术
- (8) 发挥想象力和所学的自由拓展设计 (理论设计/尽量据情实验验证)
- (9) 课程综合设计与答辩

MATLAB/Simulink软件应用技术

■ MATLAB/Simulink



课程实践平台

■ Simulink/MATLAB

■ 实践平台结构和接口构成

■ 示例及说明

■ 要求同学们尽早尽快熟悉Simulink/MATLAB应用

Simulink工作原理

■ 一个模型过程演示（请参阅系统相应说明文档）

The screenshot shows the Simulink R2020b Documentation page. The browser window has multiple tabs open, including 'TCP/IP Receive...', 'TCP/IP Send - ...', '搜索结果 - ...', 'echotcpip - ...', 'write - Mat...', 'tcpclient - M...', 'TCP/IP Interfa...', and 'Send and Receive Data o...'. The documentation page has a blue header with 'Documentation' and a search bar containing '搜索 R2020b 文档'. On the left, there is a sidebar with a '目录' (Table of Contents) section listing various documentation topics, with 'Send and Receive Data over TCP/IP Network' highlighted. The main content area shows the title 'Send and Receive Data over TCP/IP Network' with the version 'R2020b'. Below the title, there is a paragraph explaining the example: 'This example shows how to build a simple model using the Instrument Control Toolbox™ blocks in the block library in conjunction with other blocks in the Simulink® library. This example also illustrates how to send data to an echo server using TCP/IP and to read that data back into your model.' This is followed by another paragraph: 'In this example, you create an echo server on your machine that simulates sending a signal to the TCP/IP Send block and echo the result back to the Send block to send data. You then use the TCP/IP Receive block to read that same data back into your model.' Below this, there is a 'Note' box with an information icon and the text: 'Block names are not shown by default in the model. To display the hidden block names while working in the model, select **Display** and clear the **Hide Automatic Names** check box.' At the bottom, there is a list of steps: 'Step 1: Create an Echo Server', 'Step 2: Create a New Model', 'Step 3: Open the Block Library', 'Step 4: Drag the Instrument Control Toolbox Blocks into the Model', 'Step 5: Drag the Sine Wave and Scope Blocks to Complete the Model', 'Step 6: Connect the Blocks', 'Step 7: Specify the Block Parameter Values', and 'Step 8: Specify the Block Priorities'.

Documentation

搜索 R2020b 文档

目录

- « Documentation Home
- « Instrument Control Toolbox
- « Direct Interface
- Communication in Simulink

Send and Receive Data over TCP/IP Network

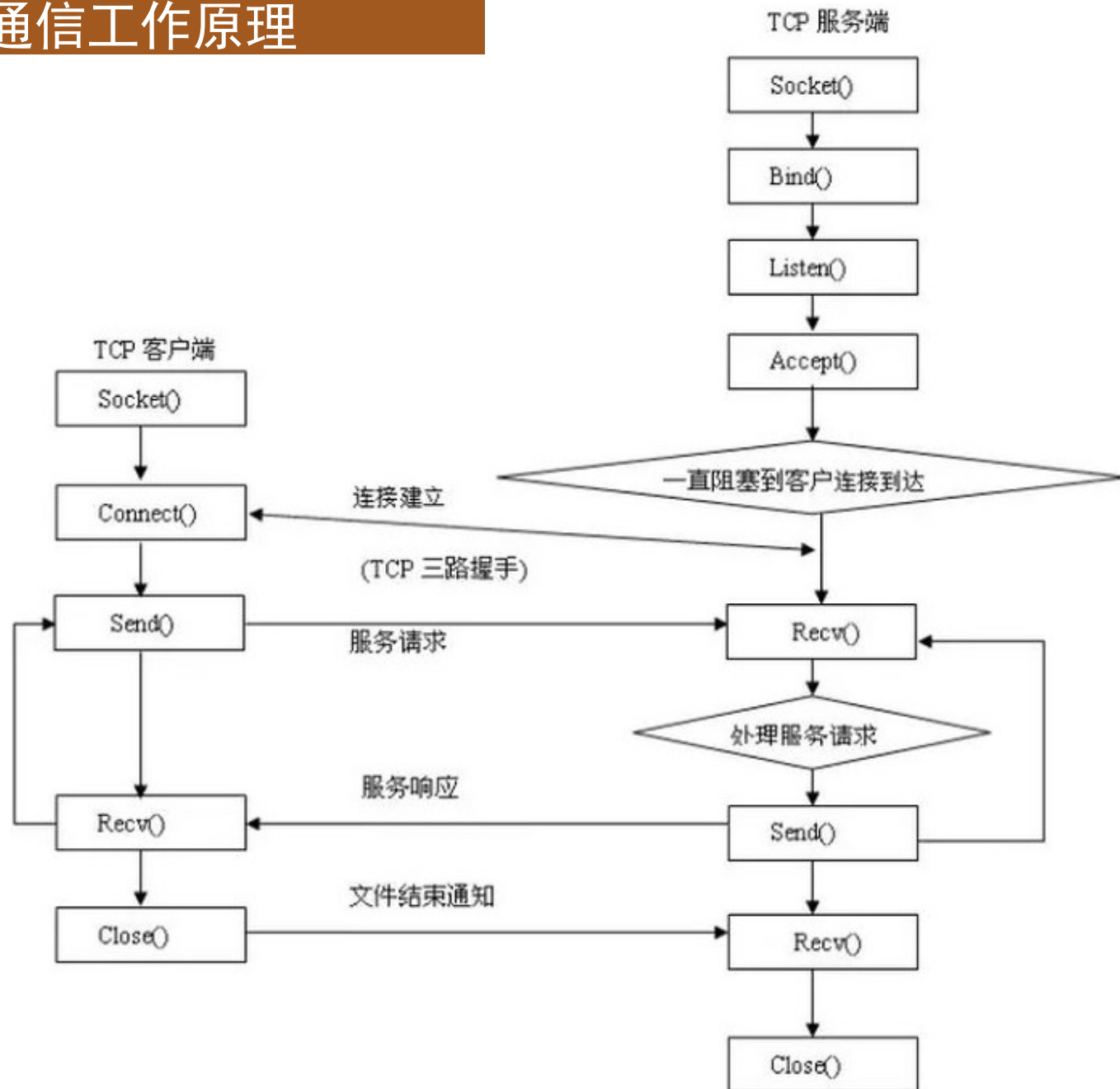
ON THIS PAGE

- Step 1: Create an Echo Server
- Step 2: Create a New Model
- Step 3: Open the Block Library
- Step 4: Drag the Instrument Control Toolbox Blocks into the Model
- Step 5: Drag the Sine Wave and Scope Blocks to Complete the Model
- Step 6: Connect the Blocks
- Step 7: Specify the Block Parameter Values

Note

Block names are not shown by default in the model. To display the hidden block names while working in the model, select **Display** and clear the **Hide Automatic Names** check box.

- Step 1: Create an Echo Server
- Step 2: Create a New Model
- Step 3: Open the Block Library
- Step 4: Drag the Instrument Control Toolbox Blocks into the Model
- Step 5: Drag the Sine Wave and Scope Blocks to Complete the Model
- Step 6: Connect the Blocks
- Step 7: Specify the Block Parameter Values
- Step 8: Specify the Block Priorities



Simulink中的S函数

■ S函数（S-Function）是Simulink的灵魂

目录

« 文档主页

« Simulink

« 模块编写和仿真集成

« 使用自定义模块扩展建模功能

« 使用 MATLAB 实现算法

« 创建和配置 MATLAB S-Function

类别

MATLAB S-Function 基础知识

创建 MATLAB S-Function

配置 MATLAB S-Function 的模块功能

文档 示例 函数 模块 视频 MATLAB Answers

试用版 产品更新

MATLAB S-Function 基础知识

R2020b

S-Function 实现的原则

S-Function 是以 MATLAB®、C、C++ 或 Fortran 语言编写的 Simulink® 模块的计算机语言描述。C、C++ 和 Fortran。使用 mex 实用工具将 S-Function 编译为 MEX 文件。请参阅编译 C MEX 函数。与其他 MEX 文件一样，S-Function 是动态链接的子例程，MATLAB 执行引擎可以自动加载和执行它们。如果您有 Simulink Coder™，则可以为内联的 2 级 MATLAB S-Function 生成代码。有关详细信息，请参阅 [Inlining S-Functions](#) (Simulink Coder)。

您可以通过以下方式实现 S-Function：

- 2 级 MATLAB S-Function - 支持访问更广泛的 S-Function API 集，并支持代码生成。
- C MEX S-Function - 使您能够将算法实现为 C MEX S-Function，或编写封装程序 S-Function 来调用现有 C、C++ 或 Fortran 代码。
- S-Function Builder - 提供图形用户界面，用于生成新 S-Function 或合并现有 C 或 C++ 代码，而无需与 S-Function API 交互。
- 代码继承工具 - 提供一组 MATLAB 命令，帮助您创建 S-Function 以合并现有 C 或 C++ 代码。

主题

What Is an S-Function?

Learn how S-functions work.

S-Function Concepts

Learn key concepts needed to write different types of S-functions.

S-Function Callback Methods

Implement S-function callback methods.

What Is an S-Function?

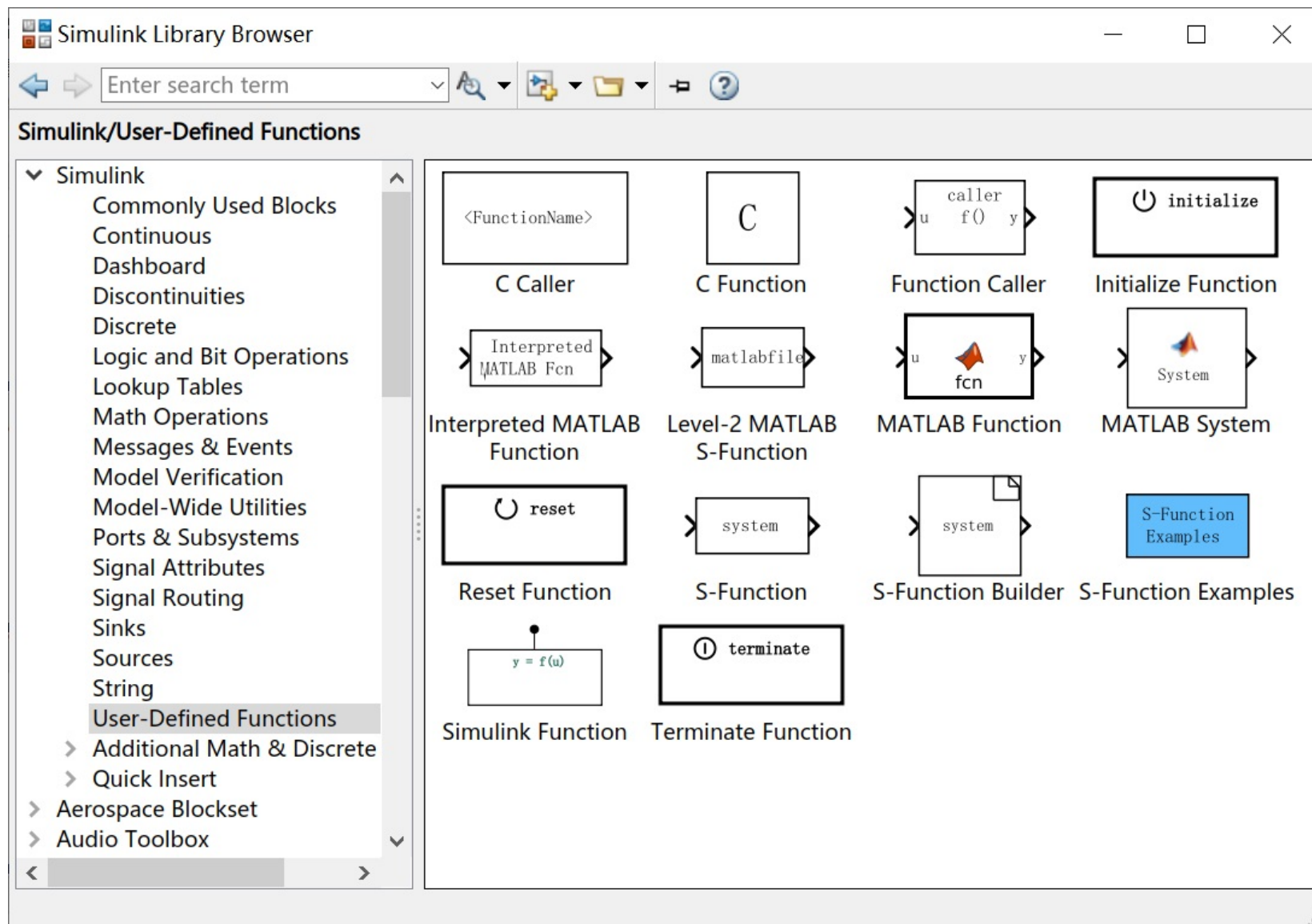
Learn how S-functions work.

S-Function Features and Limitations

Compare and contrast features supported by different types of S-function implementations.

参考：https://ww2.mathworks.cn/help/simulink/s-function-basics-matlab.html?s_tid=CRUX_lftnav

Simulink中的S函数



Simulink中的S函数

- S-Function 是以 MATLAB - M语言、C、C++ 或 Fortran 语言编写的 Simulink模块的计算机语言描述。
- 采用C、C++ 和 Fortran语言的模块行为描述，可以使用 mex 实用工具将 S-Function 编译为 MEX 文件（MATLAB/Simulink环境内的可执行文件）。
- S-Function 是动态链接的子例程，MATLAB 执行引擎可以自动加载和执行它们。
- 使用 Simulink Coder™，可以为内联的 Level 2 级 MATLAB S-Function 生成代码。

参考：https://ww2.mathworks.cn/help/simulink/s-function-basics-matlab.html?s_tid=CRUX_lftnav

S-Function 的实现方式

- Level 1 和 Level 2 级 MATLAB S-Function : Level 2 支持访问更广泛的 S-Function API 集，支持代码生成。
- C MEX S-Function : 通过C、 C++ 或 Fortran语言实现算法并封装为 C MEX S-Function，支持目标代码生成（**Simulink Coder** 和 **Embedded Coder** 及 **TLC**（**Target Language Compiler**）支持下）。
- S-Function Builder : 提供图形用户界面，用于生成新 S-Function 或合并现有 C 或 C++ 代码，而无需与 S-Function API 交互。
- 代码继承工具（Legacy Code Tool） : 提供一组 MATLAB 命令，帮助创建 S-Function 以合并现有 C 或 C++ 代码。

参考：https://ww2.mathworks.cn/help/simulink/s-function-basics-matlab.html?s_tid=CRUX_lftnav

Simulink中的S函数

- 其本质是系统中的一个环节对象的数学模型块。而该对象的行为模型是由用户自定义，采用M语言、C、C++、Fortran等程序语言进行描述的。
- 其模块行为一般难以或不便于采用显式的数学形式进行表述，而采用程序语言表述其行为。（当然也可以采用显式的数学模型来表达）
- 一个Simulink模块，是由一组输入、一组内部状态、一组内部参数和一组输出组成，其中输出是仿真时间、输入、参数和状态的函数。

参考：<https://ww2.mathworks.cn/help/simulink/sfg/what-is-an-s-function.html>

Simulink中的S函数

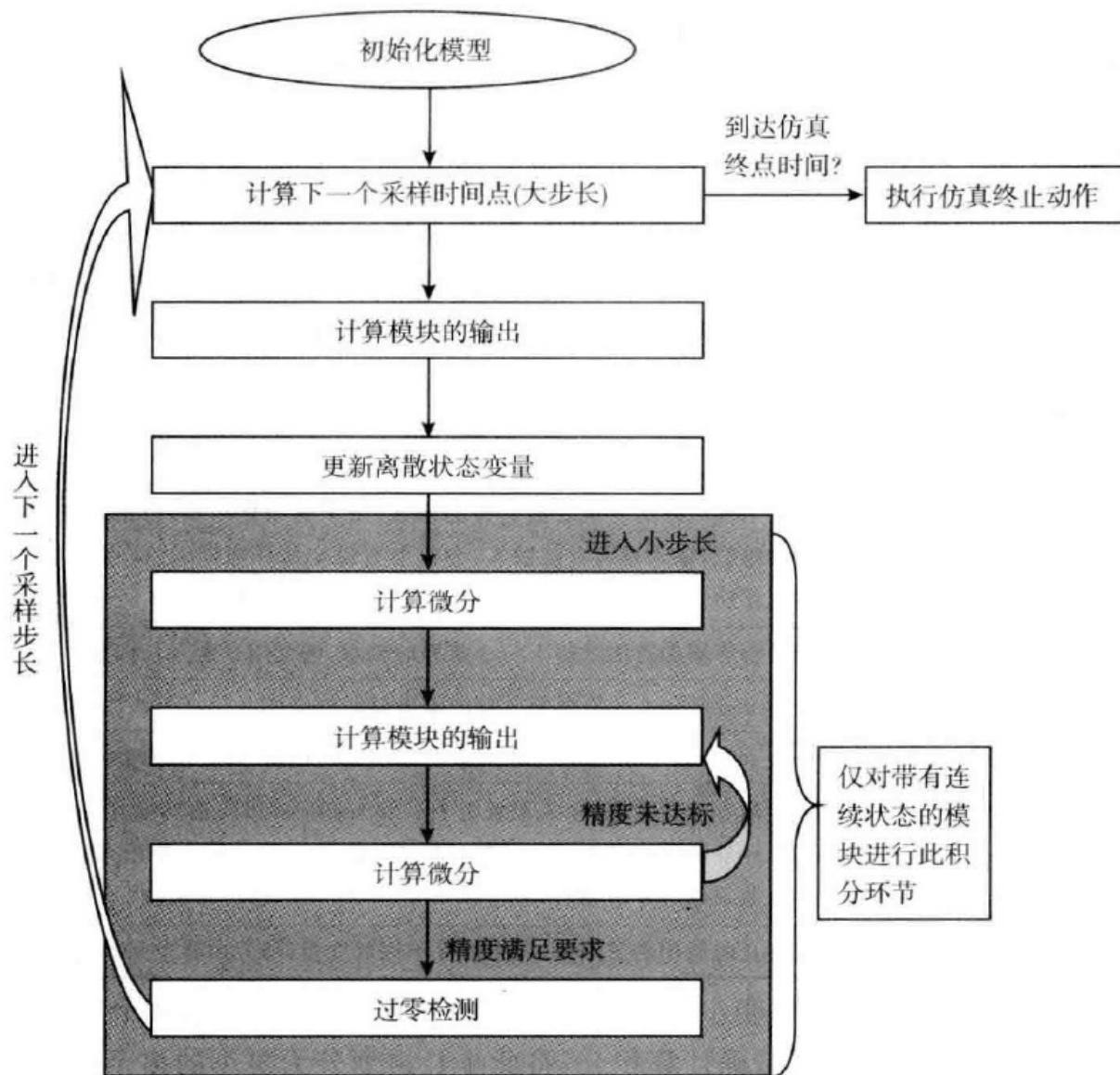
■ 一个S-Function 模块



- 3个要素：输入（端口个数可以为0）、输出（端口个数可以为0）、模型内部状态量（个数可以为0）。（参数*）

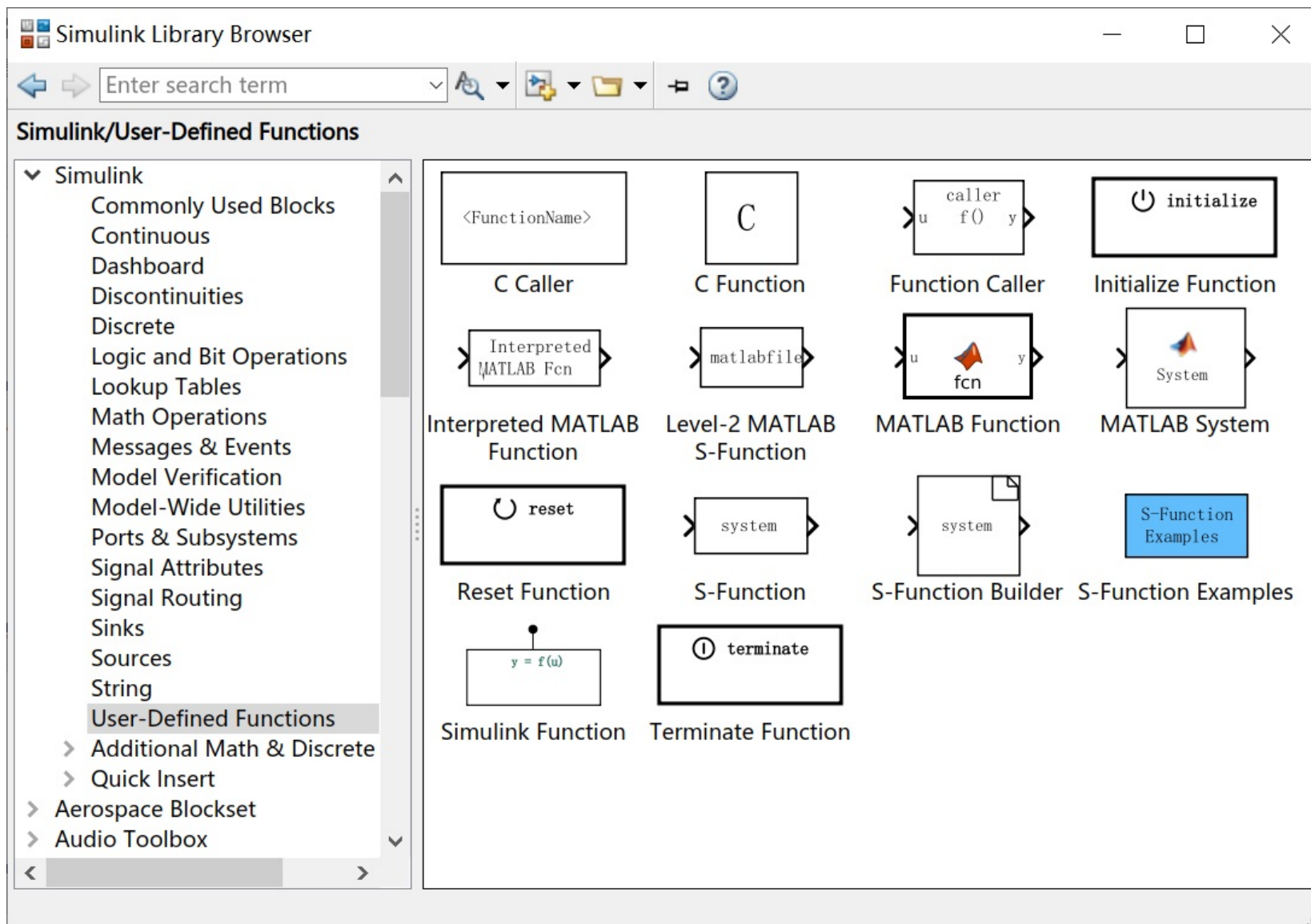
参考：<https://ww2.mathworks.cn/help/simulink/sfg/what-is-an-s-function.html>

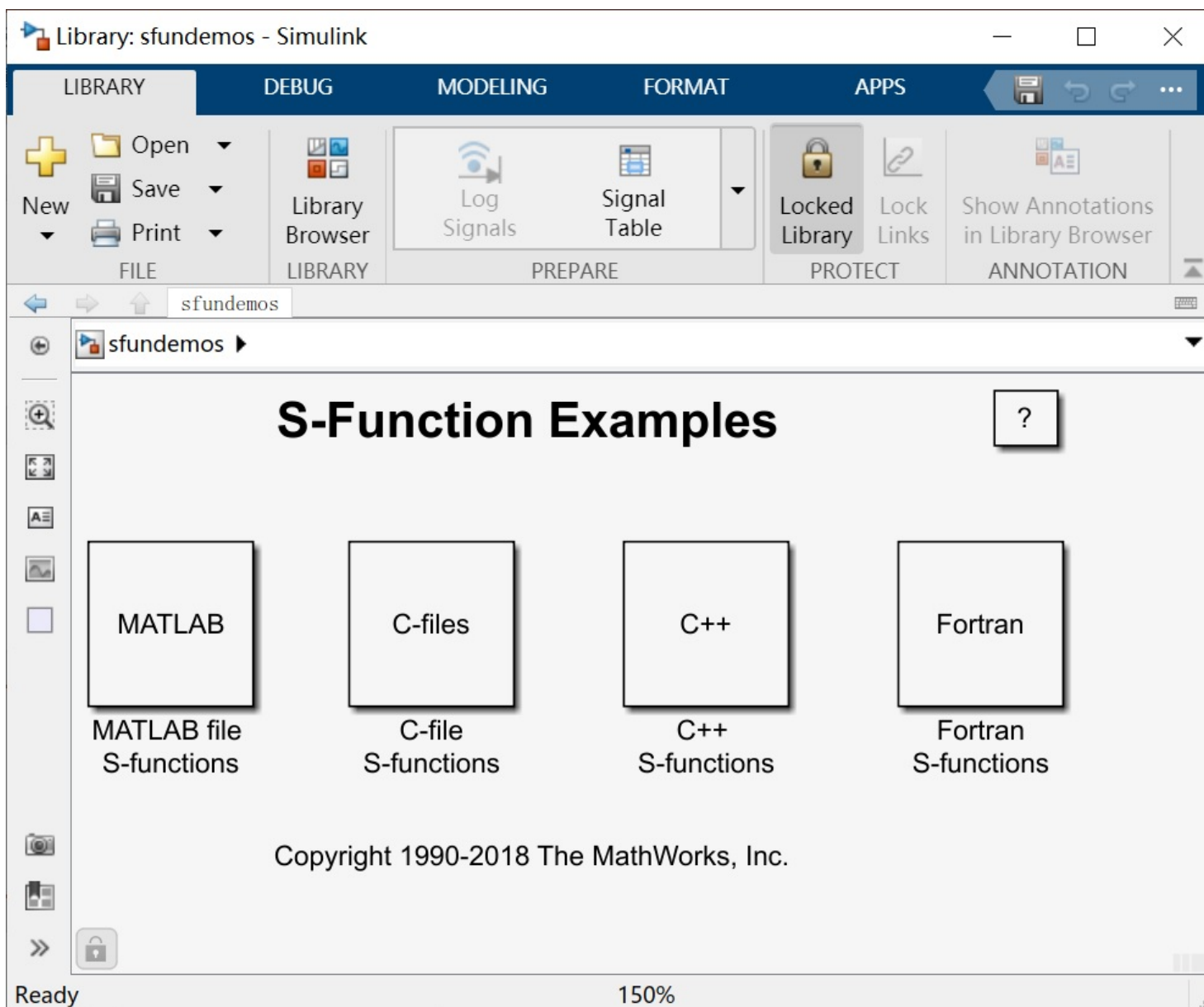
S函数在每个时间步中的仿真运行过程

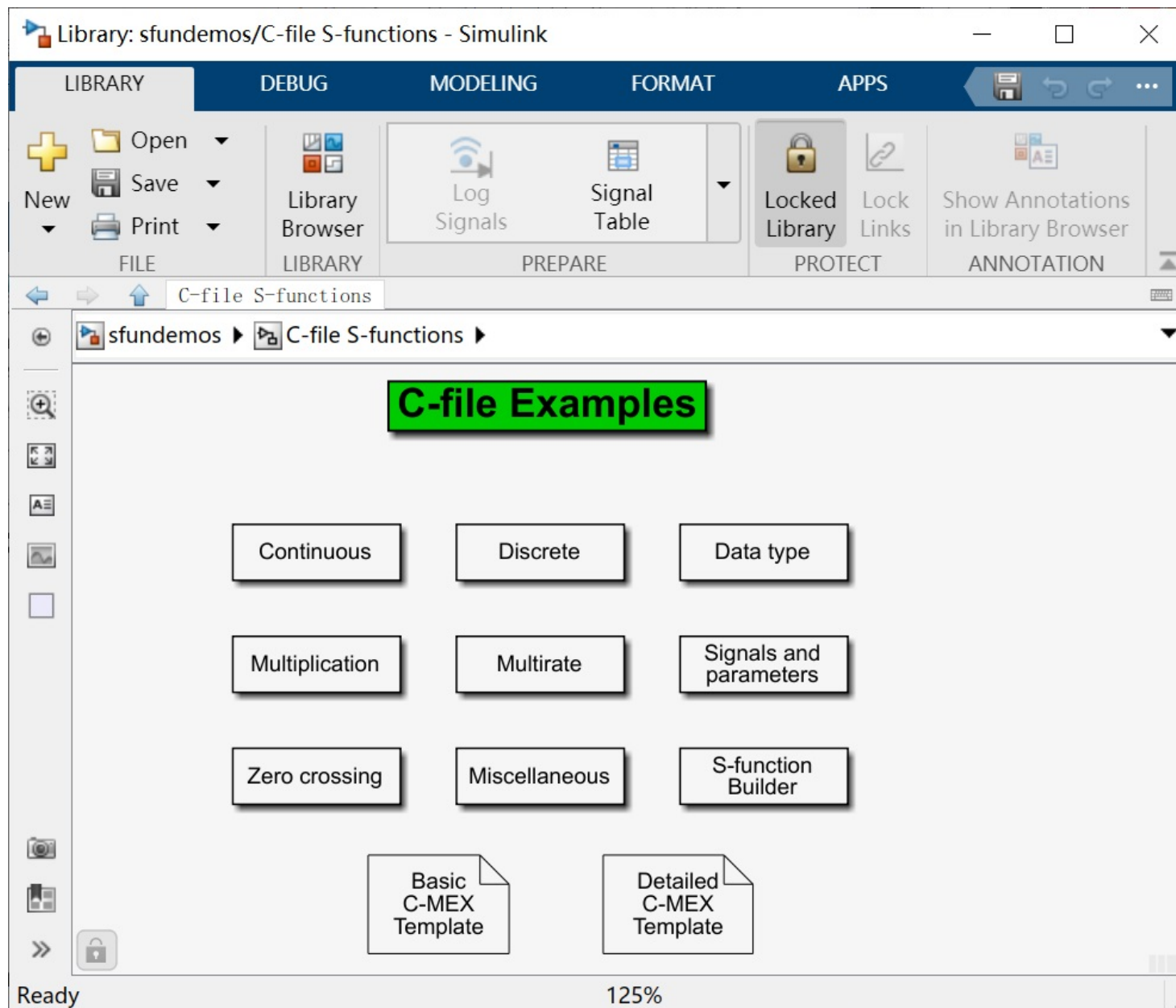


参考：参考书1
图10.4-1 S函数运行过程

Simulink中的S函数

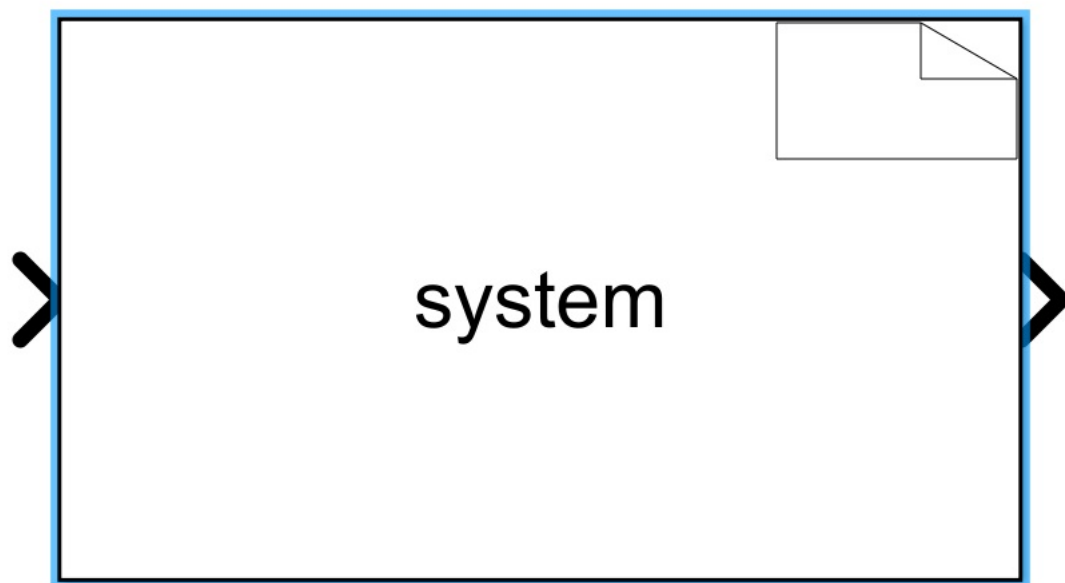






Simulink中的S函数

- 使用S-Function Builder 函数模板配置生成工具



S-Function Builder

S-Function Builder

S-Function Builder: untitled/S-Function Builder

S-FUNCTION BUILDER

S-Function Name:

Language:

Insert Ports

Comment

Indent

Build

S-FUNCTION

EDIT

BUILD

Editor

SETTINGS

```
18 void my_sfxxxx_01_Outputs_wrapper(const real_T *u0,
19                                   real_T *y0,
20                                   const real_T *xD)
21 {
22     /* Output_BEGIN */
23     /* This sample sets the output equal to the input
24        y0[0] = u0[0];
25     For complex signals use: y0[0].re = u0[0].re;
26        y0[0].im = u0[0].im;
27        y1[0].re = u1[0].re;
28        y1[0].im = u1[0].im;
29     */
30     /* Output_END */
31 }
32
33 void my_sfxxxx_01_Update_wrapper(const real_T *u0,
34                                  real_T *y0,
35                                  real_T *xD)
36 {
```

Number of discrete states

Discrete states IC

Number of continuous states

Continuous states IC

Array layout

Sample mode

Sample time value

Number of PWorks

Enable access to SimStruct ☐

Direct feedthrough ☒

PORTS AND PARAMETERS

LIBRARIES

Name	Scope	Data Type	Dimensions	Complexity
u0	input	double	[1,1]	real
y0	output	double	[1,1]	real

Simulink中的S函数 (利用S-Function Builder)

- 学习链接: “**Build S-Functions Automatically Using S-Function Builder**”
- <https://ww2.mathworks.cn/help/simulink/sfg/s-function-builder-dialog-box.html>

及其引申的相关链接:

See the examples [Moving Average with Start and Terminate](#)
and [Permutation using Cpp Classes](#) in [S-Function Demos](#). 等

参考: <https://ww2.mathworks.cn/help/simulink/sfg/s-function-builder-dialog-box.html>

课程实践平台

■ Simulink/MATLAB

■ 示例及说明

■ 实践平台接口构成

- 要求同学们尽早尽快熟悉Simulink/MATLAB应用
(包括如先练习写个UI界面, 发布为.exe等)

课程主要内容

(1) 认识飞行

- ▣ 牛顿力学（作用力与反作用力）
- ▣ 刚体转动（转矩、陀螺、进动） / 大学物理基础

(2) 认识多种多样的无人飞行系统

- ▣ 飞行原理
- ▣ 动力技术（螺旋桨、喷气式）

(3) 控制技术

- ▣ 飞行操纵原理（机翼、襟翼、旋翼、尾桨、自动倾斜器）
- ▣ 作动器（电动机、舵机（PWM调制））
- ▣ 传感器（电子指南针、加速度计、陀螺仪、GPS、高度计、高速相机、全景相机、……）
- ▣ 电子控制器（PID算法、飞行控制原理与算法）

(4) 飞行性能（飞行性能指标体系、稳定性、可靠性、易操作性）

课程主要内容

- (5) 基于4旋翼、固定翼模型机的认知验证实验
(含早期自由组合发现学习过程)
- (6) 仿真技术
 - ▣ 飞行器建模 (动力学、运动学) / 大学物理基础、高等数学
 - ▣ 软件技术 (Unity3D、MATLAB/Simulink)
- (7) 仿真技术实践 (半实物)
 - ▣ 无人AI战机模拟格斗对抗系统
 - ▣ 软件技术 (Unity3D、MATLAB/Simulink、图像处理技术、人工智能AI技术、计算加速技术
- (8) 发挥想象力和所学的自由拓展设计 (理论设计/尽量据情实验验证)
- (9) 课程综合设计与答辩

Question & Answer

任何疑问和建议，请不要犹豫！

王 赓: wgeng@sjtu.edu.cn