Simulink自主生成C/C++代码实验原理

1.文件地址

例程目录：[安装目录]\RflySimAPIs\4.RflySimModel\0.ApiExps\2.UserDefinedC++

文件夹/文件名称	说明
2.GenC++\Readme.pdf	自主生成C/C++代码实验步骤
1.CallC\Readme.pdf	Simulink调用C代码实验步骤

2.总体说明

2.1 基于模型设计的需求

基于模型的设计是一种快速、经济高效的动态系统（包括控制系统、信号处理和通信系统）开发过程。在基于模型的设计中，系统模型是整个开发过程（从需求开发到设计、实现和测试）的核心。模型是在开发过程中不断优化的可执行规范。完成模型开发之后，可通过仿真来显示模型是否能够正常工作。

基于模型的设计能够实现仿真平台和硬件平台的通用控制设计，把复杂的控制算法通过自动代码生成工具迅速移植到不同的硬件平台，提高了系统的开发效率。基于模型的开发模式，强调的是在需求定义、系统设计、产品实现、系统集成的阶段均以模型为主进行系统建模仿真和验证。完成验证工作后，可以将模型利用工具自动化地生成嵌入式系统代码。持续的设计验证和嵌入式代码的自动生成是基于模型设计的两大核心特征。

2.2 将C代码嵌入Simulink仿真

Simulink可以利用已有的C代码实现一些复杂或特定的功能，从而提高仿真的效率和准确性。例如，可以通过调用外部C代码实现一些数学函数、信号处理算法、硬件接口等。这样，不仅可以避免重复编写相同的逻辑，还可以方便地将生成的代码部署到嵌入式系统中。

simulink提供了多种方式来调用外部C代码，如使用S-Function Builder模块、编写自定义的S函数、使用Legacy Code Tool或者使用Embedded Coder。

2.3 生成针对嵌入式系统优化的 C 和 C++ 代码

MATLAB/Simulink的Embedded Coder模块可生成可读、紧凑且快速的 C 和 C++ 代码，以便用于大规模生产中使用的嵌入式处理器。它扩展了 MATLAB Coder 和 Simulink Coder 的功能，支持通过高级优化对生成的函数、文件和数据进行精确控制。这些优化可提高代码效率，并有助于与已有代码、数据类型和标定参数集成。可以集成第三方开发工具，以便为嵌入式系统或快速原型板上的全套部署构建可执行文件。

3.实验原理

3.1 Simulink调用C代码实验原理

3.1.1 MATLAB Function调用c代码

MATLAB Coder 快速入门 - MathWorks中国

MATLAB Function模块可以编写基于MATLAB语言的函数，该函数可以在Simulink仿真中执行。如果需要将c代码嵌入到MATLAB Function中，可以使用coder.ceval函数来调用c函数，需要指定c函数的名称、输入参数。coder.ceval函数会在生成代码时，将MATLAB语言转换为对应的c语言，并保持与MATLAB Function中的逻辑一致。这样就可以实现在Simulink仿真中，同时运行MATLAB语言和c语言的功能。

配置源代码文件

确保Typedef.h和相应的C源文件位于当前工作目录或指定的路径中。

Typedef.h头文件应该包含GetMaxValue和SquareOperation函数的声明

Typedef.h

```
\#ifndef \_TYPEDEF_H

\#define \_TYPEDEF_H

\#define EXLIB_API \__declspec(dllexport)

typedef unsigned char uint8_t;

typedef short unsigned int uint16_t;

EXLIB_API extern uint16_t MaxValue;

EXLIB_API extern uint16_t Result;

EXLIB_API extern uint16_t GetMaxValue(uint16_t Input1, uint16_t Input2);

EXLIB_API extern uint16_t SquareOperation(uint16_t Input);

\#endif
```

相应的C源文件应该包含这些函数的定义

Max.c

```
\#include "Typedef.h"

uint16_t GetMaxValue(uint16_t Input1, uint16_t Input2)

{

if(Input1 \> Input2)

{

return Input1;

}

else

{

return Input2;

}

}
```

Square.c

```
\#include "Typedef.h"

uint16_t MaxValue = 4500;

uint16_t Result;

uint16_t SquareOperation(uint16_t Input)

{

uint16_t Result = Input \* Input;

if(Result \> MaxValue)

{

Result = MaxValue;

}

return Result;

}
```

编写MATLAB Function代码调用C函数

MATLAB Function代码示例如下，旨在通过调用外部C函数来实现计算操作：

1.初始化输入输出

function y = Mfcn(u1,u2)

y = 0;

定义了一个MATLAB函数Mfcn，该函数有两个输入参数u1和u2，并返回一个输出y。

初始化输出变量y为0。尽管在后续代码中y的值会被更新，但初始化有助于确保变量有一个初始值，避免潜在的未定义行为。

2. 设置编译器的包含路径和包含必要的头文件

为了让MATLAB生成的C代码成功编译并链接外部C函数，需要确保编译器能找到这些函数的声明和实现。具体步骤如下：

2.1 coder.updateBuildInfo初始化和设置编译器路径

coder.updateBuildInfo用于动态修改编译设置。在本例中，'addIncludePaths'选项用于添加头文件的搜索路径。其用法如下：

coder.updateBuildInfo('addIncludePaths', 'path_to_include_directory');

这样做的目的是确保编译器在编译时能够找到所需的头文件。在本例中使用的是：

coder.updateBuildInfo('addIncludePaths','[pwd]');

[pwd]返回当前工作目录，因此这行代码将当前工作目录添加到编译器的头文件搜索路径中。

2.2 coder.cinclude包含必要的头文件

coder.cinclude指示MATLAB Coder在生成的C代码中包含一个指定的头文件。其用法如下：

coder.cinclude('header_file_name');

在本例中使用的是：

coder.cinclude('Typedef.h');

这行代码确保生成的C代码包含头文件Typedef.h，使得C编译器在编译时可以找到GetMaxValue和SquareOperation函数的声明。

3. coder.ceval调用外部C函数

coder.ceval是MATLAB
Coder提供的一个函数，用于在生成的C代码中调用外部C函数。其基本语法为：

y = coder.ceval('C_function_name', arg1, arg2, ...);

其中，C_function_name是要调用的C函数名，arg1, arg2, ...是传递给该C函数的参数。

在本例中使用如下：

y = coder.ceval('GetMaxValue', u1, u2);

y = coder.ceval('SquareOperation', y);

这些代码通过coder.ceval调用外部C函数GetMaxValue和SquareOperation。GetMaxValue返回输入参数u1和u2中的最大值，SquareOperation返回该最大值的平方。

配置代码生成设置

在MATLAB中，配置Simulation
Target设置是为了确保在仿真过程中正确调用和链接外部C代码。Simulation
Target设置主要用于Simulink模型的仿真环境，以确保外部代码在仿真期间能够正确地编

译、链接并执行。下面是如何配置Simulation
Target设置的详细步骤：

1. 打开Simulink模型

首先，打开要配置的Simulink模型。

2. 进入模型配置参数

点击菜单栏中的 **Modeling** > **Model Settings**，或者使用快捷键 Ctrl+E
打开模型配置参数对话框。

3. 设置Simulation Target

在模型配置参数对话框中，选择 **Code Generation** > **Simulation
Target**。在这里，你可以配置包含路径、库路径和源文件等。

4. 添加自定义代码

如果你需要在生成的代码中包含自定义代码（如头文件的引用），可以在 **Custom Code**
部分进行配置。

4.1. Custom Include Files

在 **Custom Code** > **Include Header** 字段中，添加你需要包含的头文件。

#**include** "Typedef.h"

4.2. Custom Source Files

在 **Custom Code** > **Source files** 字段中，添加自定义的C源文件。

Max.c

Square.c

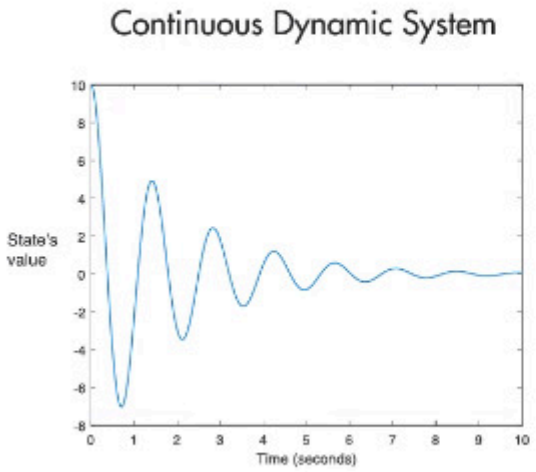
5. 验证配置

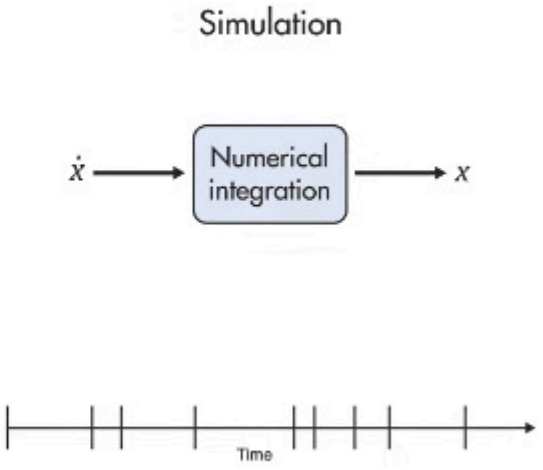
完成以上配置后，点击 **Apply** 应用设置，确保所有配置正确无误。

3.2 自主生成C/C++代码实验原理

3.2.1 Simulink求解器

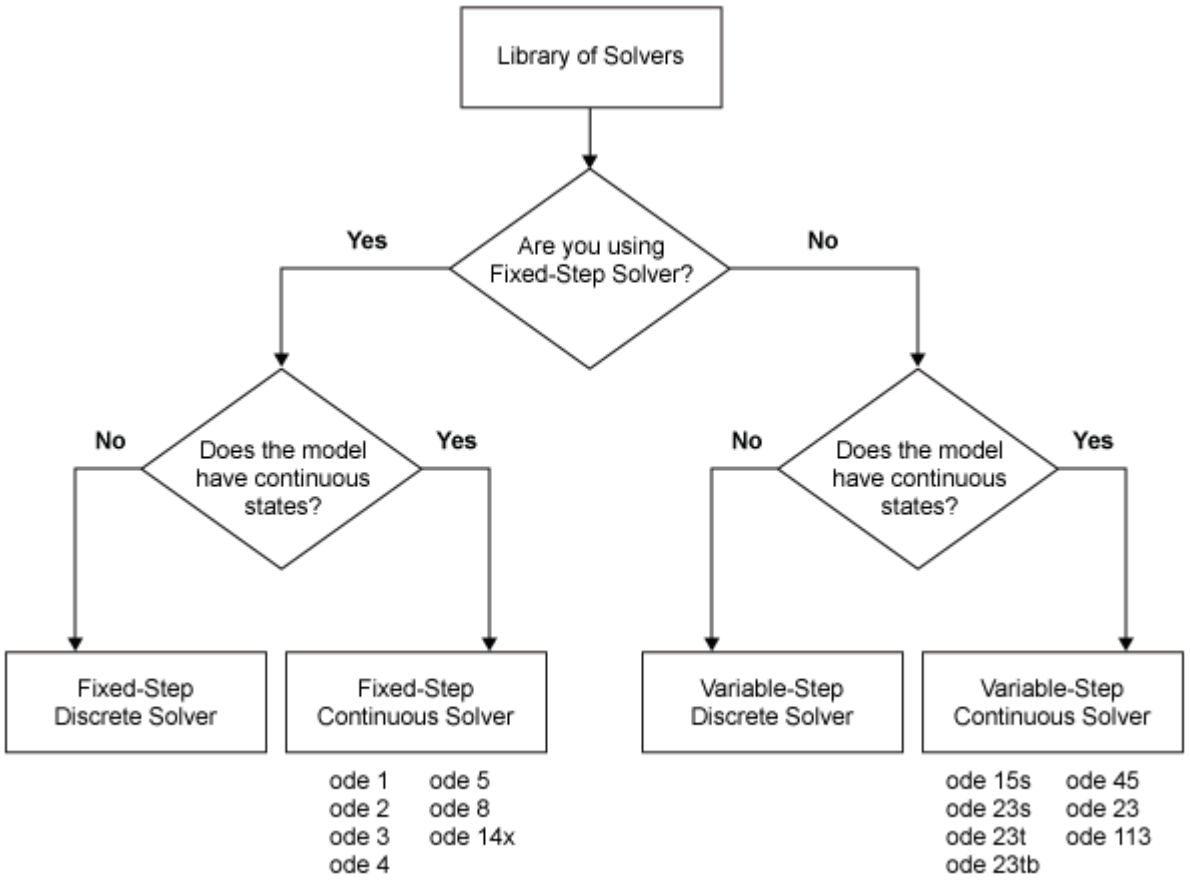
- 详细配置步骤见<Readme.pdf>





连续动态系统的状态值随时间连续需变化，而simulink通过数值积分得到每个时间步长的状态值来模拟这种变化，而这些都是通过**求解器**来实现的。

	Continuous solver	Discrete solver
Variable - step	For simulating and developing most applications	For simulating and developing controllers, filters...
Fixed - step	For hardware-in-the-loop simulations, code generation...	For deploying controllers, filters...



Simulink
提供的求解器可大致按两个属性分类①步长是否可变，②是否连续。在选择求解器时，需要先选择的时分类是定步长求解器，还是可变步长求解器。

步长分类

- **定步长求解器**：顾名思义，固定步长求解器从仿真开始到仿真结束使用相同的步长大小，来解算模型。用户可以指定步长大小，也可以让求解器选择步长大小。一般情况下，减小步长，可以提高结果的准确性，但是会增加工作量，系统仿真所需的时间。
- **可变步长求解器**：步长在不同采样点计算时，可能是变化的，步长具体增大还是减小，取决于模型的变化速度，当模型状态变化缓慢时，求解器将增加步长大小以避免执行不必要的步长。

可变步长会增加每个步长的计算时间，但是可以减少步长总数。因此，对于具有过零、快速变化的状态以及其他需要额外计算的事件的模型，保证指定的准确度的同时，较短时间结束仿真。

选择完步长类型后，还要选择离散或者连续性。

两种求解器的服务对象分别是模型中的连续或者离散模块。

- **连续求解器**:用数值积分方法，根据模型在上一个时间的状态和状态导数,来计算模型在当前时间的连续状态。连续求解器依赖单个连续模块来计算模型在每个时间步的离散状

态值。

- **离散求解器**：解算纯离散模块组成的模型。只计算模型的下一个仿真时间。执行这种离散计算时，依赖于模型中的每个模块来更新其各个离散状态。

在Simulink生成嵌入式代码时，一般使用的求解器都是定步长求解器，因为在ECU硬件实里面，算法程序都是固定周期运行的。

Simulink求解器的工作分为3步：计算出当前状态的输出，更新当前的状态决定当前到下1步的仿真步长。其作用是驱动静态的模型进行动态系统仿真。它相当于模型内部的发动机，驱动着模型随着仿真而更新模型状态。

求解顺序

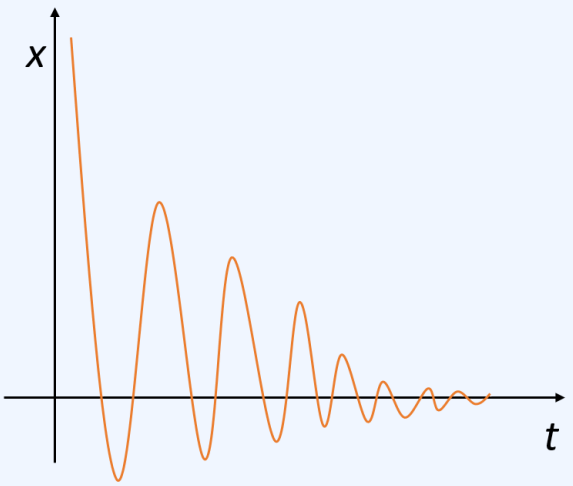
当 Simulink 运行仿真时，它将首先确定模块输出更新的顺序。这称为排序执行顺序。

代数系统不包含连续状态。在该系统中，所有的模块都是直接馈通。这意味着模块输出取决于同一时间步长的模块输入。

一些 Simulink
模块可以在不知道相同时间步长的输入值的情况下计算其输出。这是积分器模块的情况。它的输出是它的连续状态，并且它是通过对来自较早时间步长的输入进行积分来计算的（在模拟的第一时间步中，使用初始条件）。在每个时间步长结束时，在模块更新步骤中更新积分器的状态。

求解精度

积分器块通过在计算微分方程的解时设置可变步长求解器的步长来确保在给定误差容限内的模拟精度。例如，由于初始条件，质量-弹簧-阻尼器系统中质量的自然（非受迫）响应可计算为：
$M\ddot{x}+D\cdot x+Kx=0$
根据弹簧系数 K 或阻尼系数 d 的值，非受迫响应可以缓慢变化（过阻尼响应）或快速变化（振荡响应）。为了让 Simulink 准确计算这些响应，可变步长求解器将改变其步长以确保精度，同时保持良好的整体仿真速度。

	
缓慢变化的响应（可变步长解算器使用较大的步长，因为信号在每个时间步长不会发生显著变化）	快速变化的响应（可变步长求解器需要小步长来精确求解此模拟）

3.2.2 嵌入式代码生成工具链

- 平台所需详细设置步骤见<Readme.pdf>
- 完整工作流参考[1]
- 更详细地了解Embedded coder： 2.GenC++\ref

相关文献

1. 使用 Embedded Coder 生成代码 - MATLAB & Simulink - MathWorks中国
   Simulink调用外部C代码的几种方法_simulink如何利用interpreted matlabfunction
2. 模块调用c语言代码-CSDN博客

附加资源

官方文档：RflySim官方文档： https://rflysim.com/doc/zh/

社区交流：加入RflySim技术交流群：951534390


RFLYSIM