基于SysML与Simulink的飞控系统概念样机设计

刘兴华，曹云峰，王 彪，庄丽葵，周在华

(南京航空航天大学自动化学院 南京 210016)

【摘要】研究了一种基于SysML与Simulink的飞控系统概念样机设计方法。首先分析了SysML相对于UML的扩展及其比STATEMATE/UML更适合飞控系统概念样机设计的原因；其次，为了实现对飞控系统概念样机设计过程的完整支持，研究了一种实现SysML与Simulink模型集成及协同仿真的SysML扩展机制，分析了扩展机制需要解决的关键问题以及扩展机制的Profile定义；最后，以一无人飞行器飞控系统Predator为例，基于SysML与Simulink对其概念样机进行了设计，并进行了仿真验证。设计实践表明该方法能有效支持飞控系统概念样机设计。

关 键 词 概念样机；飞控系统；模型集成；系统建模语言

中图分类号 TP311； V24 **文献标识码** A doi:10.3969/j.issn.1001-0548.2011.06.016

Flight Control System Conceptual Prototype Design Based on SysML and Simulink

LIU Xing-hua, CAO Yun-feng, WANG Biao, ZHUANG Li-kui, and ZHOU Zai-hua

(Automation Engineering College, Nanjing University of Aeronautics & Astronautics Nanjing 210016)

Abstract A flight control system conceptual prototype design method based on integration of SysML and Simulink is researched. Firstly, the improvements between SysML and UML /STATEMATE that meet the flight control system design requirements are introduced. Secondly, a SysML extension mechanism that integrates SysML and Simulink model through code embedding is researched, the key problems it faces are analyzed and its profile definition is given. In the last, taking an unmanned aircraft system for example, the conceptual virtual prototype of its flight control system predator is designed, and functions, behaviors of predator are verified and validated. The design practice shows this method could meet the design requirements of flight control system conceptual prototype.

Key words conceptual prototype; flight control system; model integration; SysML

概念样机[1-3]是飞控系统数字化设计中论证与方案阶段的重要成果，其对应于传统飞控系统设计方法中以文档形式存在的设计方案[4]。采用形式化图形建模语言构造飞控系统概念样机是飞控系统数字化设计中采用的主要方式[2,5-7]，其形式化、可执行、可测试的特点使设计人员在早期即可通过运行样机(方案)消除其中可能存在的歧义、错误，并修改，以避免后期工程研制出现反复。与工程研制阶段的虚拟样机[8-10]不同，概念样机定位于飞控系统顶层设计阶段，即论证和方案阶段。国内外很多研究人员都对飞控系统概念样机进行了相关研究，如文献[2,6,10-11]基于STATEMATE研究了飞控系统概念样机，文献[7]基于UML研究了飞控系统概念样机。但概念样机设计需要建模语言/工具对论证和方案阶段提供全部支持，意味着建模语言/工具既要支持论证过程，又要支持概念样机架构及完整动态行为表达。建模工具/语言的不完整性导致以上研究工作都不能支持完整概念样机设计，如以上设计都缺乏对概念样机连续动态行为的描述，且基于STATEMATE的概念样机设计也缺乏对论证过程的支持。

目前，只有多种建模语言/工具协同，才能实现对飞控系统概念样机设计的完整支持。基于此，本文研究了一种SysML/Simulink的协同机制，并进行了飞控系统概念样机设计。

1 系统建模语言SysML

SysML[12-14]是对象管理组织OMG及国际系统工程学会INCOSE为满足系统工程领域建模需要，在

收稿日期：2010－06－02； 修回日期：2011－09－02
基金项目：国家自然科学基金(60974106)；江苏省333人才工程项目(P0994GX)
作者简介：刘兴华(1981－)，男，博士生，主要从事飞行器控制系统数字化设计等方面的研究.

重用UML2.0[12]子集基础上经过扩展而推出的系统工程领域图形化建模语言。

相比于UML，SysML在以下方面进行了改进：1) SysML的基本单元块(block)替代了UML的基本单元类(class)，与此相对应，UML中的类图、对象图和复合结构图被SysML中的块定义图和内部块图代替，前者用于描述系统之间的关联，后者用于定义系统内部结构。2) UML端口(port)被重命名为标准端口(standport)，并新增了流接口(flowport)，前者用于实现消息传递；后者实现信号等实体流传输。3) 增加了需求图和参数图，需求图描述需求之间以及需求与系统设计元素之间的关联，便于设计追溯，实现需求的形式化表达；参数图用于建立各种系统属性之间的参数关联，表征系统参数变化对系统总体功能、性能的影响。前两项改进使SysML比UML/STATEMATE能更全面地表达飞控系统概念样机，第3)项改进则使SysML更好、更完整地支持概念样机设计过程中的需求能形式化表达及系统性能分析等论证过程。

2 SysML与Simulink集成

SysML虽然能够对飞控系统论证过程提供更强大和全面的支持，但同STATEMATE/UML一样，SysML不能提供对飞控系统连续动态行为的描述，而连续动态行为描述是飞控系统概念样机设计必须解决的问题。因此，为实现完整支持飞行控制系统概念样机设计，需要研究SysML与连续行为建模语言之间的集成。鉴于Simulink是目前连续系统动态建模领域事实上的标准，本文研究了一种SysML与Simulink的集成机制。

2.1 SysML与Simulink集成原理

SysML和Simulink的底层运行机制使得通过代码嵌入实现两者集成成为可能，但需要解决3个关键问题：1) 模型代码的动态交互方式。SysML模型与Simulink模型在开发平台运行时是两个不同的独立线程，要实现两者集成，必须对两者的仿真进行同步。研究两者的模型代码可发现将Simulink代码进行适当修改并嵌入SysML模型代码中，利用SysML模型运行环境对两者的运行进行同步是一种有效可行的解决方案。2) 模型数据接口。SysML模型与Simulink模型协同仿真运行时不仅需要同步，还需要进行数据通信。对协同仿真需要传递的数据变量在Simuilnk模型中以输入输出接口方式进行定义，并在SysML模型中定义同名流端口以及同名同类型变量，就可实现SysML模型与Simulink模型的无缝数据传输及代码中变量的无缝衔接。3) 数据类型识别。Simulink模型代码中存在许多RTW定义的专用数据类型，需在SysML代码中对这些新类型进行声明，避免出现数据无法识别的问题。

解决以上3个问题后即可实现Simulink模型代码与SysML模型代码之间的集成。SysML模型与Simulink模型通过代码集成的原理示意图如图1所示。图中，SysML模型与Simlink模型之间的交互(双向虚线)通过SysML模型代码对Simulink模型代码的调用(双向实线)实现。SysML模型与代码之间的双向实线代表模型和代码之间的动态关联。

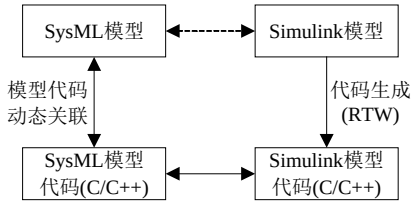

图1 基于代码的Simulink与SysML集成

2.2 SysML与Simulink集成Profile

为方便SysML与Simuink模型集成，对2.1节集成机制关键问题的解决方案以SysML扩展profile进行定义，通过应用该profile可方便实现SysML与Simulink的无缝集成。表1给出了该profile的组成成员及其成员扩展类型。

表1 SysML Profile组成

扩展类型	成员名称
stereotypes	Simulink Block
tags	Simulink Variable Name
types	boolean_T ⋮ pointer_T

表2给出了表1中SimulinkBlock衍型(stereotype)所包含标识(tags)的详细定义，其中，IncludePath指定Simulink模型代码运行时需要的头文件路径；Matlab-Root指定Matlab软件安装根目录；SimulinkProjectFile指定Simulink模型文件路径；SimulinkSampleTime指定SysML模型与Simulink模型之间进行数据交互的时间间隔；SimulinkCodeDir指定Simulink模型的RTW代码路径；SimulinkSourceFiles指定Simulnk模型RTW代码文件中需要导入到SysML模型中的文件名称。

表2　SimulinkBlock衍型定义

扩展类型	成员名称
tags	IncludePath
	MatlabRoot
	SimulinkProjectFile
	SimulinkSampleTime
	SimulinkCodeDir
	SimulinkSourceFiles

3　基于SysML与Simulink的概念样机

飞控系统设计采用数字化设计方法后，传统设计流程中顶层的论证与方案阶段即变成数字化设计流程中的概念样机设计阶段。在该阶段，SysML的9种图及Simulink分别发挥不同的作用。有的用于概念样机设计的需求分析及系统分析过程，如用例图和需求图；有的用于最终飞控系统概念样机表达，如块定义图、状态图和Simulink。它们相互协作配合共同支持完整飞控系统概念样机设计。

概念样机的表达一般从结构、功能、行为及性能等角度进行。表3给出了SysML和Simulink对飞行控制系统概念样机表达的支持，其中块定义图和内部块图用于定义飞控系统静态模型(系统结构)，前者描述子系统之间的关联，后者描述子系统内部的组成结构。活动图、状态图和Simulink用于定义系统的动态模型，前两者用于描述离散状态行为，后者用于描述连续动态行为。静、动态模型相互配合协作，共同构成飞控系统概念样机，并通过运行体现系统的功能和初步性能。

表3　SysML与Simulink对概念样机表达支持

概念样机视图	模型表达支持
静态模型(结构)	块定义图
	内部块图
动态模型 (功能、行为及性能等)	状态图
	活动图
	Simulink

图2给出一个由块定义图描述的概念样机实例。该系统由3个SysML块A、B、C构成。其中块A、B具有离散状态行为，块A、B的右上角小图标代表该块的动态行为通过状态图[13]/活动图描述；块C具有连续动态行为，将块C的stereotype设置为SimulinkBlock后，可采用Simulink对其连续行为进行描述。图3给出了块C动态行为的Simulink描述，块C的两个流端口H_g和H对应于图4中的输入输出端口，在将图4的Simulink模型导入C块时，这两个流端口会自动产生，数据类型为real_T。

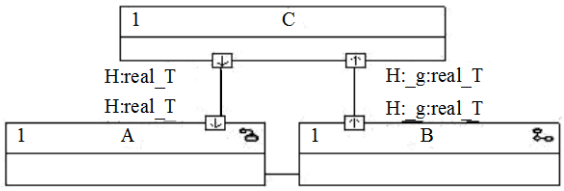

图2　概念样机示例

图4给出了该概念样机的底层代码结构，其中框A为对应于概念样机模型SysML部分的代码，框B为对应于Simulink部分的代码，框C为实际构成概念样机的底层代码。从图中可以看出，Simulink代码中的ert_main函数不包含在概念样机中，其功能被SysML模型中stereotype指定为SimulinkBLock的类C替代。

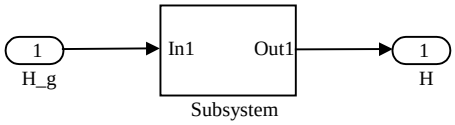

图3　块C连续行为描述

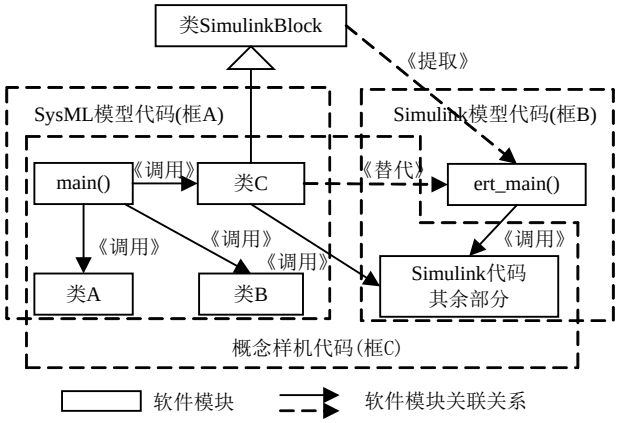

图4　概念样机代码结构

类SimulinkBlock是ert_main功能的抽象，在将SysML中块C的stereotype指定为SimulinkBLock后，该块对应的底层代码类C会自动继承类SimulinkBlock，从而便于在类C中增加替代ert_main的代码。以下为类SimulinkBlock中主要部分关键函数的定义：

```
class SimulinkBlock
{
    virtual void doStep() = 0;
    virtual bool startBehavior();
    int getM_nSampleTime() const;
    void setM_nSampleTime(int p_m_nSampleTime);
    void initStatechart();
    void cancelTimeouts();
    bool cancelTimeout(const IOxfTimeout* arg);
}
```

4 飞行控制系统概念样机设计

形式化、可执行、可测试的特点使得采用概念样机后的系统顶层设计不再是传统的瀑布式过程，而是一个边设计、边运行、边调试的迭代递增式开发过程。本文以一无人飞行器飞控系统Predator为例，详细介绍其概念样机设计(该飞行器控制律假设之前已经设计完成)，但为了叙述方便，按照瀑布式过程进行描述。

4.1 需求分析

需求分析阶段首先基于用户需求建立用户需求模型，随后对用户需求进行分析和细化，导出系统需求并建立系统需求模型，最后基于系统使用方式和系统需求模型，建立系统用例模型。通过该阶段的分析，明确用户需求及其对应的系统需求，并将此作为后续系统分析与设计的依据。该阶段的输出是用户需求模型、系统需求模型和系统用例模型。

4.2 系统分析

系统分析阶段以系统模型中的用例为入口点，依据用例分析系统应该具有的属性、操作及行为等特性，使用块定义图及顺序图、活动图或状态图构建用例黑盒可执行模型，并通过仿真验证设计的系统特性是否满足用例及其背后的系统和用户需求。

待所有用例的黑盒可执行模型设计完毕后，合并各用例特性得到系统特性模型，并通过参数图建立系统特性之间的关联，得到系统参数模型。该阶段的输出是系统特性模型和参数模型。

4.3 系统设计

系统设计阶段以系统分析阶段确定的系统特性模型和系统参数模型作为入口点，首先基于WOM方法确定系统架构，随后将系统特性模型中的系统特性根据需要分配至各个子系统块并对块的端口、接口及行为等进行优化设计，得到描述飞控系统结构、功能、行为及初步性能等的完整概念样机，并通过模拟运行验证样机设计是否正确以及是否满足背后系统需求和用户需求。

图5展示了表达Predator概念样机顶层架构的块定义图。Predator顶层由6个块构成，其中GPS、角速率陀螺、气压高度/空速传感器等块的粗边框表示这些块是主动(active)块，类似于独立线程。块飞控计算机的并发机制由图6所示的飞控计算机内部块图中的几个主动块表达，对应于飞控软件内部的多线程。各个块根据其特性不同分别采用状态图或Simulink描述其动态行为，如GPS等块右上角的图标显示其动态行为由状态图描述，块横侧向控制单元和纵向控制单元的动态行为由Simulink描述，分别代表横侧向和纵向控制器，它们根据块控制算法解算单元的指令实现各种控制模态的切换及控制指令的输出。图6下方的几个块表达了系统的航路数据库信息。另外各执行机构模块由于其动态特性一般采用Simulink描述，在Predator概念样机中将其与飞行器动力学模型合并为一体，因此图中没有给出。

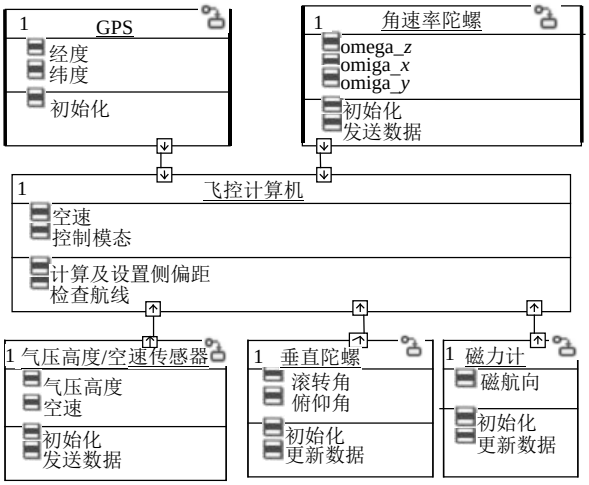

图5 Predator块定义图

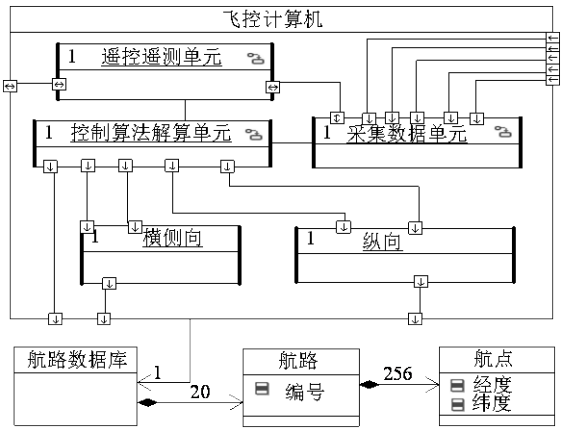

图6 飞控计算机内部块图

4.4 检查及仿真验证

概念样机在飞控系统数字化设计中的作用类似于传统设计方法的设计方案，只有经过检查及仿真验证，确认其功能、行为等满足设计要求后设计工作才能转入随后的工程研制。飞控系统概念样机的检查分为两个层次。

1) 检查基于SysML/Simulink语法检查器进行，主要检查概念样机是否满足建模语言的语法要求以及是否运行正确，并对概念样机中存在的警告和错误进行提示和定位。

2) 检查通过运行概念样机验证系统的功能、行为及初步性能，且一般采取“自底向上”的策略，即

先检查各底层块如GPS、气压高度/空速传感器、采集数据单元的功能、行为是否正确，然后验证由底层块采集数据单元、横侧向控制单元等构成的父块飞控计算机的功能和行为是否正确，最后引入动力学模型进行完整系统的功能、行为和初步性能的闭环验证。

第二层次仿真验证时，图6中块控制算法解算单元的运行状态图(部分状态边框被加粗显示)如图7所示。该图显示Predator飞行器当前处于爬升状态(边框被加粗显示)。概念样机仿真运行时纵向飞行航迹的部分阶段如图8所示。图中，*L*点之前Predator处于“自主导航”状态，目标航点高度为1 000 m，到达*L*点后，操纵人员发出“爬升2”指令，Predator系统纵向退出“自主导航”并进入“爬升2”固定的俯仰角爬升，爬升至*M*点后，操纵人员发出“自主导航”指令，Predator以上次退出自主导航时的目标航路点为当前目标航路点并重新进入“自主导航”状态，表现为在*MN*段Predator高度下降至1 000 m，并保持定高飞行，到达当前目标航路点(到达目标航路点方圆50 m内即认为到达目标航点)后，以下一个航路点为目标航路点继续自主导航飞行，爬升至该目标航路点的高度1 200 m后保持定高飞行，直至到达该目标航路点。

通过两个层次的仿真后，可验证所设计样机(方案)的功能、行为及初步性能满足设计需求，可确保设计方案中没有歧义和错误，设计工作可以转入下一步的工程研制阶段。

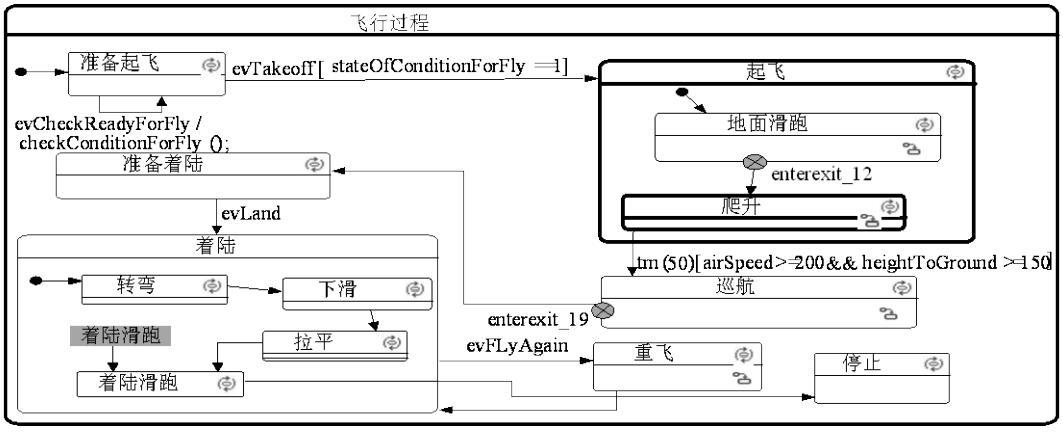

图7　块控制算法解算单元的运行状态图

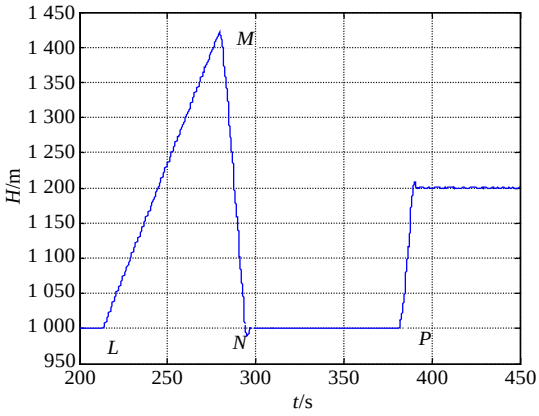

图8　Predator纵向飞行航迹片段

5　结　论

本文研究了一种基于SysML与Simulink的完整飞控系统概念样机的设计方法，通过设计概念样机，使设计人员在早期即可对设计方案进行评估，消除其中的歧义和错误，避免后期工程研制时出现反复。该方法较原有基于STATEMATE和UML的飞控系统概念样机设计方法，可对飞控系统概念样机设计提供更全面、更强大的支持。

参　考　文　献

[1]陈宗基，黄浩东，秦旭东. 飞行控制系统虚拟原型技术[J]. 航空学报, 2002, 25(2): 441-447.
CHEN Zong-ji, HUANG Hao-dong, QIN Xu-dong. Virtual prototyping for flight control systems[J]. Acta Aeronautica Et Astronautica Sinica, 2002, 25(2): 441-447.

[2] 吴建民，敬忠良，肖刚. 基于模型仿真的航空电子系统螺旋式开发方法[J]. 系统工程与电子技术，2007，29(3): 488-491.
WU Jian-min, JING Zhong-liang, XIAO Gang. Spiral development for avionics based on prototyping[J]. Systems Engineering and Electronics, 2007, 29(3): 488-491.

[3] WANG G G. Definition and review of virtual prototyping[J]. Journal of Computing and Information Science in Engineering, 2002, 2(3): 232-236.

[4] 王永熙. 飞行控制系统与液压系统设计[M]. 北京: 航空工业出版社, 2003.
WANG Yong-xi. Flight control system and hydraulic system design[M]. Beijing: Aerospace Industry Press, 2003.

(下转第910页)