

基于 X 语言的起飞场景民机协同设计与仿真一体化方法

古鹏飞^{1,2}, 张霖^{1,2*}, 陈真^{1,2}, 叶俊杰³

(1. 北京航空航天大学, 北京 100191; 2. 复杂产品先进制造系统教育部工程研究中心, 北京 100191;
3. 中国航空工业集团公司雷华电子技术研究所, 江苏 无锡 214063)

摘要: 针对大型复杂产品, 传统基于模型的系统工程(model-based system engineering, MBSE)方法存在系统设计和仿真验证的多语言、多平台集成实现, 难以保证高效、准确地反馈系统设计的缺陷, 无法达到快速优化设计目的等问题, 提出采用基于面向复杂系统支持 MBSE 的新一代一体化建模仿真语言—X 语言实现面向起飞场景的民机跨域子系统的一体化建模仿真。从民机起飞过程需求分析出发, 建立了起飞场景的系统级模型和物理级模型, 并在 X 语言的建模仿真软件—XLab 进行统一的建模仿真验证, 给出了基于 X 语言面向复杂产品协同设计与仿真一体化方法, 为不同领域的设计人员实现复杂产品设计的协同与优化提供了全新的理论与方法参考。

关键词: 复杂产品; X 语言; 基于模型的系统工程; 建模仿真; 协同优化

中图分类号: TP391.9 文献标志码: A 文章编号: 1004-731X (2022) 05-0929-15

DOI: 10.16182/j.issn1004731x.joss.22-0104

Collaborative Design and Simulation Integrated Method of Civil Aircraft Take-off Scenarios Based on X Language

Gu Pengfei^{1,2}, Zhang Lin^{1,2*}, Chen Zhen^{1,2}, Ye Junjie³

(1. Beihang University, Beijing 100191, China;
2. Engineering Research Center of Complex PMBSE Product Advanced Manufacturing System, Ministry of Education, Beijing 100191, China;
3. Aviation Industry Corporation of China Lei Hua Institute of Electronic Technology, Wuxi 214063, China)

Abstract: For the large and complex products, the current traditional model-based systems engineering (MBSE) method of the integrated implementation of multiple modeling and simulation languages and platforms for system design and simulation verification can not ensure the efficient and accurate feedback of system design to realize the quick design optimization. *X language, a new generation of integrated modeling and simulation language based on complex systems and supporting MBSE, is used to realize the integrated modeling and simulation on cross-domain subsystems of civil aircraft for the take-off scenarios.* From the demand analysis of the take-off process of civil aircraft, the system-level model and physical-level model of the take-off scenarios are established, and the unified modeling and simulation verification is carried out by XLab, an X-language-oriented modeling and simulation software. The integrated method of collaborative design and simulation for complex products based on X language provides a new theory and method reference for the designers in different fields to realize the coordination and optimization of the complex product design.

Keywords: complex products; X language; model-based system engineering (MBSE); modeling & simulation; collaborative optimization

收稿日期: 2022-02-12 修回日期: 2022-04-14

基金项目: 国家重点研发项目(2018YFB1701600)

第一作者: 古鹏飞(1995-), 男, 博士生, 研究方向为基于模型的系统工程、模型工程、云制造。E-mail: by2003151@buaa.edu.cn

通讯作者: 张霖(1966-), 男, 博士, 教授, 研究方向为复杂系统建模仿真、云制造、模型工程等。E-mail: johnlin9999@163.com

引言

飞机是一类典型的系统组成复杂、产品技术复杂、制造过程复杂、工作环境复杂的复杂产品^[1]。由于生产制造过程中涉及多学科、多领域的耦合，导致其设计难度大，质量要求高等特点。早期的飞机设计制造过程，需建立真实的物理样机验证设计的正确性与合理性。然而，这种设计模式会显著降低设计研发效率，代价过高。20世纪90年代以来，随着科学技术的进步，特别是 CAD/CAE(computer aided design/computer aided engineering)等建模与仿真技术与工具的飞速发展，数字样机的构建已成为飞机设计制造全生命周期的关键技术。相较于物理样机来说，数字样机基于先进的建模仿真工具和技术来构建产品的数字化模型。由于这类模型能够在一定程度上真实反映产品的物理、行为等特性，因此，通过构建产品的数字样机可以加快设计方案的验证，从而有效地提高产品的设计研发效率，降低成本。数字化技术的发展促使飞机的设计制造逐渐从传统的基于经验设计的模式向数字化智能设计模式的转变^[2-3]。

在飞机设计制造过程中，如何利用建模仿真技术，来分析飞机运行过程中运动学、动力学特性或相关子系统的设计是否满足需求，是一个重要的课题。国内外学者对此开展了较为广泛的研究。文献[4]提出一种基于 Matlab 的 Model Variants 技术可自动升阶的飞控系统的仿真建模方法，该方法只需修改初始化脚本文件便可满足不同设计阶段的需求，从而达到提高设计开发效率的目的。文献[5]针对飞机防滑刹车系统存在的低速打滑和抗干扰能力不足的问题，实现了自抗干扰控制器(auto disturbance rejection controller, ADRC)的设计，数值仿真结果表明，在有无干扰 2 种情况下，ADRC 均优于传统带压力偏调的 PID 控制器。文献[6]基于 Modelica 在 Mworks 平台上建立了飞机液压系统仿真模型，综合全面考虑了飞机液压系统多领域耦合的特性，验证了所设计液压系统可实现相关功能，符合设计要

求。文献[7]提出了一种面向飞机起落架系统的高保真多学科设计优化方法，该设计方法同时考虑起落架系统结构和动态特性，实现了对重量、成本和结构性能 3 个维度的优化。文献[8]基于 AMESim 建立了民机液压供给系统的仿真模型，基于该仿真模型对系统结构和方案设计、设备参数分配、后期测试验证等进行分析来保证液压系统的最佳性能，显著缩短了研发周期，降低了成本。然而单个系统的数字化建模仿真分析只是解决了飞机设计的部分问题，当飞机执行某一任务时，飞机多个子系统之间的交互无法避免。单个子系统的独立仿真分析并不能解决多系统交互设计的缺陷^[9]。因此，必须从系统工程的角度去研究分析才能更好地挖掘设计存在的潜在缺陷，加快满足设计需求的步伐。

系统工程是从整体角度看待问题、思考问题，统筹协调将系统整体向前推进发展的科学方法，系统工程方法对各种复杂系统具有广泛而普适的意义^[10-11]。传统的系统工程方法以自然语言为载体并辅以文字编辑软件工具实现数据在各研发部门之间的交流传递，然而这种基于文档的系统工程方法(document-based system engineering, DBSE)由于易造成传输的数据的分歧性和二义性，已经逐渐被摒弃。当前，主流的系统工程方法是由 INCOSE 在 2007 年提出的基于模型的系统工程(model-based systems engineering, MBSE)^[12]，MBSE 的核心理念是基于统一规范的模型形式传递信息，贯穿整个设计开发过程。当前，实现 MBSE 的主流建模语言是由对象管理组织在 2001 年在统一建模语言 UML 的基础上扩展提出的面向系统工程的标准建模语言 SysML^[13]。SysML 以图形建模的形式基于 9 种图从需求、结构、行为 3 个维度支持系统工程分析。在建模工具方面，目前主流支持 SysML 的有 IBM 开发的 Rhapsody 软件以及达索公司的商业软件 Cameo Systems Modeler (CSM)^[14]。MBSE 相关软件工具是应用 MBSE 必不可少的一个环节，然而，由于 SysML 模型本身不可执行，只能进行系统架构设计逻辑验证，无法实现物理性能的建模仿真，

难以更深层次的反映系统的设计问题。针对上述问题, 相关学者通过 FMI(functional mock-up interface) 实现系统模型和物理模型协同仿真验证设计。文献[15]提出了一种 MBSE 方法实现民机起飞过程仿真, 其基于 FMI 实现了民机起飞过程多子系统的 SysML 功能架构模型和 Simulink 物理模型的联合仿真, 可快速有效的实现需求追溯、确认和早期数字验证。然而, 由于基于接口的联合仿真会影响模型的可信程度甚至难以仿真系统的某些行为特性, 因此, 设计和仿真的集成问题引起了大量学者的关注。

文献[16-17]提出了第一个详细的设计和仿真集成的方法, 即通过扩展统一建模语言(unified modeling language, UML)实现向 Modelica 的映射。文献[18]提出了基于从查询/视图/转换(query view transformation, QVT)的 SysML 到 Modelica 的映射方法, 给出了用于转换元模型的 SysML4Modelica 扩展包定义, 但 QVT 并不支持模型间的双向转换。文献[19]提出了一种基于 SysML 的统一行为建模语言 UBML, 通过联合 Simulink, 基于 TGG(triple graph grammar)实现了复杂机电系统的设计模型到仿真模型的自动映射, 但是 Simulink 的仿真主要面向控制系统, 难以实现系统级的仿真。文献[20]基于元模型的转换思想, 提出了基于 SysML 的可视化模型转换方法, 通过转换元模型的实例化实现系统设计和仿真模型的可视化转换, 但这种转换只是比较粗粒度的转换, 难以支持复杂系统的映射转换需求。文献[21]基于 ATL(ATLAS Transformation Language)实现了 SysML 到 Modelica 的映射, 并给出了用于转换元模型的 M-Design 扩展包定义, 但是并不支持模型间的双向转换。综上可知, 虽然当前设计和仿真集成的方法在一定程度打通了设计和仿真之间的壁垒, 但由于现有系统建模语言和物理建模语言的差异性, 难以真正实现设计模型和仿真模型的相互映射。基于此, 文献[22-23]基于 SysML、Modelica 的建模思想及描述规范, 在 DEVS 的框架下提出了一种面向复杂系统支持 MBSE 的

新一代一体化建模仿真语言—X 语言。X 语言通过两种可互相转换的图形和文本的形式实现对系统的系统架构和物理特性统一描述, 并通过统一的基于 DEVS 的仿真引擎进行仿真验证, 不断反馈验证设计的优劣, 达到提高开发设计效率的目的。

本文提出了一种基于 X 语言面向复杂产品的协同设计一体化建模仿真方法, 并针对民机起飞场景进行了应用验证。首先在确定民机起飞需求的基础上, 在面向 X 语言的一体化建模仿真平台 XLab 上, 基于需求图、用例图以及活动图完成起飞场景的需求及功能分析, 从而完成系统架构设计。基于此, 通过 X 语言的 discrete 类构建飞机起飞过程逻辑架构模型以及运动学和动力学模型、function 类构建飞行环境模型, 最后通过 couple 类构建逻辑架构模型、运动学和动力学模型之间的交互逻辑, 最终基于统一的仿真器实现模型的一体化仿真验证。一体化的建模仿真语言及平台可完整实现系统的需求分析、功能分析、系统设计、仿真验证的全生命周期, 加快验证系统的设计是否满足最初的设计需求, 达到快速优化设计的目的。

1 X 语言及其一体化建模仿真方法

1.1 X 语言概述

X 语言是一种面向复杂系统支持 MBSE 的新一代一体化建模仿真语言^[22-23], X 语言深度融合现有建模语言 SysML、Modelica 的思想及描述规范, 基于离散事件系统规范(discrete event system specification, DEVS), 实现系统架构设计和物理特性的一体化建模, 并基于统一的建模仿真软件—XLab 实现全系统模型的仿真验证。X 语言是一种面向对象的语言, 在 X 语言中定义了 7 种特定类(continuous 类、discrete 类、couple 类等), 每种类都具有图形和文本 2 种建模形式, 二者之间一一对应, 可相互转换。

X 语言图形建模包括 7 种类型的图, 分别是: 需求图、用例图、定义图、连接图、方程图、活动图、状态机图, 其中, 需求图和用例图实现系

统需求模型建立、明确系统功能；定义图、连接图实现模型的结构描述；方程图、活动图、状态机图实现模型的连续、离散以及连续离散混合行为的描述，如图1所示。在X语言建模仿真平台XLab中，系统开发人员通过定义图、连接图来完成对系统模型的架构建模，通过方程图、活动图、状态机图补全各系统功能行为描述后，可以

自动转换成对应的文本模型，启动后台仿真引擎便可进行仿真验证。基于此，可实现系统设计和仿真的一体化。一旦仿真结果无法满足系统需求，可快速定位系统设计问题并对系统存在的设计问题进行优化，不断迭代上述过程直至满足系统需求。因此，基于X语言的一体化建模仿真方法可有效提高系统研发效率，减少开发成本。

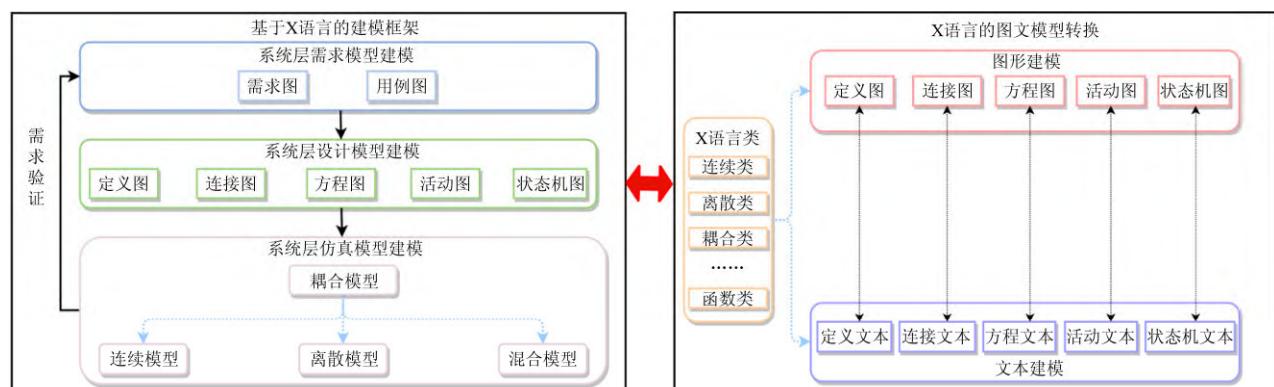


图1 X语言一体化建模仿真框架
Fig. 1 Integrated modeling and simulation framework of X language

1.2 基于X语言面向MBSE的一体化建模仿真方法

当前，MBSE主流核心方法论有2种：面向Rhapsody工具的Harmony SE以及面向Cameo Systems Modeler工具的MagicGrid。

其中，Harmony SE方法论核心分为三部分：需求分析、功能分析以及设计综合。从实现细节方面，Harmony SE从行为描述上主要依赖于活动图、顺序图以及状态机图来完成对复杂的系统对象从黑盒的外部行为识别以及白盒的内部功能逻辑的设计，基本上淡化了参数图的重要性。

MagicGrid方法论通过矩阵的形式实现复杂系统的设计过程。从纵向来看，其从需求、行为、结构以及参数4个核心视角作为系统工程研究的支柱；从横向来看，主要分为问题域和解决域2个层级逐步分析系统的解决方案。其核心的实现流程也是从需求分析出发，逐步定义出系统的黑盒的结构和行为，再到对系统的白盒的结构和行为以及系统的参数约束。从实现细节方面，MagicGrid从行为

描述上依赖于活动图以及状态机图，但是更加重视参数图的建模。

然而，虽然Rhapsody、Cameo Systems Modeler提供了更细粒度描述复杂的系统对象的方法(Harmony SE、MagicGrid)，但归根究底，这种基于SysML构建的模型本身并不可执行，从本质上来说，Rhapsody仅实现对复杂系统的设计过程，无法实现系统设计的闭环。因此，真正的MBSE方法必须引入仿真的手段，只有实现对复杂系统设计和仿真的统一，才能真正实现系统设计的闭环。

目前，传统的实现复杂系统设计和仿真统一的MBSE建模仿真方法有2种：第1种是基于SysML建立系统级需求、功能、逻辑架构模型，基于Matlab/simulink、Modelica等建立物理级模型，最后基于FMI集成建立完整的仿真模型，不断迭代验证需求满足情况，实现对系统设计的快速验证^[15]，如图2所示；第2种是基于SysML建立系统级需求、功能、设计模型，通过扩展SysML并开发SysML与Modelica转换引擎实现系统级设计模型

到仿真模型的转换生成, 从而达到快速验证系统设计的目的。这 2 种方法都需要多种语言、多种平台集成实现。目前, 现有的主流 MBSE 建模工具如 Rhapsody、Cameo Systems Modeler 已经完全支持通过 FMI 实现设计和仿真集成, 然而, 采用 FMI 进行设计和仿真集成过程中, 当对物理模型参数进行优化时, 一旦物理模型需要修改, 则需要将修改后的物理模型重新封装成 FMU, 再集成进行需求验证, 效率低且无法通过优化算法进行自动优化; 通过扩展 SysML 并开发 SysML 与 Modelica 转换引擎的方法, 虽然能实现某些场景模型的自动转换, 然而当模型足够复杂时, 由于 SysML 与 Modelica 元模型的不完全一致性, 导致模型转换的准确性难以保证。而且, 现有的主流 MBSE 建模工具如 Rhapsody、Cameo Systems Modeler 并未提供 SysML 与 Modelica 转换引擎。

X 语言从语言层面便可实现对复杂系统设计和仿真的一体化, 无需通过 FMI 或者开发异构语言之间的转换引擎实现这一过程。这种通过一种语言实现复杂系统设计和仿真的一体化, 加快了对系统设计方案的验证, 通过不断迭代可实现对系统架构设计结构或者参数优化, 从而提高研发效率, 并且有效避免模型数据传递的不一致性和歧义性。

基于 X 语言面向 MBSE 的一体化建模仿真方法, 首先基于 X 语言的需求图、用例图以及活动图构建需求模型、功能模型, 完成概念模型建模分析; 基于 X 语言提供的特定类(离散类、连续类、

函数类等)完成系统级逻辑架构模型和物理级模型的建模, X 语言基于 DEVS 理论, 通过构建耦合类基于事件端口的交互机制实现系统级逻辑架构模型和物理级模型集成, 形成完整的全系统仿真模型。再通过自主开发的一体化仿真引擎仿真验证当前需求的满足度, 不断迭代反馈实现对各子系统参数的优化, 达到对系统设计方案的快速验证与优化, 如图 3 所示。由于 X 语言完全从语言本身支持系统级逻辑架构模型和物理级模型的一体化建模仿真, 相较于 FMI 设计和仿真集成方法, 参数的优化无需进行模型的重新构建再封装集成的过程。当系统级或物理级参数无法满足需求需要优化时, 只需修改该参数便可通过一体化仿真引擎仿真验证。因此, 基于 X 语言面向 MBSE 的一体化建模仿真方法可更快实现系统设计方案验证与优化的进程, 显著提高复杂产品的研发效率。另外, 由于采用基于 X 语言及其仿真引擎的一体化建模仿真, 从理论上来说, 通过智能优化算法实现参数的自动优化也是可行的。

2 基于 X 语言民机起飞场景建模仿真

当前, MBSE 方法已经成为各类复杂系统研发和设计的首选方法。国内外航空、航天领域相关研究机构以及开发厂商正在逐渐采用 MBSE 方法来尝试建立完整的包含系统设计和仿真的虚拟数字样机。

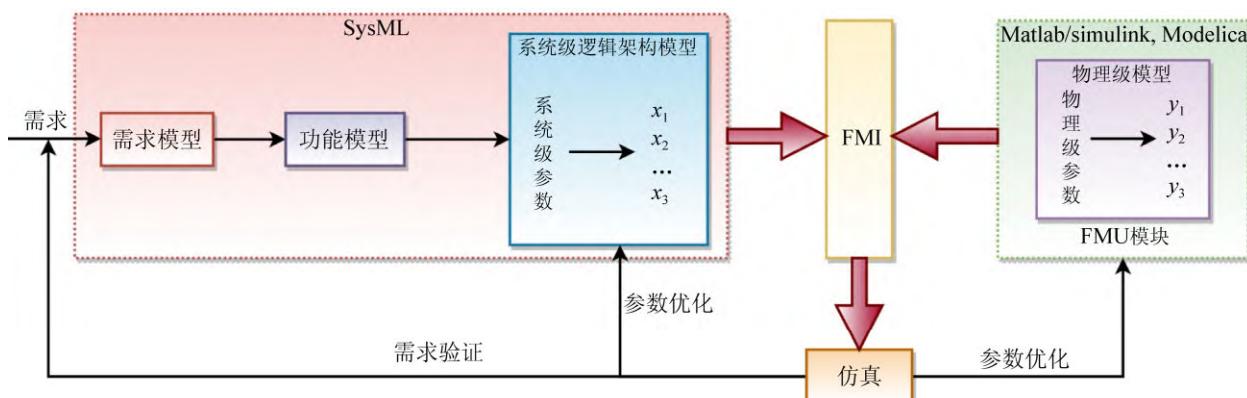


图 2 基于 FMI 集成建模仿真优化方法框架
Fig. 2 Framework of modeling, simulation and optimization methods based on FMI integration

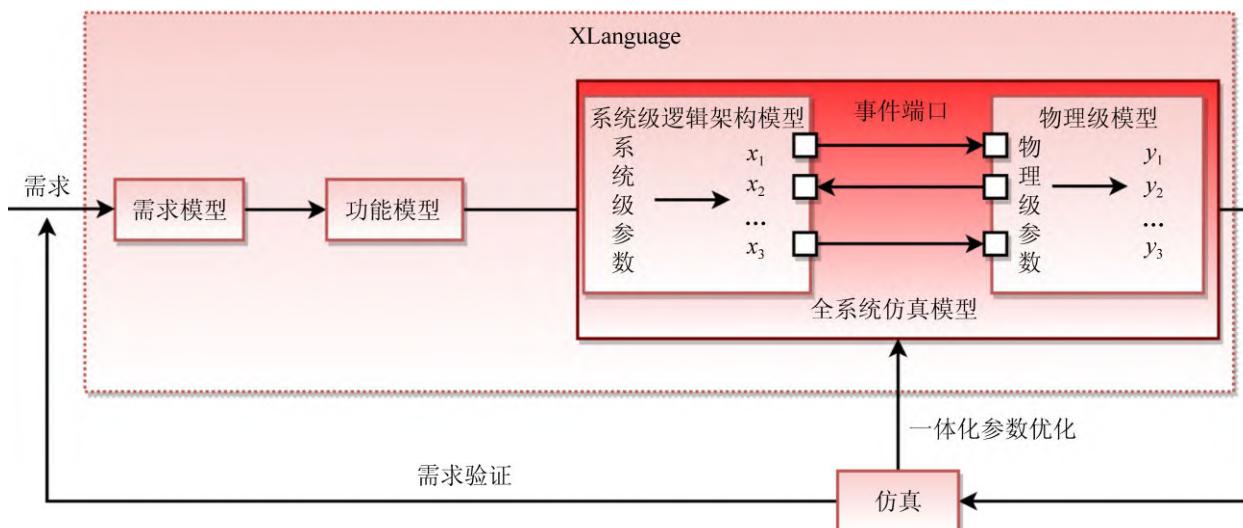


图3 基于X语言建模仿真优化方法框架
Fig. 3 Framework of modeling, simulation and optimization methods based on X language

综上分析，本文采用基于X语言面向MBSE的一体化建模仿真方法实现对民机起飞场景从需求分析、系统设计到仿真的一体化建模仿真，达到对民机起飞系统设计的快速验证与优化的目的。

综合MBSE理论方法和飞机起飞场景的特点，基于X语言面向MBSE的一体化建模仿真方法，分析并设计了基于X语言的民机起飞场景一体化建模仿真设计架构，如图4所示。首先，通过捕获飞机起飞场景的顶层需求，构建相应的需求模型。其次，以需求模型为基础，对飞机顶层系统的功能、架构进行分析，构建可满足需求追溯的飞机起飞的

功能模型和顶层系统架构模型。随后，以所构建的飞机顶层系统架构模型为指导，逐步建立具有离散逻辑行为和连续动态行为的子系统模型。基于X语言的端口集成生成系统架构可执行模型，从而实现飞机起飞场景系统设计和仿真验证的一体化。这种一体化的建模仿真方法可以快速的通过仿真验证系统设计的正确性，不断迭代验证进而达到快速验证与优化的目的。

基于上述架构，本文将进行整个飞机起飞场景基于X语言的需求分析、系统功能架构分析以及子系统模型的构建。

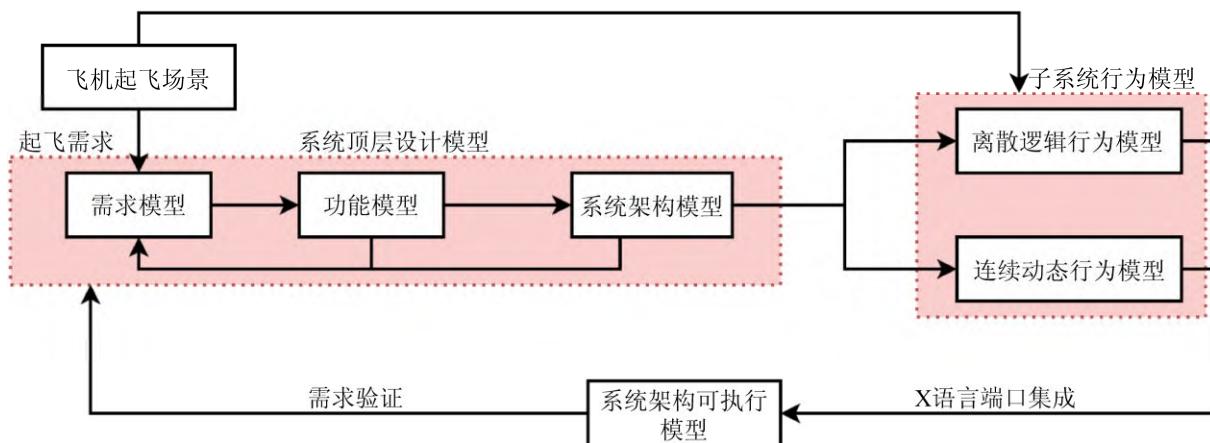


图4 基于X语言的民机起飞场景一体化建模仿真优化框架
Fig. 4 Framework of modeling, simulation and optimization for civil aircraft take-off scenarios based on X language

2.1 民机起飞系统需求分析

本文以典型民机波音 747-400 为研究对象, 构建其整个起飞场景仿真模型, 针对波音 747-400 起飞场景分析飞行需求的核心利益相关方—飞行员的需求。通过查询文献[24], 结合 X 语言需求图对民机起飞场景的需求进行了建模。如图 5 所示, 该需求图描述了民机起飞过程中利益相关方—飞行员的核心系统级需求及其包含的 8 个功能组件级需求。X 语言需求图中的需求模型包含了 “Id”、

“Type”、“Level”和“Text” 4 种属性, 分别表示对需求的编号、类型、层级以及具体的描述。比如, 需求 req1 是一个编号 kr.2 的功能型组件级需求, 该需求要求飞机可以正常接受起飞指令。

为了细化起飞场景的需求, 本文基于 X 语言用例图来理清系统所能实现的功能。如图 6 所示, 该用例图描述了起飞场景的核心用例—起飞用例及其包含的 4 个子用例(滑跑用例、抬轮用例、爬升用例以及通讯用例)。

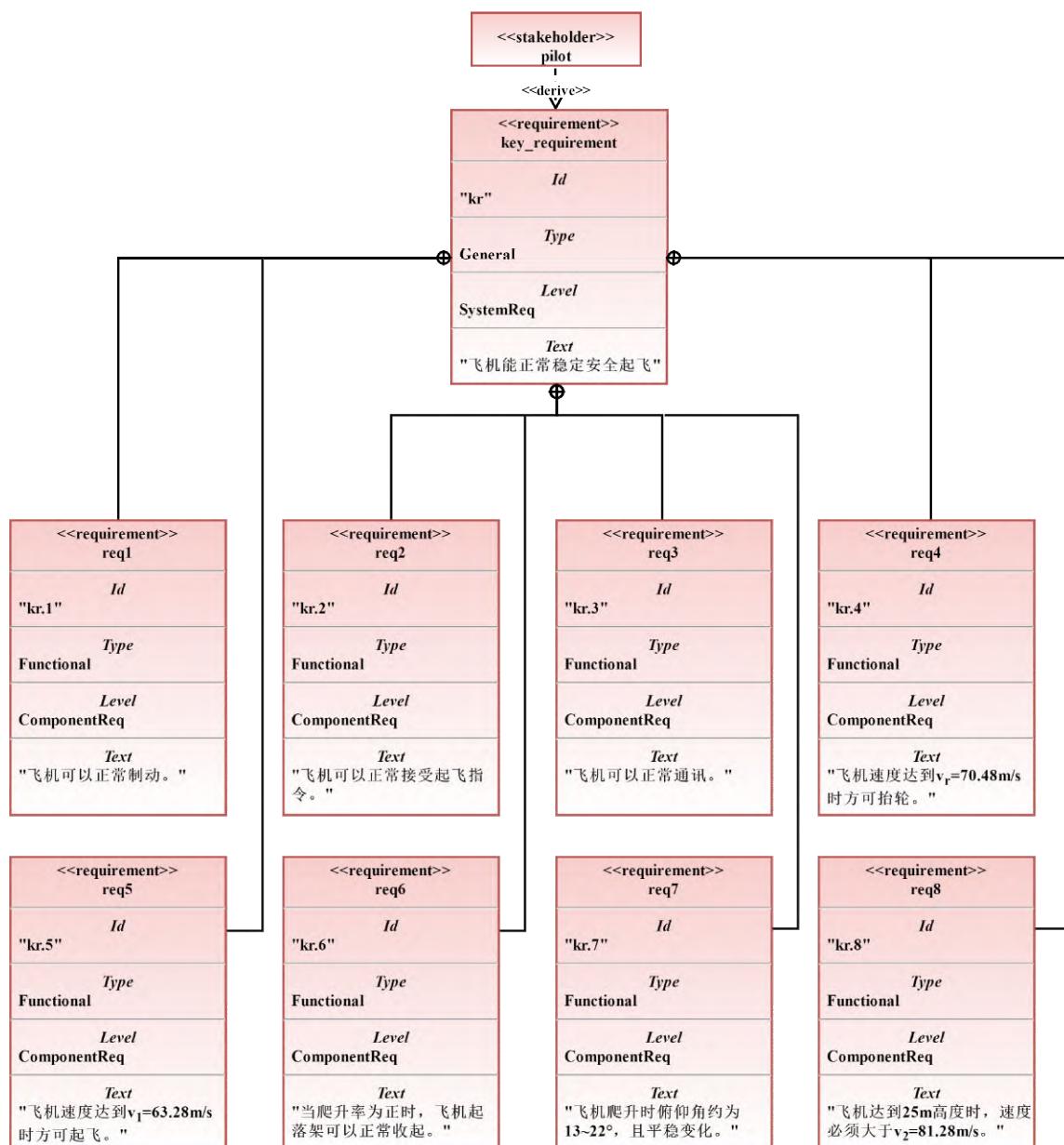


图 5 基于 X 语言的民机起飞场景需求图
Fig. 5 Requirement diagram for civil aircraft take-off scenarios based on X language

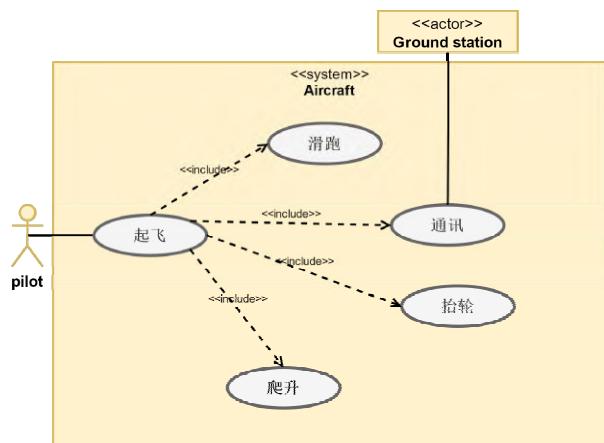


图 6 基于 X 语言的民机起飞场景用例图
Fig. 6 Use case diagram for civil aircraft take-off scenarios based on X language

2.2 系统功能架构分析

当构建完起飞场景需求模型和用例模型之

后，可通过 X 语言的活动图建立飞机起飞功能活动图，如图 7 所示。飞行开始之前，飞行员进行检查并与地面基站进行通讯，确认无误后，在收到地面基站发出的起飞信号后释放制动手，随后将发动机的增压比调整为 1.1。在发动机稳定后逐渐提高飞机推力。在飞机速度超过抬轮速度时，飞行员调整升降舵以抬起飞机前轮。此时，飞机加速并逐渐获得正爬升率。在飞机离地 12 m 以上的高度时，收起起落架完成起飞过程。通过泳道形式将飞机起飞场景的功能活动流分配给相应参与者，可清晰的分析出整个起飞场景参与的系统以及各系统之间的交互情况，这样便可构建整个飞机起飞系统的顶层模型。

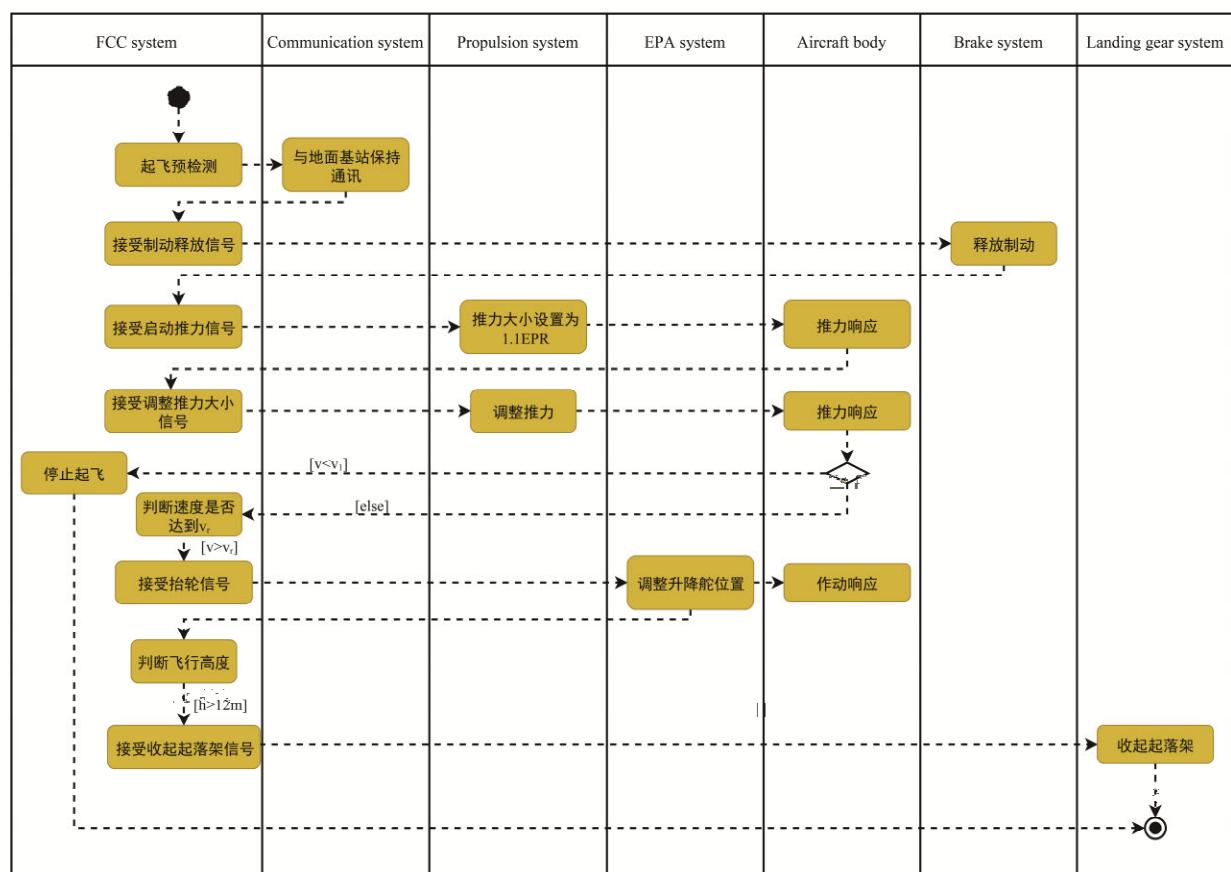


图 7 基于 X 语言的民机起飞场景活动图
Fig. 7 Activity diagram for civil aircraft take-off scenarios based on X language

2.3 飞行系统建模

基于上述功能架构分析，整个波音 747-400 起飞过程会包括多个飞机子系统，比如 FCC 系统、推力系统、制动系统等。在 X 语言中，遵循自顶向下的建模理念，基于耦合类来建立整个飞行系统内部各子系统的组成及其之间的交互关系。X 语言耦合类在图形建模层面是由定义图和连接图

进行描述, 如图 8~9 所示。其中, 定义图中描述了整个飞行系统在起飞过程中所涉及的子系统; 连接图描述了各子系统之间的信号或数据的交互关系。连接图中标黄的子系统具有离散特性, 标红的子系统具有连续特性。另外, 在建立完飞行系统的图形模型之后可自动生成文本模型, 如图 10 所示。

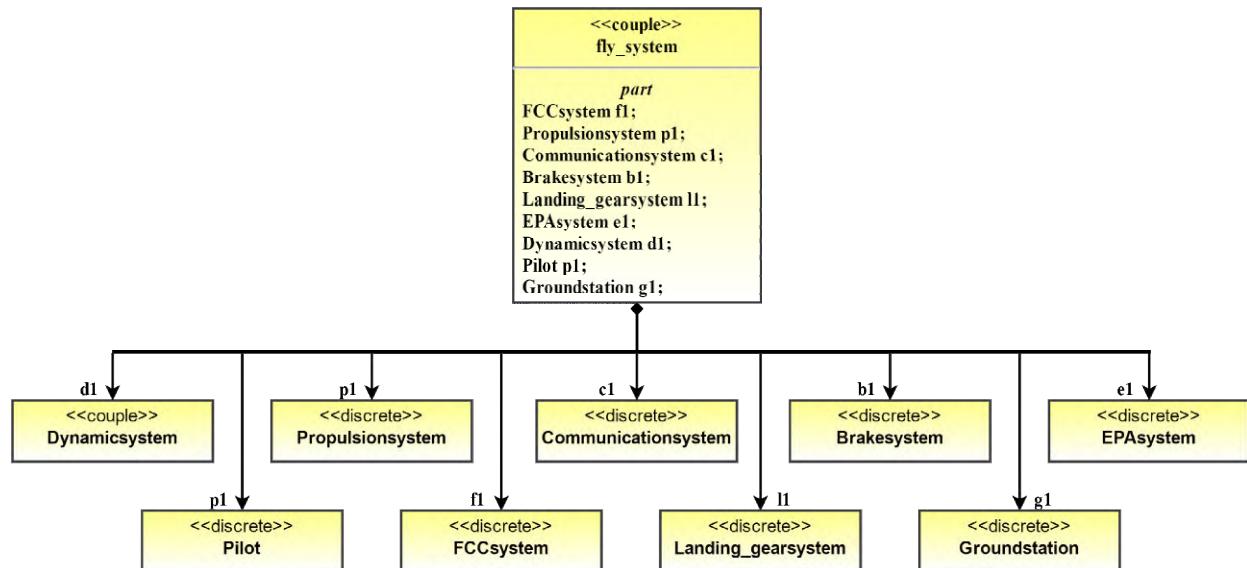


图 8 基于 X 语言飞行系统定义图
Fig. 8 Definition diagram of flight system based on X language

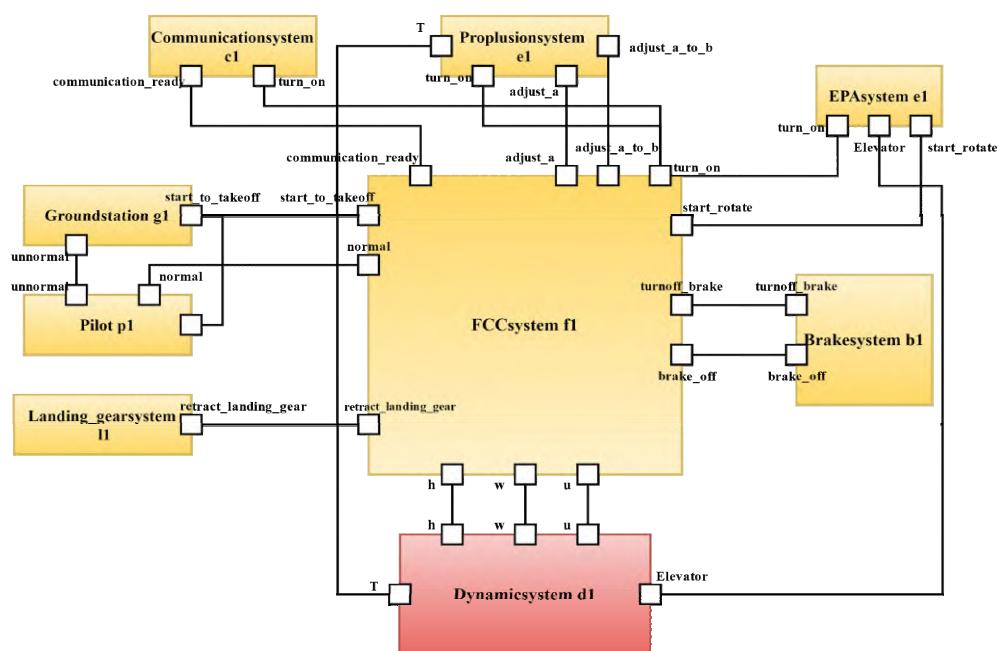


图 9 基于 X 语言飞行系统连接图
Fig. 9 Connection diagram of flight system based on X language

```
couple fly_system
import FCCsystem;
import Propulsionsystem;
import Communicationsystem;
import Brakesystem;
import Landing_gearsystem;
import EPAsystem;
import Dynamicsystem;
import Pilot;
import Groundstation;
part:
FCCsystem f1;
Propulsionsystem p1;
Communicationsystem c1;
Brakesystem b1;
Landing_gearsystem l1;
EPAsystem e1;
Dynamicsystem d1;
Pilot p1;
Groundstation g1;
connection:
connect(f1.turn_on,e1.turn_on);
connect(f1.turn_on,p1.turn_on);
...
end;
```

图 10 基于 X 语言飞行系统部分文本模型
Fig. 10 Partial text model of flight system based on X language

2.4 飞行子系统建模

基于所建立的飞行系统模型,可以定义飞行过程中各子系统的信号或数据交互逻辑。在上述分析的基础上,基于 X 语言建立各子系统的结构和行为逻辑。一旦所有子系统构建完毕,便可进行仿真。本文就子系统中典型的具有离散特性的 FCC 系统和具有连续特性的动力学耦合系统展开介绍。

2.4.1 FCC 系统

飞行控制计算机系统在整个过程中,作为主控系统,协调控制着整个起飞过程其他子系统的逻辑行为。飞控系统在接受起飞指令后,会进入预检测状态,确认无误后,开始准备起飞,在接受到通讯系统正常指令后,开始给制动系统发送指令释放制动器,在接受制动释放指令后,发送推力指令 1 给推力系统,3 s 后发送推力指令 2 给推力系统,当速度大于抬轮速度 v_r 时,发送升降舵调整指令给电力系统,当飞行高度大于 12 m 时,发送收起起落架指令给起落架系统完成起飞过程。

这里,基于 X 语言的离散类来实现飞控系统的结构和行为描述。如图 11~12 所示,其中,定义图中描述了飞控系统的输入输出信号和相关状态变量;状态机图描述了飞控系统的状态和行为逻

辑。另外,在建立完飞控系统的图形模型之后可自动生成文本模型,如图 13 所示。

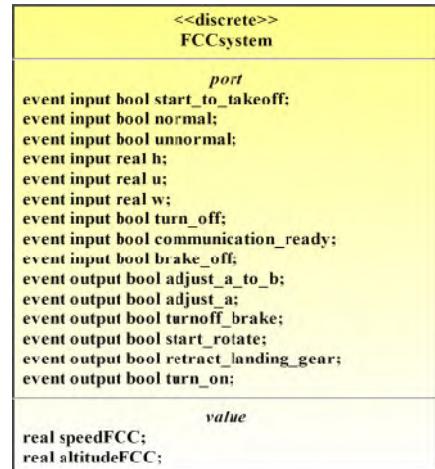


图 11 基于 X 语言 FCC 系统定义图
Fig. 11 Definition diagram of FCC system based on X language

2.4.2 动力学耦合系统

飞机起飞场景的动力学耦合系统主要包括飞行环境模型、动力学模型及运动学模型。

飞机起飞过程包括滑跑、抬轮、爬升 3 个阶段,整个过程主要受到空气作用于飞机的升力 L 和阻力 D 、地面的支持力 F_N 、摩擦力 F_f 、重力 G 和推力 T 。

飞机在滑跑阶段,与地面接触,此时存在地面的支持力 F_N 和摩擦力 F_f 。当飞机滑轮离开地面时,支持力 F_N 和摩擦力 F_f 会瞬间消失。通过对飞机起飞过程受力分析,在机体坐标系下,支持力 F_N 、 X 轴方向和 Z 轴方向的合力如式(1)所示。

$$\left. \begin{aligned} F_N &= -[G + D \sin(\theta - \alpha) - T \sin(\varphi_T + \theta) - \\ &\quad L \cos(\theta - \alpha)] \\ F_x &= T \cos \varphi_T - F_N \sin \theta + F_f \cos \cos \theta - \\ &\quad G \sin \theta - L \sin \alpha + D \cos \alpha \\ F_z &= -T \sin \varphi_T + F_N \cos \theta + F_f \sin \sin \theta + \\ &\quad G \cos \theta + L \cos \alpha + D \sin \alpha \end{aligned} \right\} \quad (1)$$

式中: φ_T 为发动机安装角; θ 为飞机俯仰角; α 为飞机迎角。

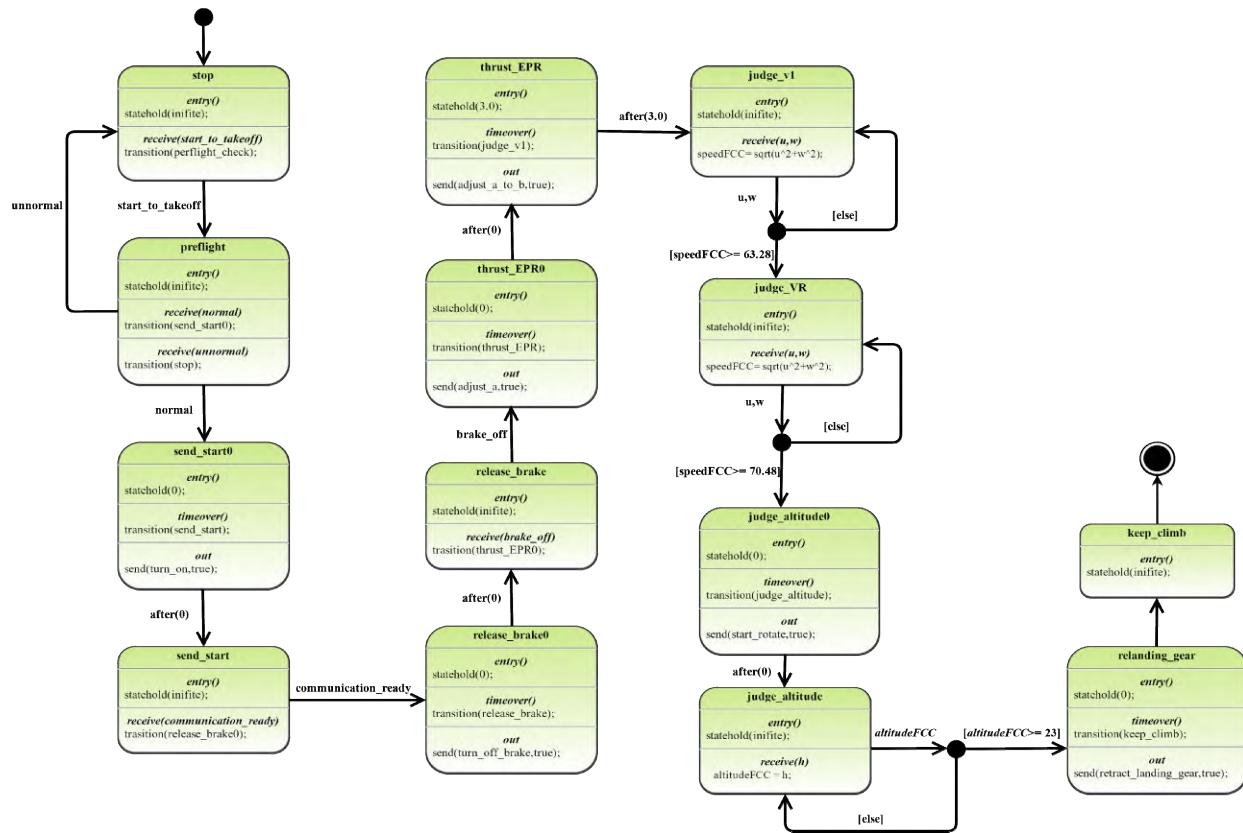


图 12 基于 X 语言 FCC 系统状态机图
Fig. 12 State machine diagram of FCC system based on X language

```

discrete FCCsystem
value:
real speedFCC;
real altitudeFCC;
port:
event input bool start_to_takeoff;
event input bool normal;
event input bool unnormal;
event input real h;
...
state:
initial state stop
when entry() then
statehold(infinite);
end;
when receive(start_to_takeoff) then
transition(preflight_check);
end;
end;

```

图 13 基于 X 语言 FCC 系统部分文本模型
Fig. 13 Partial text model of FCC system based on X language

由于飞机起飞场景纵向方向的运动变化是其核心, 因此本文仅对飞机的纵向运动规律进行分析, 如式(2)所示, 飞机起飞过程的运动学方程为

$$\begin{cases}
\frac{F_X}{m} = \dot{u} + wq \\
\frac{F_Z}{m} = \dot{w} + uq \\
M = \dot{q}I_Y \\
\dot{\theta} = q \\
x = u \cos \theta + w \sin \theta \\
h = -u \sin \theta + w \cos \theta
\end{cases} \quad (2)$$

式中: u , w 分别为 X 轴和 Z 轴方向速度; \dot{u} , \dot{w} 分别为 X 轴和 Z 轴方向加速度; m 为飞机的质量; θ 和 q 分别为俯仰角和俯仰角速率; M 和 I_Y 分别为俯仰力矩和 Y 轴惯量; x 和 h 分别为 X 轴和 Z 轴方向飞行距离。

基于上述分析, 在 X 语言中, 遵循自顶向下的建模理念, 基于 couple 类建立动力学耦合系统顶层模型(各子系统以及之间的交互逻辑)。随后, 基于 discrete 类分别建立动力学系统模型和运动学系统模型, 基于 function 类建立飞行环境模型。下面, 就动力学耦合系统顶层模型以及运动学系统模型

的构建展开介绍。

基于 couple 类构建力学耦合系统顶层模型。在图形建模层面,主要由定义图和连接图进行描述。其中,定义图描述了力学耦合系统的各子系统和输入输出端口,连接图描述了各子系统之间的交互逻辑,如图 14~15 所示。另外,在建立完力学耦合系统图形模型之后可自动生成相对应的文本模型,如图 16 所示。

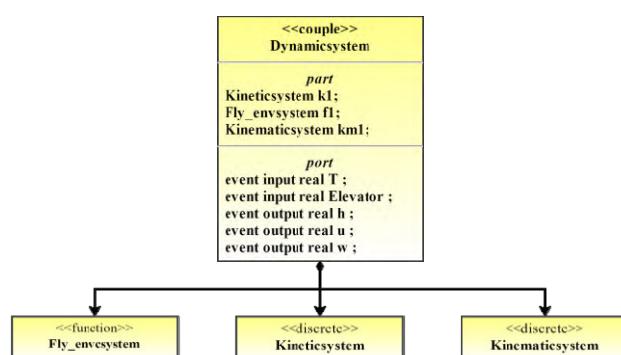


图 14 基于 X 语言力学耦合系统定义图
Fig. 14 Definition diagram of dynamic coupling system based on X language

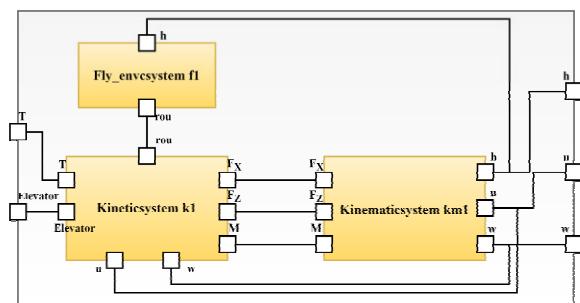


图 15 基于 X 语言力学耦合系统连接图
Fig. 15 Connection diagram of dynamic coupling system based on X language

```
couple Dynamicsystem
import Kineticsystem; import Fly_envsystem; import Kinematicsystem;
part: Kineticsystem k1; Fly_envsystem f1; Kinematicsystem km1;
port: event input real T; event input real Elevator; event output real h; event output real u; event output real w;
```

图 16 力学耦合系统部分文本模型
Fig. 16 Partial text model of dynamic coupling system based on X language

基于 discrete 类构建运动学系统模型。在图形建模层面,主要由定义图和状态机图进行描述,如图 17~18 所示。其中,定义图描述了运动学系统的参数、状态变量以及输入输出的端口,状态机图描述了运动学方程的求解过程。另外,在建立完运动学耦合系统图形模型之后可自动生成相对应的文本模型,如图 19 所示。

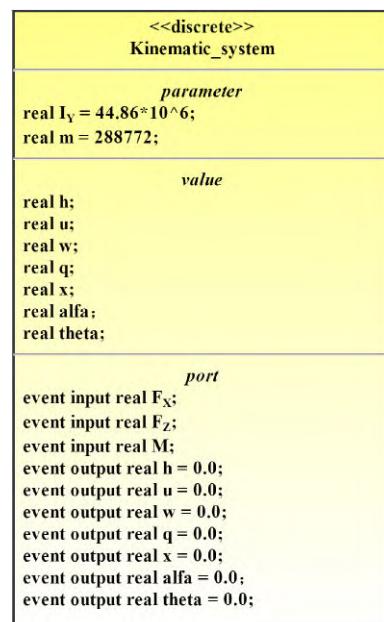


图 17 基于 X 语言运动学系统定义图
Fig. 17 Definition diagram of kinematic system based on X language

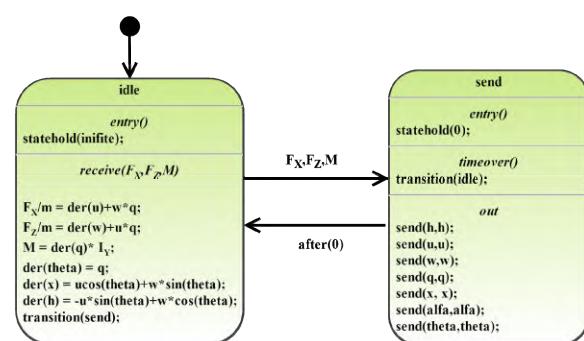


图 18 基于 X 语言运动学系统状态机图
Fig. 18 State machine diagram of kinematic system based on X language

3 基于 X 语言飞行系统仿真分析

在完成所有民机起飞场景相关子系统模型的构建后,顶层的飞行系统架构模型通过各起飞子系

统的端口连接关系生成完整的飞行系统可执行的仿真模型进行仿真,通过得到的仿真结果来验证当前所设计的起飞系统是否满足利益相关者提出的需求,一旦需求无法得到满足,可以通过仿真结果辅助分析当前系统亟需优化的问题,经过多次迭代优化,最终达到所设计的起飞系统能完全满足利益相关者需求。这里,主要通过仿真结果验证利益相关者所提出的飞行性能需求满足情况。由图5构建的飞机起飞场景的需求模型可知,飞机起飞过程的性能需求主要包括: req4、req5、req7 和 req8。

```
discrete Kinematicsystem
parameter:
real Iy =44.86*106;
real m=288772;
value:
real h;
real u;
...
port:
event input real FX;
event input real FZ;
...
state:
initial state idle
when entry() then
statehold(infinite);
end;
when receive(FX,FZ,M) then
u = u + time_step* der_u;
w = w + time_step*der_w;
...
transition(send);
end;
end;
```

图19 基于X语言运动学系统部分文本模型
Fig. 19 Partial text model of kinematic system based on X language

图20~23展示了飞机在起飞的过程中各种性能参数(飞行高度、水平飞行距离、速度以及俯仰角等)的变化情况。结合图20~22可以发现,飞机速度在接近80 m/s时进行抬轮动作;当飞机离地面25 m高度时,飞行速度接近85 m/s。由此可见,当前的飞行系统设计满足性能需求req4、req5、req8。另外,由图23可知,飞机在起飞过程中,飞机的俯仰角和迎角在起飞早期保持一致,大约在增大到6°左右逐渐分离,后续俯仰角持续增大最终保持在15°左右,迎角逐渐减小至0,这说明在起飞过程中,当俯仰角增大到6°左右,飞机正在离开地面并最后以15°左右的俯仰角爬升。因此,性能

需求req7也得以满足。

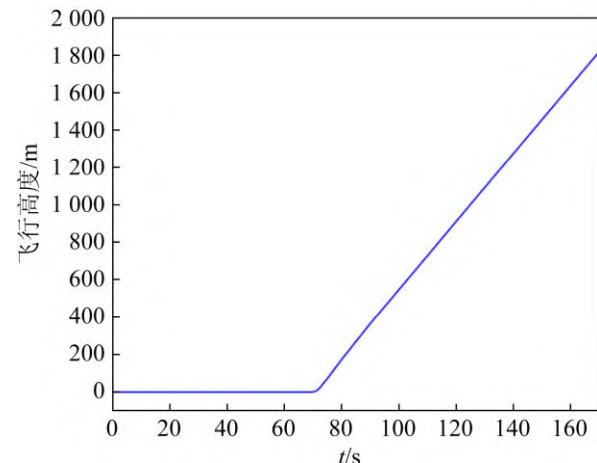


图20 飞行高度仿真结果
Fig. 20 Simulation result of flight height

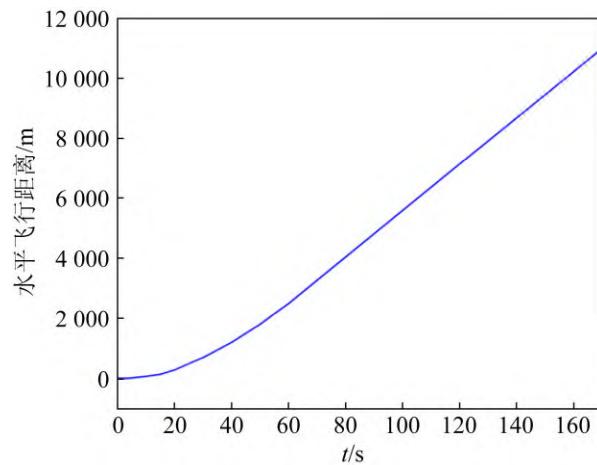


图21 水平飞行距离仿真结果
Fig. 21 Simulation result of horizontal distance

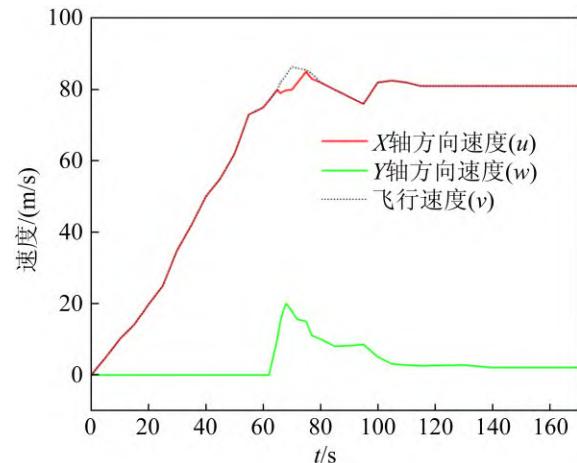


图22 速度仿真结果
Fig. 22 Simulation result of velocity

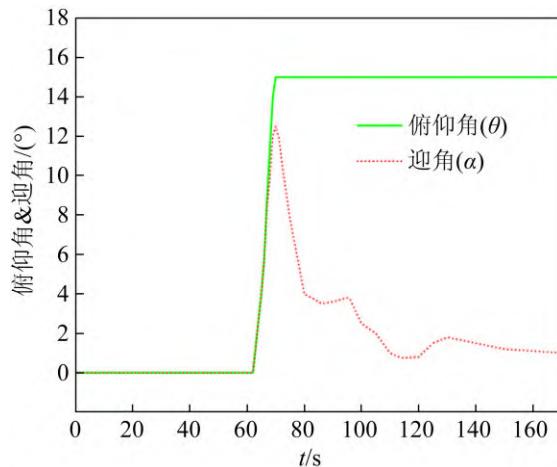


图23 俯仰角&迎角仿真结果
Fig. 23 Simulation result of angle of pitch & angle of attack

综上所述,基于X语言面向MBSE的一体化建模仿真方法通过一体化的建模仿真语言和建模仿真软件,可快速实现对民机起飞系统需求、功能分析到系统架构设计和仿真的全闭环验证过程,可有效提高系统开发效率。

4 结论

本文提出了基于X语言面向MBSE的一体化建模仿真方法,基于该方法并结合X语言及其建模仿真软件实现了对飞机起飞场景从需求、功能分析到系统架构设计和仿真的一体化验证,解决了当前主流的系统设计和仿真集成(FMI和异构语言转换引擎)两种方法的不足。比如,上述2种方法均需要多语言、多平台的实现;基于FMI集成需要研究目标软件的FMI封装技术且每次进行参数的优化都需要重新构建模型,二次封装成FMU模块再集成进行仿真优化,过程繁琐且无法通过优化算法进行自动的优化;基于异构语言转换引擎由于不同语言的差异性,难以保证模型的一致性等问题。

基于X语言面向MBSE的一体化建模仿真方法可建立民机起飞从需求、功能分析到系统架构设计和仿真的一体化全闭环建模仿真系统,可有效地对复杂产品进行早期验证,达到提高效率的目的。

相较于基于FMI的设计和仿真集成方法,由于X语言完全从语言本身支持离散逻辑模型和连续动态模型的一体化建模,参数的优化过程无需进行模型的重新构建再封装集成的过程,因此,可以通过智能优化算法,基于X语言面向MBSE协同设计进行参数的自动优化。

本文仅研究了民机起飞过程的建模仿真优化方法,下一步可考虑研究飞机的巡航、降落等过程,通过相关遗传优化算法,实现飞行过程相关参数的自动优化。

参考文献:

- [1] 张霖,任磊,陶飞.复杂产品制造数字化集成平台技术[J].国防制造技术,2010(4): 4-10.
Zhang Lin, Ren Lei, Tao Fei. Complex Product Manufacturing Digital Integration Platform Technology[J]. Defense Manufacturing Technology, 2010(4): 4-10.
- [2] 郑党党,张志国,刘俊堂.飞机设计中的数字样机技术[J].航空制造技术,2016(5): 83-86.
Zheng Dangdang, Zhang Zhiguo, Liu Juntang. DMU Technology of Aircraft Design[J]. Aeronautical Manufacturing Technology, 2016(5): 83-86.
- [3] 陈裕.民用飞机数字样机建模与仿真技术初步研究[J].民用飞机设计与研究,2018(3): 54-57.
Chen Yu. Preliminary Research on Digital Mockup Modeling and Simulation Technology for Civil Aircraft [J]. Civil Aircraft Design & Research, 2018(3): 54-57.
- [4] 马松辉,黄立梅,李正强,等.一种可自动升阶的飞控系统仿真建模方法[J].计算机仿真,2019, 36(9): 57-65.
Ma Songhui, Huang Limei, Li Zhengqiang, et al. A Simulation and Modeling Method of Flight Control System That can be Automatically Upgraded[J]. Computer Simulation, 2019, 36(9): 57-65.
- [5] 刘文胜,许丰瑞,马运柱,等.飞机防滑刹车系统控制器优化设计[J].计算机仿真,2016, 33(2): 93-98.
Liu Wensheng, Xu Fengrui, Ma Yunzhu, et al. Optimization Design of Aircraft Anti-skid Braking System Controller[J]. Computer Simulation, 2016, 33(2): 93-98.
- [6] 张宝坤,赵建军,刘伟.关于飞机液压负载功率系统优化设计研究[J].计算机仿真,2017, 34(5): 82-87.
Zhang Baokun, Zhao Jianjun, Liu Wei. Research on the

- Optimization Design of Aircraft Hydraulic Load Power System [J]. Computer Simulation, 2017, 34(5): 82-87.
- [7] Wong J, Ryan L, Kim I Y. Design Optimization of Aircraft Landing Gear Assembly Under Dynamic Loading[J]. Structural & Multidisciplinary Optimization (S1615-147X), 2018, 57(3): 1357-1375.
- [8] Tao L. Design Analysis and Optimization for the Civil Aircraft Hydraulic Supply System Based on AMESim[C]// IEEE International Conference on Aircraft Utility Systems. Beijing: IEEE, 2016: 890-894.
- [9] Greg G. Product Lifecycle Management: Driving the Next Generation of Lean Thinking by Michael Grieves[J]. Journal of Product Innovation Management (S0737-6782), 2007, 24(3): 278-280.
- [10] Russell M. Using MBSE to Enhance System Design Decision Making[J]. Procedia Computer Science (S1877-0509), 2012, 8(1): 188-193.
- [11] 钱学森, 许国志, 王寿云. 组织管理的技术——系统工程[J]. 上海理工大学学报, 2011, 33(6): 520-525.
Qian Xuesen, Xu Guozhi, Wang Shouyun. Technology of Organizational Management—System Engineering[J]. Journal of Shanghai University of Science and Technology, 2011, 33(6): 520-525.
- [12] INCOSE. Systems Engineering Vision 2020[R]. Seattle: INCOSE, 2007.
- [13] Engineer M. OMG Systems Modeling Language (OMG SysML™) Tutorial[C]// INCOSE International Symposium. Singapore: INCOSE, 2009: 1840-1972.
- [14] Casse O. SysML in Action with Cameo Systems Modeler[M]. Great Britain: ISTE Press Ltd and USA: Elsevier Ltd, 2017.
- [15] 赵良玉, 叶俊杰, 何琪, 等. 基于MBSE的民用飞机起飞场景仿真[J]. 系统仿真学报, 2021, 33(10): 2499-2510.
Zhao Liangyu, Ye Junjie, He Qi, et al. Simulation of Civil Aircraft Takeoff Scenario Based on MBSE[J]. Journal of System Simulation, 2021, 33(10): 2499-2510.
- [16] Schamai W. Modelica Modeling Language (ModelicaML): A UML Profile for Modelica[R]. Sweden: Linköping University, Department of Computer and Information Science, The Institute of Technology, 2009.
- [17] Schamai W, Pohlmann U, Fritzson P, et al. Execution of Umlstate Machines Using Modelica[C]// The 3rd International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools. Oslo, Norway: EOOLT, 2010: 1-10.
- [18] Object Management Group. SysML-Modelica Transformation Version 1.0[EB/OL]. [2012-11-09]. <https://www.omg.org/spec/SyM/>.
- [19] Cao Y, Liu Y, Fa N H, et al. SysML-Based Uniform Behavior Modeling and Automated Mapping of Design and Simulation Model for Complex Mechatronics[J]. Computer-Aided Design (S0010-4485), 2013, 45(3): 764-776.
- [20] 李新光, 刘继红. 基于SysML的系统设计-仿真模型可视化转换[J]. 计算机辅助设计与图形学学报, 2016, 28(11): 1973-1981.
Li Xinguang, Liu Jihong. A Method of SysML-Based Visual Transformation of System Design-Simulation Models[J]. Journal of Computer-Aided Design & Computer Graphics, 2016, 28(11): 1973-1981.
- [21] 周书华, 曹悦, 张政, 等. 基于SysML和Modelica的复杂机电产品系统设计与仿真集成[J]. 计算机辅助设计与图形学学报, 2018, 30(4): 728-738.
Zhou Shuhua, Cao Yue, Zhang Zheng, et al. System Design and Simulation Integration of Complex Electromechanical Products Based on SysML and Modelica[J]. Journal of Computer Aided Design and Graphics, 2018, 30(4): 728-738.
- [22] Zhang L, Ye F, Laili Y, et al. X Language: An Integrated Intelligent Modeling and Simulation Language for Complex Products[C]// 2021 Annual Modeling and Simulation Conference (ANNSIM). Fairfax, VA, USA: IEEE, 2021: 1-11.
- [23] Zhang L, Ye F, Xie K, et al. An Integrated Intelligent Modeling and Simulation Language for Model-based Systems Engineering[J]. Journal of Industrial Information Integration, 2022(28): 100347.
- [24] The Boeing Company. Boeing 747 Flight Crew Operations Manual[EB/OL]. [2006-10-01]. <https://www.soupan8.com/file/15866920/>.