

支持MBSE的系统全过程设计 应用框架研究与实现

胡峻豪, 冯 雷, 朱 睿, 李荣强

(中航工业成都飞机设计研究所, 成都 610091)

[摘要] 针对当前工程系统设计中需求和功能分析多基于文档,难以与物理模型交互进行系统整体仿真验证的问题,研究了基于模型的系统工程(MBSE)方法论和功能模型接口(FMI),提出了一种支持MBSE的系统全过程设计应用框架,并基于该框架设计了某型飞行器舵机伺服系统。结果表明,该框架能满足从需求、功能分析到物理仿真的全过程模型实现和联合仿真,提高了系统设计效率,降低了反复迭代次数,适用于大型、复杂系统的全过程设计建模与仿真。

关键词: 基于模型的系统工程; 功能模型接口; 功能模型; 物理模型

DOI: 10.16080/j.issn1671-833x.2016.13.105



胡峻豪

硕士,毕业于西北工业大学航天学院,现就职于中航工业成都飞机设计研究所,研究方向为基于模型的系统工程、多学科联合仿真。

由于在成本和效率方面的巨大优势,计算机仿真已成为当前工程研制的必要环节。但目前的计算机仿真多集中在物理仿真层面,通过建立系统物理模型,模拟系统工作原理,分析系统性能;对于前期需求与功能逻辑建模重视不够,无法将系统的需求、功能和性能作为一个有机整体来分析、验证,从而造成顶层设计与专业工程设计间反复迭代。

针对上述情况,近年兴起了基于模型的系统工程(Model-Based System Engineering, MBSE)方法论^[1],在分析系统需求的基础上,采用SysML和UML作为描述系统功能和行为的建模语言,使用模型状态机精确描述系统的功能逻辑和故障处理模式,建立系统的功能模型,并通过模型的仿真、验证,分析需求的满足程度和系统的功能逻辑流转,方便设计人员在系统设计早期验证设计的全面性和正确性。

如果能将系统的功能模型和物

理模型联合起来,则可用系统功能模型设计系统行为处理逻辑和故障处理模式,用其结果驱动物理模型,并根据物理模型仿真结果改善功能模型,实现基于系统功能、物理模型的确认、验证,大大降低设计成本、缩短研制周期。

但是,由于系统建模语言与传统物理建模语言存在很大差异,导致同一系统的功能模型和物理模型难以交互和协作。Carlioni等^[2]通过详细分析和比较常用工具的语义规范,提出了HSIF(Hybrid Systems Interchange Format),用以调解工具间差异,但该方法要求建模工作者深入了解软件机理,专业需求较高;Schamai等^[3]提出直接使用Modelica建立UML的状态机,但该机制的状态机在处理可用事件时,只能采用全并行的方式,对于仿真时拍不好控制;Sakairi等^[4]尝试使用S函数将Simulink模型导入状态机,实现了物理模型和功能模型的联合仿真,但专

用性太强,不具有代表性。

基于此,本文选择功能模型接口 FMI (Functional Mockup Interface) 标准协议作为桥梁,利用通用标准接口实现功能模型和物理模型的信息交互,提出了从系统需求分析、功能建模、物理建模到最终联合仿真的支持 MBSE 的系统设计应用框架。基于此框架,设计人员在进行系统设计和验证时,只需专注于模型本身,而不用去分析工具底层的差异。同时,由于打通了功能模型和物理模型的壁垒,在工程样机之前即可对系统进行全面验证,减少迭代次数。

最后,运用此设计框架设计了某飞行器的舵机伺服系统,并对设计结果进行了仿真分析,结果证明该系统设计框架能够快速设计舵机伺服系统,并能对其进行全面的功能、行为和性能验证。

支持 MBSE 的系统设计应用框架

支持 MBSE 的系统设计应用框架分为 3 层: 功能模型层,主要描述系统的功能逻辑行为; FMI 协议层,主要用于模型的封装和协议标准化,实现不同建模工具和语言间的模型交互; 物理模型层,主要用于实际系统的物理建模和仿真,支持 MBSE 的系统设计应用框架如图 1 所示。

功能和物理模型之间关系采用松耦合方式关联: 功能模型不考虑物理模型的具体结构和实现,只需关注系统的功能逻辑; 而物理模型也只用考虑真实物理系统工作原理,两者通过互留的接口进行交互。因此,功能和物理模型可分别设计,并行工作,最后通过 FMI 协议层进行综合。

此外,由于 FMI 协议的通用性,既可以将功能模型封装导入物理模型,也可以将物理模型封装导入功能模型。也就是说,在设计过程中,既可以自顶向下正向设计,也可以自底向上逆向验证、完善。

1 功能模型层

本层的主要工作是采用 MBSE 方法论对系统进行需求分析和功能建模。在需求分析和管理方面,目前已有许多成熟的工具,常见的有 CaliberRM、RequisitePro、Doors 等。其中,CaliberRM 侧重于分布式开发团队协作开发,提高系统设计效率,但不支持对已有文件和信息的重用与共享; RequisitePro 侧重于改进项目团队的沟通协调,增强协作开发能力,但需求追踪和需求变更功能较差; Doors 则是基于整个项目的需求管理系统,用来捕捉、链接、跟踪、分析、管理信息,确保项目与需求和规范的一致性,是面向管理者、开发者、追踪用户及整个生命周期的综合需求管理套件。

功能建模方面,通常使用系统建模语言 (Systems Modeling Language, SysML) 来创建系统结构、行为、需求和约束的模型。主要建模方法有 Weikiens 系统建模 (System Modeling, SysMOD) 方法、INCOSE 面向对象系统工程方法 (Object Oriented System Engineering Method, OOSEM) 和 IBM Telelogic Harmony-SE 方法^[1]。其中,IBM 针对其 Harmony-SE 方法配备了较成熟的建模工具—Rhapsody,该软件能很好地覆盖 Harmony-SE 方法的整个设计周期,已成为功能建模的主流选择。

本层设计的具体实现过程如下:

在需求分析阶段采用 Doors 对需求进行条目化管理,并利用 Rhapsody Gateway 将需求导入到 Rhapsody 进行需求分析,得到系统需求分析模型,并抽取出用例。在用例基础上进行功能分析和架构分析,建立系统活动图、顺序图,定义出系统应具备的属性、操作和行为,并细分出系统架构; 其次,根据角色间交互关系,生成对应端口,将得到的上述属性、操作和行为分配到相应的端口,得到系统的 IBD (Internal Block Diagram)。这里需要注意的是: 为满足后续 FMI 的数据传递要求,此处定义的端口须是流端口 (FlowPort) 且只能为单向传递,各属性必须有初值。最后,建立系统状态图。可在每个状态内通过代码实现状态逻辑判断和处理,并通过状态间的切换、流转反映系统的功能行为逻辑。将之前得到的属性、操作和行为合理分配,分别作为状态出、入口动作,状态切换触发器和分支判定语句,并需要根据细化和补充属性、操作和行为信息,最终得到可运行的状态机。至此,完成系统功能模型的构建。

对于逻辑清晰、结构简单的系统,可以省略绝大部分操作,直接定义出系统的输入、输出接口和状态切换逻辑,建立系统的 IBD 和状态图,即可获得系统功能模型。

2 FMI协议层

本层主要功能是实现模型的

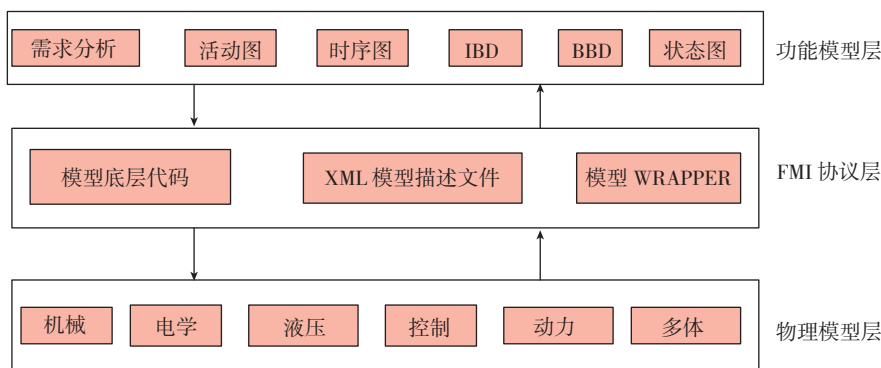


图1 支持MBSE的系统设计应用框架

Fig.1 System design application framework for MBSE

FMI封装。FMI封装标准制定了模型描述格式和数据存储格式的定义,改善了以往不同仿真软件拥有各自的专用接口造成的联合仿真困难。不同仿真工具遵循同一种封装接口标准,不仅能够实现仿真工具之间的交互仿真,还能使联合仿真工具解析模型描述信息的效率有所提升。基于FMI标准封装的仿真模型称为FMU(Functional Mockup Unit)。将FMU导入仿真平台中,可有效地完成多专业联合仿真,提高模型的通用性及重用性^[5]。

FMI标准定义了两种模型封装形式^[6]:“model exchange”和“co-simulation”。两者最大的区别是“model exchange”需要仿真工具提供常微分方程(Ordinary Differential Equations, ODE)来执行仿真,因此模型需要暴露出所有的内部方程;“co-simulation”自带模型求解器,不需要仿真工具额外提供仿真算法。

这样看来,“co-simulation”形式似乎更方便、实用,但该方式在处理离散事件时效率不高。这是因为系统的功能模型多是基于离散事件表达的,而“co-simulation”API并不能像“model exchange”API那样进行下一事件时间报告。所以,本文最终选择“model exchange”方式进行模型的封装^[7-8]。

封装函数基于QTronic提供的FMU SDK调用其现成的FMI API,使用建模软件生成的系统模型代码,封装上满足FMI标准的“wrapper”,其详细实现过程如下。(1)定义输入、输出接口和内部变量。找到并列列出所有需要与外界进行交互的接口,并根据信息的流向分别定义为输入、输出端口,同时对端口的规范性进行检查。(2)创建FMU模型描述文件。按照FMI标准定义的XML文件模板,对第一步中获得的信息进行填充,创建出FMU的模型描述文件:modelDescription.xml,主要用于描述

FMU的接口信息。(3)生成系统模型标准代码。调用模型软件的代码生成器,生成待导出的系统模型代码。(4)创建FMU“wrapper”。将第3步中的模型代码按照FMI标准实现。FMU“wrapper”支持FMI接口并将其转化为模型软件对应的接口,在“wrapper”上有对FMU响应的各变量以及一系列内部变量(如当前仿真时间等)。绝大多数的FMI函数主要用于读取或设置这些变量值,最主要的工作通过fmiEventUpdate函数实现,其具体实现过程如下:根据最近一次调用fmiSetTime获得的时间来设置模型软件时间;根据先前的fmiSetXXX(变量名)来设置块输入变量;调用模型软件生成的代码来执行一个行为步长;更新“wrapper”上FMU输出所对应的变量;将下一事件时间设为最早的超时门限。(5)编译并封装FMU。将上述步骤中的模型代码和模型描述文件按FMI标准封装为.fmu文件。

3 物理模型层

本层的主要功能是搭建系统的物理模型,模拟系统的真实运行场景,获取系统的各项性能指标。

当前各专业的仿真软件有很多,分别为针对某一个或几个领域的问题进行建模仿真,如用于多体动力学建模的ADAMS、控制仿真的Matlab/Simulink、机械液压的AMESim等,但随着当前系统的规模和复杂度越来越高,单一的专业仿真软件已无法胜任完整的系统建模工作^[9]。

基于此,本文选择了统一建模语言Modelica。它采用面向对象和组件的思想,对不同领域物理系统的模型进行统一表述,实现统一建模,并且建立的模型能真实反映系统的拓扑结构。同时,其基于方程的非因果建模方式极大地减轻了建模工作量(尤其是复杂系统),还可避免因公式转换和推导引起的错误,保证了模型的健壮性。此外,Modelica还支持层

次结构建模,语言本身带有可重用的机械、电子、液压、控制、热流等领域的标准库和扩展库,用户还可建立自定义的领域模型库,实现领域知识的双重用。

当前,基于Modelica的建模仿真工具层出不穷,常见的有Dymola、OpenModelica和SimulationX。其中Dymola作为Modelica的原生工具,拥有最及时的Modelica原生库和建模规范更新,是最纯粹的Modelica建模工具;OpenModelica是基于Modelica语言的开源软件,其最大的好处是免费且安装方便,能满足基本的建模需求,但模型库匮乏、功能简单,可看作Dymola的精简版;SimulationX是德国ITI公司开发的基于Modelica的商业建模软件,不仅有完整的Modelica模型库,还提供了丰富的自主开发的商业模型库,并提供二次开发平台TypeDesigner,用户不仅可以直接对SimulationX所有模型进行修改,还可以基于Modelica语言创建新的模型,并能够把用户自己的C代码模型以图形化模块的方式集成进SimulationX软件包,能满足大型、多物理系统的建模需求。SimulationX还支持FMI协议,满足本设计框架的联合仿真需求。因此本文选择SimulationX作为物理建模层的建模和仿真工具。

某飞行器舵机伺服系统应用验证

基于上述应用设计结构,设计了某飞行器舵机伺服系统^[10-11]。考虑到舵机伺服系统主要逻辑判断和处理在舵机控制器(Actuator Control Equipment, ACE)进行,因此功能模型主要针对ACE子系统,其与作动器的交互通过消息传递,物理模型主要针对物理作动、执行单元,功能和物理设计同步进行,最后通过FMI协议层进行模型汇总和联合仿真。

具体过程如下:

首先,采用 Doors 对舵机伺服系统的需求进行条目化管理,并利用 Rhapsody Gateway 将其导入到 Rhapsody 进行需求分析,得到该伺服系统的需求分析模型,如图 2 所示。

其次,根据第 1 节中所述方法构建系统功能模型,其接口 IBD 如图 3 (a) 所示;系统的具体功能行为和故障处理逻辑通过状态流转及状态处理算法(写在各状态中)实现,最终状态如图 3 (b) 所示。

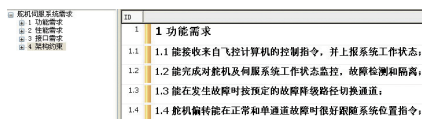
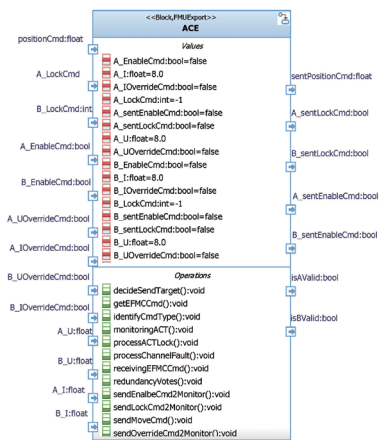
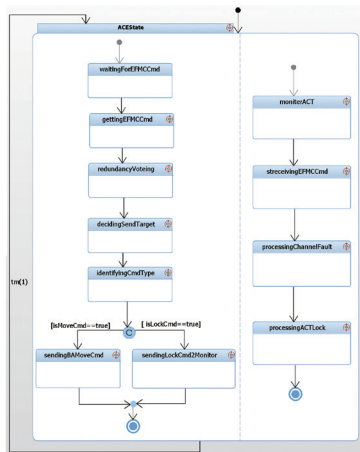


图2 系统需求分析模型

Fig.2 System requirement analysis model



(a) 系统 IBD



(b) 系统状态图

图3 系统功能模型

Fig.3 System functional model

物理模型层则是采用 SimulationX 构建舵机执行机构的物理模型,该执行机构使用“主主式”工作方式:当单通道发生故障时,另一通道接管故障通道职能,保证系统正常运行。对系统抽象、简化后模型如图 4 所示。

功能和物理模型设计完成后,即可通过 FMI 协议层进行模型的汇总和联合仿真。具体做法如下:将功能模型,通过 FMI 协议层封装为 FMU 模型,作为带接口的功能块导入到

SimulationX 中,并将其接口与相应的物理模型连接,得到系统的联合仿真模型,如图 5 所示;最后,给出相应的初始信号,设置好仿真求解器和步长,即可进行功能和物理模型的联合仿真。图 6 为双通道正常模式下联合仿真结果:从图 6 (a) 可以看出,当 A、B 通道有效性值在整个仿真过程保持为 1 (即正常)时,舵机位置曲线(具体位置信息已作归一化处理)如图 6 (b) 所示,能很好地跟随输入指令曲线。

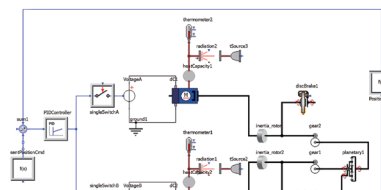


图4 系统物理模型

Fig.4 System physical model

图 7 为单通道故障模式仿真图。从图 7 (a) 中可以看出,在仿真时间 40~70s 时,令 B 通道故障(即 B 通道有效性值为 0),舵机位置响应曲线(如图 7 (b) 所示)在故障切换时,发生抖动并很快恢复正常,满足单通道故障模式下的位置跟随。

由以上仿真分析可以看出,舵机在正常和故障模式下都能很好地响应位置指令,并且能在发生故障时按要求进行模式切换,在模式切换的瞬间有振颤,并且很快恢复正常,及时

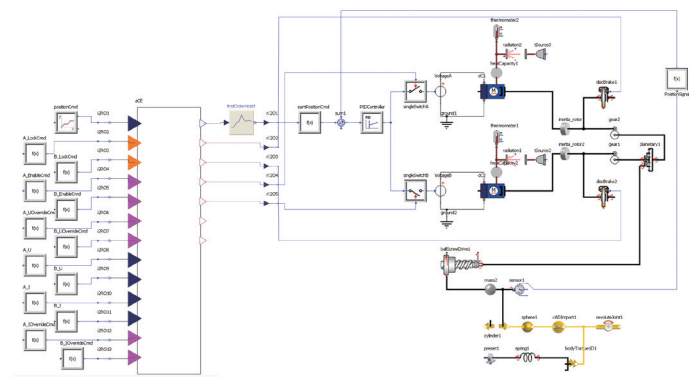
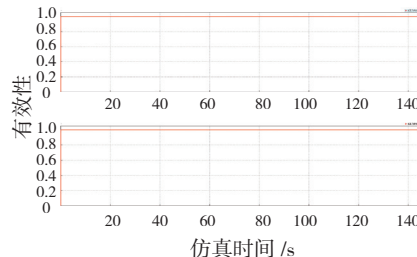
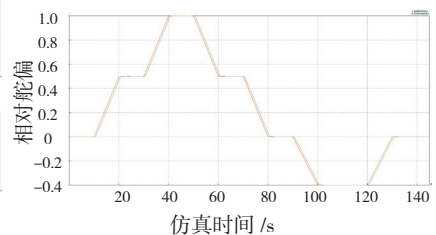


图5 联合仿真模型

Fig.5 Co-simulation model



(a) AB 通道有效性图



(b) 舵机位置响应曲线

图6 双通道正常模式仿真

Fig.6 Dual-channel normal simulation

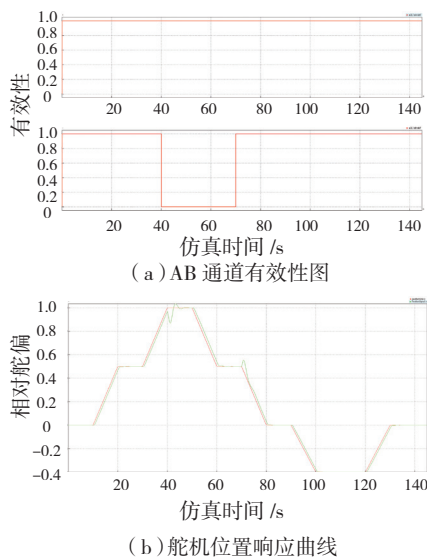


图7 单通道故障模式仿真

Fig.7 Single-channel failure simulation

跟随位置指令,达到预期设计需求。

结束语

本文提出的支持MBSE的系统设计应用框架为大型、多物理领域系统提供了整体设计思路,使设计人员在初步设计阶段即可对系统进行全面的需求、架构、功能和性能仿真验证,覆盖从需求到物理元件的设计全过程,可避免后期工程物理样机的研制出现反复,从而达到降低设计成本、提高效率的目的。

参考文献

- [1] HOFFMANN H P. Model-based systems engineering best practices[M]. Beijing: Aviation Industry Press, 2011: 1-62.
- [2] CARLONI L P, PASSERONE R, PINTO A. Languages and tools for hybrid systems design[J]. Foundations and Trends in Electronic Design Automation, 2006, 1(1/2):1-193.
- [3] SCHAMAI W, POHLMANN U, FRITZSON P, et al. Execution of UML state machines using Modelica[C]//The 3rd International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools, Atlanta, 2010.
- [4] SAKAIRI T, PALACHI E, COHEN C, et al. Designing a control system using SysML and Simulink[C]//2012 Proceedings of SICE Annual Conference (SICE). IEEE Electronic Library, 2012: 2011-2017.
- [5] BROMAN D, BROOKS C, WETTER M. Determinate composition of FMUs for co-simulation[C]//EMSOFT '13 Proceedings of the Eleventh ACM International Conference on Embedded Software Article No.2, IEEE Electronic Library, 2013:1-12.
- [6] 吴紫俊,赵建军.多领域功能样机可交换模型规范实现研究[J].系统仿真学报, 2012,24(10):2083-2086.
- WU Zijun, ZHAO Jianjun. Research on functional mock-up interface for multi-domain model exchange[J]. Journal of System Simulation, 2012,24(10):2083-2086.
- [7] 吴义忠,刘敏,陈丽平.多领域物理系统混合建模平台开发[J].计算机辅助设计与图形学学报, 2006,18(1):120-124.
- WU Yizhong, LIU Min, CHEN Liping. Development of hybrid modeling platform for multi-domain physical system[J]. Journal of Computer-aided Design & Computer Graphics, 2006, 18(1):120-124.
- [8] MULLER W, WIDL E. Linking FMI-based components with discrete event systems[C]//Proceedings of 2013 IEEE International Systems Conference (SysCon). Orlando, 2013:676-680.
- [9] 王西超,曹云峰,庄丽葵,等.面向复杂系统虚拟样机系统建模的方法研究[J].电子科技大学学报, 2013,42(5):648-655.
- WANG Xichao, CAO Yunfeng, ZHUANG Likui, et al. Collaborative modeling approach for virtual prototype of complex systems[J]. Journal of University of Electronic Science and Technology of China, 2013,42(5):648-655.
- [10] 张志强,高金行,李俊霖.冗余舵机伺服系统设计[J].机械与电子, 2012(11):23-25.
- ZHANG Zhiqiang, GAO Jinxing, LI Junlin. Design of dual redundancy rudder servo system[J]. Machinery & Electronics, 2012(11):23-25.
- [11] GHEORGHE A, ZOLGHADRI A, CIESLA C, et al. Model-based approaches for fast and robust fault detection in an aircraft control surface servo loop: from theory to flight tests[Application of Control[J]. IEEE Control Systems, 2013, 33(10):615-630.

Research and Implementation of System Design Application Framework for MBSE

HU Junhao, FENG Lei, ZHU Rui, LI Rongqiang

(AVIC Chengdu Aircraft Design & Research Institute, Chengdu 610091, China)

[ABSTRACT] Nowadays, requirements and functional analysis are mainly based on documents. The fact remains that physical model interaction deficiency exists in current system design, which leads to increasing burden on engineers to design and verify the system thoroughly. To solve this problem, MBSE is studied and FMI is adopted to merge systems functional models with physical models, and a system design application framework for MBSE which covers all phases of system design is proposed. In the end, an example is presented to demonstrate how to apply this framework to design and simulate servo systems. It is concluded with its high compatibility to massive modeling and complex systems, this framework also contributes effectively to meeting all the systems requests from requirements to physical units, raising system designing efficiency, and reducing iterations.

Keywords: MBSE; FMI; Functional model; Physical model

(责编 古京)