

基于 VR-Link/RTW 实现导弹制导 仿真联邦成员设计*

陈 涛, 张国耀, 李莎莎

(哈尔滨工程大学信息与通信工程学院, 哈尔滨 150001)

摘 要: 为了实现由 Simulink 建立的导弹控制模型与雷达模型进行实时的交互, 文中利用 RTW 工具将 Simulink 模型转化为 C++ 代码, 介绍了将 Simulink 模型改造成 HLA 成员的实现方法, 对 RTW 代码的生成、模型代码的执行过程以及修改 RTW 代码使其能在 HLA 规范下使用作了阐述。最后给出了基于 VR-Link 的导弹控制联邦成员的具体实现方法, 具有很好的工程实用价值。

关键词: Simulink 模型; RTW; VR-Link

中图分类号: TJ765.3 **文献标志码:** A

Realize Missileguidance Simulation Federation Member Based on VR-Link/RTW

CHEN Tao, ZHANG Guoyao, LI Shasha

(College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China)

Abstract: In order to realize real time interaction between missile control model built by Simulink and radar model, the Simulink model was converted to C++ code using RTW tools, the process of changing Simulink model into HLA federate was introduced, and the RTW code generation, the execution process of model code and the method of amending the RTW code to regulate the use of HLA were described. In the end, the specific implementation of missile control federate based on VR-Link was given. This method is featured with high engineering value.

Keywords: Simulink model; RTW; VR-Link

0 引言

目前, 存在许多已开发好的 Simulink 模型, 正在设计开发的模型有相当大的一部分模型是在 Simulink 平台上开发的。RTW (real-time workshop) 是 Matlab 提供的与 Simulink 一起使用的一个工具, 它可以直接从 Simulink 模型中产生优化的、可移植的和个性化的 C/C++ 代码, 可以通过改造这些代码使其适应新的环境^[1-3]。将 Simulink 仿真模型嵌入到 HLA 框架之下, 无疑可以解决 Simulink 模型与其他应用程序的交互和互操作^[4]。

导弹制导仿真系统中, 导弹控制模型是由 Simulink 建立的, 而雷达模型是由 VC++ 编写的, 在仿真过程中, 导弹控制模型需要接收雷达

的测角信息, 雷达也要实时地接收导弹飞行的实时数据。为了实现导弹控制模型与雷达模型的数据交互, 文中提出了将导弹控制模型利用 RTW 工具转化为 C/C++ 代码, 在 VC6.0 中调用这些代码, 将其改造为一个基于 VR-Link 的 HLA 联邦成员的方法。

1 RTW 代码的生成

RTW 用户界面由 Simulink 提供的仿真参数框“Simulink Parameters”选项打开^[2]。

下面以 example.mdl 为例说明 RTW 代码的生成方法:

1) 定义输入和输出信号名称以备嵌入后使用, 利用鼠标右键单击模型中需要设置的信号, 选取 Signal Properties 菜单弹出的对话框, 选择

* 收稿日期: 2008-09-04

基金项目: 国防基础科研基金资助

作者简介: 陈涛(1974—), 男, 辽宁铁岭人, 副教授, 研究方向: 系统仿真, 图形图像处理。

RTW storage class 为 ExportedGlobal, 把信号设置为全局变量。

2) 设置 Simulink time 参数, Start time = 0.0, Stop time = inf。

3) 设置 Solver options 参数, Type 为 Fixed-step, 设置 Fix Step Size 参数为一个合适的值 (≤ 0.5), 文中选取步长为 0.02s, 求解器选择 ode4。注意, 对于 GRT (generic real-time target) 和大多数其他目标, 必须指定为固定步长类型的求解器。

4) 在 Real-Time Workshop 中的 Category 中选择 Target configuration, 配置 System target file, 选择系统目标文件 (System target file) grt.tlc, language 选择 C++, 为了以后可以在 VC 6.0 中编译程序, 选择“Visual C/C++ Project Makefile only for the grt target”。

5) 选中 Generate Code Only 并按下 Build。

2 模型代码的执行过程

RTW 根据模型开发者的指定生成模型的算法代码, 同时还提供模型代码的执行驱动代码 (包括函数 MdlInitialize(), MdlOutputs(), MdlUpdate() 等), 此外还有操作系统接口例程、依赖 I/O 的例程、解算器、数据登记例程等。Simulink 模型都是在一个基本的运行框架下调用这些运行时间接口来执行仿真过程的, 该基本运行时间框架如图 1 所示^[3]。

模型的执行过程首先进入初始化阶段, 这包括注册模型、分配模型运行时所使用的数据结构的工作区并对其初始化, 随后运行 MdlStart 函数, 已完成对模型状态变量的初始化以及对条件执行子系统模块的初始化等, 然后进行模型的单步执行, 包括在时刻 t 处计算模型的输出 (MdlOutputs)、记录工作空间中的 I/O 数据及对模型的离散状态的更新 (MdlUpdate)。

MdlOutputs 和 MdlUpdate 的区别是: MdlOutputs 负责更新模型中的连续部分的状态并产生模型的输出, 它既可以在每次积分的整个积分步长开始时调用, 也可以在每次积分的每个积分步之间调用, 而 MdlUpdate 负责更新模型中的离散部分的状态, 只用在整个积分步长开始

调用。对于模型中存在的连续时间状态, 需要调用模型的导数计算过程 (MdlDerivatives), 对连续状态的导数进行积分以生成时刻 $t_{\text{new}} = t + h$ 时的状态 (这里 h 为步长)。最后时间向前移到 t_{new} 。在仿真过程中反复调用这些函数, 仿真时间得以推进, 最后仿真结束时调用 MdlTerminate。整个运行框架由 main() 函数实现。

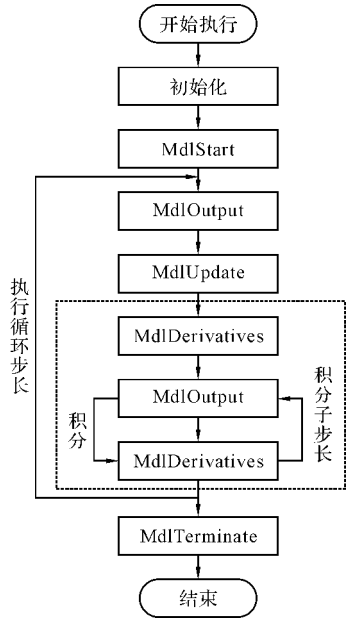


图1 Simulink 仿真运行时间框架

3 模型代码的修改

为了让生成的 C++ 代码能在 HLA 规范下使用, 需要对 RTW 生成的模型代码进行修改, 主要包括两个方面^[4]。

1) 对原来的模型运行框架进行修改。将它的运行框架根据需要分为三个部分: 模型初始化模块、模型单步运算模块和模型终止模块。模型初始化模块在加入联邦之前调用模型的初始化函数对模型进行初始化; 需要自行定义一个模型单步运算的函数, 单步运算的函数如下 (以 example.mdl 为例):

```
void Sim_OneStep(example _rtModel * MODEL_example)
{
    example_output(0);
    rt_ertODEUpdateContinuousStates(&MODEL_example->solverInfo);
}
```

该函数每执行一次模型仿真时间向前推进一个仿真步长。在程序结束时调用模型终止部分。

2) 为了让模型代码能通过 RTI 与进行数据交互, 将模型的输入输出连接到某个对象类的属性中, 或是某个交互类的参数中, 这样通过公布/订阅对象类或是接收交互来实现与其他联邦成

员的数据交换。

4 基于 VR-Link 的导弹控制仿真联邦成员的实现方法

联邦成员的功能是利用 VR-Link 工具库中的 HLA 功能实现的,VR-Link 提供了一个仿真应用于 RTI 之间的接口,使用户不必直接访问 RTI 的 API,极大地减少了用户的开发工作量。具体的开发过程如下:

1) 加入联邦

在加入联邦之前须调用 RTW 代码的模型初始化部分对模型进行初始化,仿真开始时,需要通过 DtExerciseConn 产生演练,若演练已产生,则加入演练,其成员函数 drainInput() 负责与 RTI 通信、接收、发送数据。但构造 DtExerciseConn 时,必须提供联邦执行名和联邦成员名,同一联邦的联邦执行名和 FED 文件名必须相同^[5]。

```
char execName="VR-Link";//设定联邦执行名
char * fedName="Missilecontrol";//设定联邦成员名
DtExerciseConn exConn;
exConn=new DtExerciseConn(execName,
    fedName,
    new DtRprFomMapper(1,0),
    "VR-Link.fed");
```

2) 对象类的公布和订阅

演练产生以后,需对 FOM 中的对象类属性进行发布、订阅。若发布、订阅的是对象类中所有的属性,只需构造对象发布者(DtEntityPublisher)和映射实体列表(DtReflected-EntityList),也可以使用 DtObjClassDesc 的成员函数 publish()/subscribe() 来发布/订阅对象类的一部分属性。在导弹制导仿真系统中,每个联邦成员的识别是利用 EntiytType 来实现的。

```
DtEntityTypeMissilecontrol(1,2,225,1,2,0,0);
DtEntityPublisher entityPub;
entityPub=new DtEntityPublisher
(Missilecontrol,exConn,
DtDrOther, DtForceFriendly, DtEntityPublish-
```

```
er::guiseSameAsType());
```

3) 仿真过程中数据交互的实现

在仿真进行过程中,导弹控制模型接收雷达模型的数据,并进行计算输出导弹的方位角、俯仰角和导弹的坐标位置。为了让模型代码能通过 RTI 进行数据交互,将导弹模型的输入输出数据连接到 DtSetDataInteraction 交互类上,利用 DtSetDataInteraction 交互类将其产生的数据发送出去。在仿真开始时注册一个回调函数,当有雷达数据交互发生时就会调用回调函数接收雷达数据。在接收数据的时候,通过 DtEntityIdentifier 来识别是哪个联邦成员发送来的数据,当接收到雷达的数据后,调用 Sim_OneStep 函数计算导弹信息并推进仿真时间步长。以下函数实现数据的发送(发送导弹位置信息):

```
DtSetDataInteraction data;
DtEntityIdentifier Misslie(1,2,1);
data.setSenderId(Misslie);//设定数据标识
data.setNumFixedFields(3);
data.setDatumParam(DtFixed,1,DtDatumX-
pos);
data.setDatumValFloat32(DtFixed,1,position
[0]);
//设置 X 坐标的值
data.setDatumParam(DtFixed,2,DtDatumY-
pos);
data.setDatumValFloat32(DtFixed,2,position
[1]);
//设置 Y 坐标的值
data.setDatumParam(DtFixed,3,DtDatumZ-
pos);
data.setDatumValFloat32(DtFixed,3,position
[2]);
//设置 Z 坐标的值
exConn->sendStamped(data);//通过演练发
送数据
```

以下函数实现数据的接收(接收雷达测角信息):

```
DtSetDataInteraction::addCallback(exConn,
dataCb,NULL);//注册回调函数
```

(下转第 84 页)

LabWindows/CVI 为开发平台研制的弹体静态参数综合测试系统,充分发挥了虚拟仪器技术的优势,使系统具有人机界面友好、操作方便、功能完善、性价比高、可扩展性强的特点。系统在数据的多参数采集、存储、显示、信号处理、数据传输等方面显示了强大的性能优势。实际使用结果表明,该测试系统设计合理,结构简单,安装维护方便,能够满足国军标的精度要求。

参考文献:

[1] 陈贤隆,徐肖琪,丁彩云. 弹箭静态参数测量原理和技术[M]. 北京:宇航出版社,1993.
 [2] 王芳,朱蕴璞,徐肖琪. 基于 PC 的弹箭静态参数综合测试系统[J]. 弹箭与制导学报,2006,26(4): 166-168.
 [3] 张彦忠,周明光,宋志刚. 利用虚拟仪器技术构建

非电量测量系统[J]. 微计算机信息,2006(12): 164-165.

[4] 范金锁,张志成,张合新. 基于虚拟仪器的导弹控制系统测试研究[J]. 仪器仪表学报,2007(4):260-266.
 [5] 张毅刚,乔立言. 虚拟仪器软件开发环境 Labwindows/CVI6. 0 编程指南[M]. 北京:机械工业出版社,2002.
 [6] 赵会兵. 虚拟仪器技术规范与系统集成[M]. 北京:清华大学出版社. 北方交通大学出版社,2003.
 [7] Richard Boynton. Mass properties measurement handbook[Z]. SAWE Paper No. 2444 ,1998.
 [8] G Bucci, A German, C Landi. Virtual instrument based on real-time measurement[C]// Proc. of IMTC/ 94, Hammamatsu, Japan, 1994: 357-360.

(上接第 76 页)

```
void dataCb ( DtSetDataInteraction * inter,
void * user) //接收数据的回调函数
{ DtEntityIdentifierRadar(1,2,2);
  if(inter->senderId() == Radar)
  {
m__Azimuth = inter - > datumValFloat32 (Dt-
Fixed,1);
//接收方位角的值
m__Pitch = inter - > datumValFloat32 (Dt-
Fixed,2);
//接收俯仰角的值
}
}
```

5 结论

综上所述,通过将 Simulink 模型与 VC6. 0 混合编程并将其改造成 HLA 联邦成员,文中将对现有的 Simulink 模型与 VC6. 0 混合编程以及

改造成 HLA 兼容的系统所涉及的关键技术进行了研究,在实践上利用上述技术和 MAK 公司的 VR-Link 工具包开发了导弹控制联邦成员,使得 HLA 中所体现出来的可重用性和可互操作的思想得到卓有成效的支持。

参考文献:

[1] Using Simulink [DB/OL],Matlab Online Manual.
 [2] Real-Time Workshop User's Guide [DB/OL], Matlab Online Manual.
 [3] 杨涤,李立涛,杨旭,等. 系统实时仿真开发环境与应用[M]. 北京:清华大学出版社,2002.
 [4] 田新华,冯润明,温干飞,等. 一种将 Simulink/Stateflow 模型改造成 HLA 成员的方法[J]. 系统仿真学报,2002,14(7):883-886.
 [5] MAK Technologies. MAK VR-Link 3. 9. 6 Developer's Guide[M]. Cambridge, USA: MAK Technologies,2005.
 [6] 曹罗生. VC++ 与 Matlab/Simulink 混合编程的实现与研究[J]. 科学技术与工程,2007,7(21):5685-5687.