Journal of System Simulation

Volume 30 | Issue 9 Article 18

1-8-2019

Design of Simulation Platform for Space Operation Mission

Yuan Jing
1. National Key Laboratory of Aerospace Flight Dynamic, College of Astronautics, Northweatern Polytechnical Univesity, Xi'an 710072, China;;

Jianpin Yuan
1. National Key Laboratory of Aerospace Flight Dynamic, College of Astronautics, Northweatern Polytechnical Univesity, Xi'an 710072, China;;

Mingming Wang
1. National Key Laboratory of Aerospace Flight Dynamic, College of Astronautics, Northweatern Polytechnical Univesity, Xi'an 710072, China;;

Wang Fei
1. National Key Laboratory of Aerospace Flight Dynamic, College of Astronautics, Northweatern Polytechnical Univesity, Xi'an 710072, China;;

See next page for additional authors

Design of Simulation Platform for Space Operation Mission

Abstract

Abstract: The design of simulation platform for space mission operations is investigated. The platform has the advantages of high generality and extensibility, being easy to build up new task. FMI (Functional Mockup Interface) standard is adopted to achieve *integration of multisource models*, and Python scripts are used to realize the task schedule. Unity3D is adopted to implement the visual display system which could display space manipulation process with high fidelity 3D virtual scenes. *Configuration tool is developed to map the 3D objects in visual scene with simulation physical variables for complex space control mechanism*, which greatly improves the platform's generality and extensibility. Finally a typical space operation mission of capturing an on-orbit target by deployed robotic arm is taken to demonstrate the functions of the platform.

Keywords

space operation, simulation system, visual display, FMI standard

Authors

Yuan Jing, Jianpin Yuan, Mingming Wang, Wang Fei, and Zhang Chi

Recommended Citation

Yuan Jing, Yuan Jianpin, Wang Mingming, Wang Fei, Zhang Chi. Design of Simulation Platform for Space Operation Mission[J]. Journal of System Simulation, 2018, 30(9): 3366-3376.

第 30 卷第 9 期 系统仿真学报© Vol. 30 No. 9
2018 年 9 月 Journal of System Simulation Sep., 2018

空间操控任务控制仿真平台设计

袁静[1]，袁建平[1]，王明明[1]，王菲[1]，张弛[2]
(1. 西北工业大学航天学院 航天飞行动力学技术国家级重点实验室，西安 710072；2. 北京世冠科技有限公司，北京 100027)

摘要：提出一种空间操控任务控制仿真平台设计方案，具有通用性强、任务组建速度快、操控过程模拟逼真度高等特点。平台采用 FMI 实现*多源模型耦合*，通过 Python 脚本进行任务控制，基于 Unity3D 实现虚拟视景仿真，通过*配置工具建立复杂空间操控机构的虚拟场景模型与数字仿真模型变量的映射*，提高了平台的通用性和可扩展性。以典型在轨维护机械臂抓捕过程为例，进行了仿真验证。

关键词：空间操控；仿真系统；视景显示；通用模型接口标准 FMI
中图分类号：TP391.3 文献标识码：A 文章编号：1004-731X (2018) 09-3366-11
DOI: 10.16182/j.issn1004731x.joss.201809018

Design of Simulation Platform for Space Operation Mission

Yuan Jing[1], *Yuan Jianpin*[1], *Wang Mingming*[1], *Wang Fei*[1], *Zhang Chi*[2]
(1. National Key Laboratory of Aerospace Flight Dynamic, College of Astronautics, Northwestern Polytechnical Univesity, Xi'an 710072, China;
2. Global Crown Technology Co. Limited, Beijing 100027, China)

Abstract: The design of simulation platform for space mission operations is investigated. The platform has the advantages of high generality and extensibility, being easy to build up new task. FMI（Functional Mockup Interface) standard is adopted to achieve *integration of multisource models*, and Python scripts are used to realize the task schedule. Unity3D is adopted to implement the visual display system which could display space manipulation process with high fidelity 3D virtual scenes. *Configuration tool is developed to map the 3D objects in visual scene with simulation physical variables for complex space control mechanism*, which greatly improves the platform's generality and extensibility. Finally a typical space operation mission of capturing an on-orbit target by deployed robotic arm is taken to demonstrate the functions of the platform.

Keywords: space operation; simulation system; visual display; FMI standard

引言

随着对空间研究、开发与应用能力的不断提高，各国相继研制并发射了大量面向各种任务要求的航天器，航天器的结构、组成日趋复杂，性能、技术水平不断提高。为了提升航天器的性能、延长航天器的使用寿命、降低费用和风险，对于以空间交会对接、碎片规避、轨道与姿态重置、在轨释放有效载荷、在轨维修等为内容的空间操作技术成为未来空间活动的必然[1]。因而大范围快速轨道/姿态机动和控制技术，以及精确的环境力系建模已经成为了空间操作任务最基本也是最关键的技术。空间操控任务过程仿真是检验任务设计指标、技术可行性的重要手段，涉及多学科耦合、多语言建模等复杂系统仿真。通过空间操控平行试验技术，将空间操控试验进行虚拟化，进

收稿日期：2016-11-07 修回日期：2017-02-03;
基金项目：国家“863”计划(2015AA2247);
作者简介：袁静(1967-)，女，四川，硕士，高工，研究方向为航天飞行动力学建模与仿真；袁建平(1957-)，男，陕西，博士，教授，博导，研究方向为航天飞行动力学。

而在仿真模拟环境下进行虚拟化的推演、评估、控制与决策等活动，是空间操控任务概念设计、过程控制、态势判定、指挥决策和试验评估的重要环节。

在航天器任务仿真设计方面，美国 AGI 公司开发的 STK(Satellite Tool Kit)、NASA 与商业公司合作开发的开源工具 GMAT(General Mission Analysis Tool)在业界几乎无人能及，但目前为止，这两款工具尚不支持复杂空间操控过程的分析和实时仿真。在空间操控任务仿真方面，国内同行已经进行了不少工作，首先是卫星研制和服务部门为其具体的型号任务研制了不同的数字仿真系统或半物理仿真系统，此外，安昊等[2]采用 Matlab 算法库及其图形界面开发了交会对接全数字仿真系统，王明明等[3]设计了基于 DDS 的空间机器人遥操作实时仿真系统，周剑勇等[4]基于 TCP/IP 协议构建了飞行器机动飞行分布式仿真系统。这些系统都针对特定的任务目标采用了不同的软件和架构，选用的动力学模型开发工具和语言多根据各单位的积累或个人的喜好确定。大部分系统只针对特定任务设计，在开始新的仿真任务时，除动力学算法外，也需要对接口和仿真场景进行较大的适应性改造。近年来，随着仿真模型复杂度的提高，提出多学科融合仿真的概念，航空系统[5-6]较多使用 Simulink 和 modelica 建模，如赵杨杨[7]尝试了基于 FMI 的航空器模拟一体化开发系统。

本文介绍的空间操控任务控制仿真平台，基于通用模型接口标准 FMI 开发，可以将不同机构采用不同工具或语言开发的模型集成在一个平台上运行，实现多学科模型融合。通过对航天器、空间环境、任务策略等参数进行配置，能够快速构建新的空间操控动力学仿真任务。

1 仿真平台总体设计

空间操控任务控制仿真平台由航天器任务仿真平台、空间势态信息计算平台、视景显示平台和任务操控平台构成。基于 TCP/IP 协议的易用性和快速响应特点，采用基于 TCP/IP 的分布式网络结构作为系统架构，系统架构如图 1 所示。

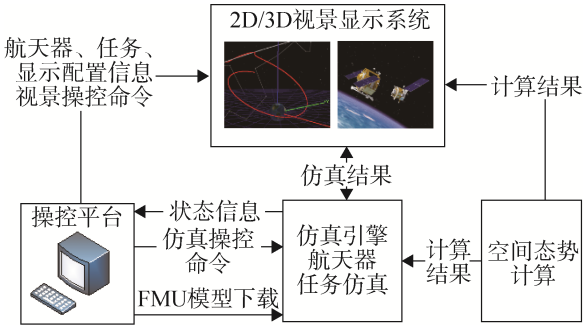

图 1 平台系统架构图

Fig. 1 System architecture of the platform

航天器任务仿真平台采用 FMI 模型接口标准开发，依据飞行任务脚本调度运行封装成 FMU 的空间操控任务的动力学模型，包括：航天器轨道动力学、姿态动力学、GNC 系统和空间操控机构多体动力学仿真模型，仿真结果通过网络发送给视景显示平台。

空间态势信息平台主要计算太阳、月亮以及太阳系其它行星的空间位置，并根据需要计算一定范围的空间碎片飞行轨迹等等，计算结果发送到仿真模型和视景显示系统用于计算或进行显示。

视景显示平台包括二维视景显示、三维远景模式显示和三维近景模式显示，分别用于二维星下点显示、卫星三维运动轨迹远距离显示、卫星交会对接及空间操作过程的近距离显示。视景显示平台基于 Unity3D 进行开发，Unity3D 是一个开源的视景引擎，具有较好的视觉效果，基于 Mono 的 C++/C# 脚本可以灵活处理数据，方便实现参数配置。此外，Unity3D 支持 VR 功能，提供通用的 VR 设备接口，便于后期系统进行交互仿真功能扩展。

任务操控平台用于建立工程，加载 FMU，设置任务参数，设置仿真速度，启动、暂停和停止仿真，显示和记录仿真结果。平台提供二维结果显示曲线，用户可以定制变量，以曲线形式输出仿真结果。此外，平台提供 Python 脚本的编辑和运行窗口，用以运行任务控制和任务评估任务。

DOI: 10.16182/j.issn1004731x.joss.201809018

第 30 卷第 9 期 系统仿真学报 Vol. 30 No. 9
2018 年 9 月 Journal of System Simulation Sep., 2018

2 平台关键技术

2.1 基于 FMI 标准的仿真引擎设计

通用模型接口标准 FMI 是一套独立于具体仿真工具的标准。FMI 标准最早由 Daimler AG 公司提出，主要目的是提高不同供应商和制造商提供的仿真模型之间的交互性。而后经 Modelisar[8-9]将其完善，推出 FMI1.0 和 FMI2.0 标准。遵循 FMI 标准的模型文件被称为 FMU(Function Mock-up Unit)。目前，几十种主流仿真平台，如 Matlab、Simulink、SimulationX、Dymola 等都在模型的导入导出功能模块中增加了对 FMI 标准的支持，可以直接将其模型导出成 FMU[10]。对于基于 C/C++/Fortran 开发的模型，FMI 技术组织提供了开发工具，可以帮技术人员将其模型封装成 FMU。

FMI 标准包括两部分内容：基于 C 语言标准的函数接口(API)和符合 XML 语法的模型描述文件 。使用 FMI 标准封装的 FMU 模型文件以 DLL 格式存在以便在各种基于 Windows 的平台上运行，而模型中的相关信息写入 xml 描述文件(Model Description.xml)中。

对 FMU 的仿真过程，如图 2 所示，包括：1)创建模型对象：使用 FMI 标准定义的创建模型对象接口函数 fmi_Instantiate Model 创建 FMU 对象，完成 FMU 的实例化，包括解析 FMU 文件，加载其中的模型文件(DLL 库)，读取 xml 模型描述文件并调用模型文件中的创建对象接口函数等；2)初始化模型对象：调用 FMI 标准的接口函数 fmiInitialize 进行初始化操作；3)按照设定步长调用 fmiDoStep()进行更新计算。

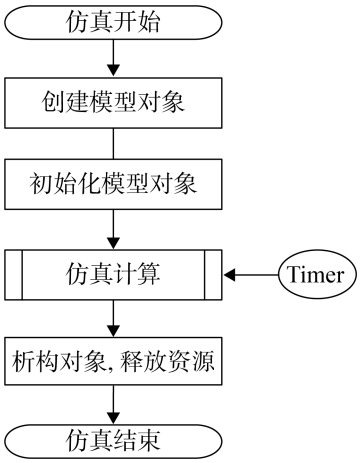

图 2 FMU 仿真过程示意图
Fig. 2 FMU simulation process

仿真引擎中调用的模型均以共仿真(Co-Simulation)模式运行。共仿真模式的特点是带有完整求解函数和固定的数据交换时长。为保证仿真过程中各计算单元(FMU)之间数据交换的正确性，仿真引擎采用“步调统一”的调度策略，即限制运行较快的模型，等待运行较慢的模型，待所有模型均执行完对应步骤后，进行数据交换，完成当前步骤的仿真过程，进入下一次仿真调度。图 3 为调度过程示意图。

为提高运算速度，仿真引擎采用多线程技术将 FMU 分配到计算机不同执行单元上计算，提高整体仿真速度。多线程分配策略如图 4 所示。

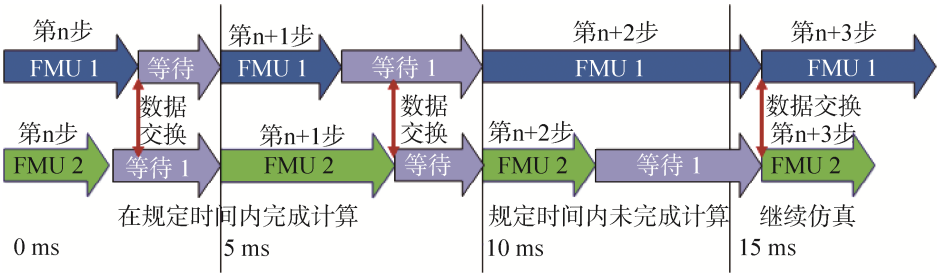

图 3 FMU 共仿真调度策略
Fig. 3 FMU co-simulation schedule strategy

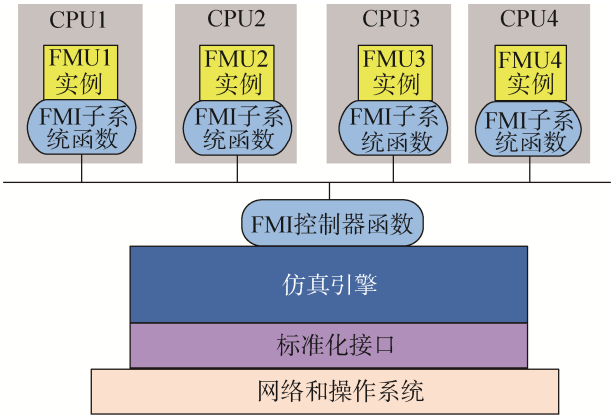

图 4 多线程分配策略
Fig. 4 Multi-thread assignation strategy

2.2 基于 Python 脚本的操控任务过程控制

一个典型的空间操控任务过程如图 5 所示，大致分为以下几个阶段：

(1) 在轨飞行段：太阳能帆板展开，激活相应的敏感器进行数据采集；根据任务逐步进行姿态机动、轨道调高、调相 S_1；

(2) 寻的/交会段：相对导航配置初始化，在轨实时规划交会对接的推力脉冲，利用小推力机动逐步接近至 S_2；

(3) 最终逼近段(从 S_3 到对接)：保持当地水平坐标系(LVLH)下的姿态，逐步接近 S_4 点，进行状态保持并与对接轴对准，渐近地从 S_4 点低速接近目标并在 S_{41} 点保持状态，逐步逼近实现与目标航天器对接；

(4) 在轨操作段：根据任务要求进行在轨操作；

(5) 分离段：与目标飞行器分离。

此外，飞行阶段还包括空间定向与安全机动：太阳定向、LVLH 姿态对准、偏航操纵、撤离或故障模式下的逃逸机动等。

在任务设计时，通过一系列飞行事件对任务过程进行控制，如图 6 所示。

航天器飞行过程的仿真，轨道姿态的调整以及在轨操作涉及的动力学计算模型，都封装成 FMU 运行在仿真平台上。而飞行条件的计算和设置，则可以通过脚本语言实现。FMI 标准提供了 FMU 模型的调用接口定义，用以设置 FMU 输入参数值或获取 FMU 输出结果。仿真平台内嵌 Python 语言运行环境，利用 Python 脚本可以获取仿真模型 FMU 的输出结果，计算飞行条件，或人为设置飞行条件，通过向指定的仿真模型 FMU 传递控制指令或控制参数的方式，实现任务控制。任务控制脚本与操控策略及仿真模型的关系如图 7 所示。

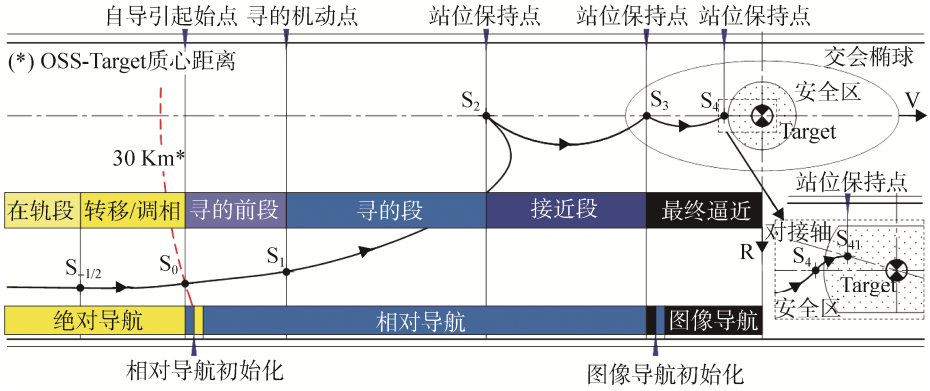

图 5 典型空间操控任务过程
Fig. 5 Process of typical space operations

https://dc-china-simulation.researchcommons.org/journal/vol30/iss9/18
DOI: 10.16182/j.issn1004731x.joss.201809018

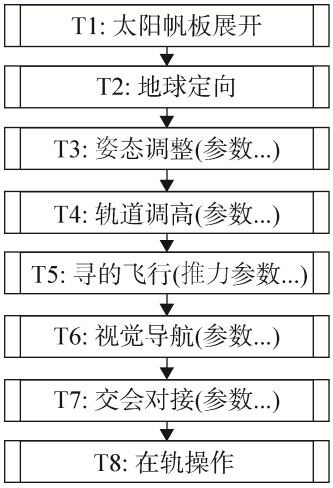

图 6 典型空间操控任务飞行事件
Fig. 6 Flight events of typical space operations mission

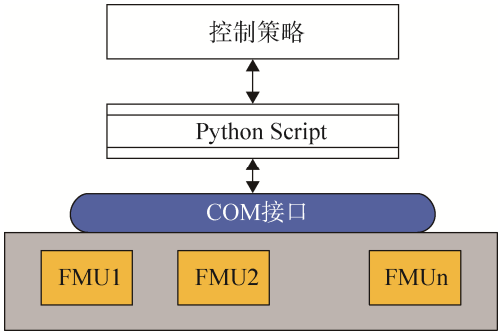

图 7 Python 脚本根据操控策略控制 FMU 运行示意图
Fig. 7 Python script control FMU based on operations strategy

2.3 基于 Unity3D 的虚拟场景开发

基于 Unity3D 的仿真视景软件的开发过程包括：1)使用三维建模工具或直接在Unity3D中创建3D 模型；2)制作虚拟场景，将模型加载到虚拟场景中；3)开发物体对象关联的脚本，使关联对象按照设计路径运动；4)场景调试通过后，打包发布，生成.unity3d 格式文件；5)在 Visual Studio 平台上创建主控程序，完成仿真数据的接收和处理，通过 Winform 类创建视景的交互界面，将前面生成的 unity3d 文件通过 COM 组件 UnityWebPlayerControl 嵌入 Winform，并通过该组件提供的函数 SendMessage 进行 Winform 与虚拟场景的交互，使用从仿真后台接收的数据来控制三维视景的驱动。

采用 Unity3D 开发空间操控三维虚拟场景，首先要能够正确进行坐标表示。Unity 3D 中使用左手笛卡尔坐标系来描述物体的坐标信息，与航天系统中通常使用的右手笛卡尔坐标系不同。必须要经过正确的坐标转换才能保证航天器位置和姿态的正确显示。对于 J 2 000 惯性系表示的航天器位置和姿态，其坐标系与 Unity3D 中的世界坐标系之间关系如图 8 所示。

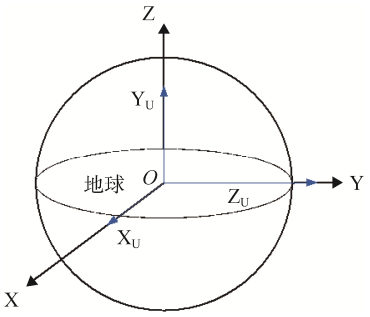

图 8 Unity 中世界坐标系与地心赤道惯性坐标系
Fig. 8 World and earth equator J 2000 Coordinates

为在 3D 场景中正确描绘航天器的位置和姿态，采用下面的转移矩阵进行位置和姿态的坐标变换，将 J 2 000 坐标转换为世界坐标。

$$\begin{bmatrix} X_U \\ Y_U \\ Z_U \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

此外，与航天器常规飞行任务仿真相比，空间操控过程更注重显示飞行器的运动细节。由于显示卫星常规的绕地飞行轨迹时，对地球和卫星均按照较大的比例进行缩小，在需要显示空间操控细节时，若按照通常的方式对场景进行放大，则由于数据精度带来的误差被放大，会导致场景中的物体剧烈抖动。虽通过精度平滑可以减少抖动，但不能彻底解决问题。本系统为了逼真显示航天器操控细节，单独设计了近景场景：以主航天器质心为坐标原点，采用径切法坐标系，相机部署在坐标系的 N 轴，通过前面介绍的转移矩阵转换成世界坐标系中的位置和方向，选取适当的显示比例，可以清晰稳定显示航天器编队、交会对接以及空间操控过程。

2.4 支持复杂操控机构的虚拟场景通用化设计

空间操控机构的研发，目前正是业界的研究热点，到目前为止，已经研制出机械臂、网、绳等形态繁多的操控机构试验样机。为支持不同任务场景对不同结构外型的航天器及其操控机构进行实时仿真，定义 XML 映射文件，建立航天器本体及其附属机构 3D 模型与仿真模型输出变量间的数据接口映射关系(如图 9 所示)。

模型和视景显示平台在收发数据时，按照映射文件的约定解析或打包数据，即可完成信息交换。

从 2.3 节虚拟场景的开发流程可知，根据映射文件进行数据解析的任务是在 C#主控程序中进行的，经主控程序解析的数据必须与.unity3d 场景中的对象通过对象名或对象 ID 进行关联，其前提是场景中已存在 3D 模型。为支持从零开始快速构建新的仿真视景场景，我们开发了一个图形界面的配置工具，用于绘制或导入 3D 模型(如图 10 所示)，组装航天器子部件或机构；通过代码后台创建虚拟场景，定义模型对象，生成.unity3d 文件以及 3D 模型与仿真模型输出变量之间的映射文件。

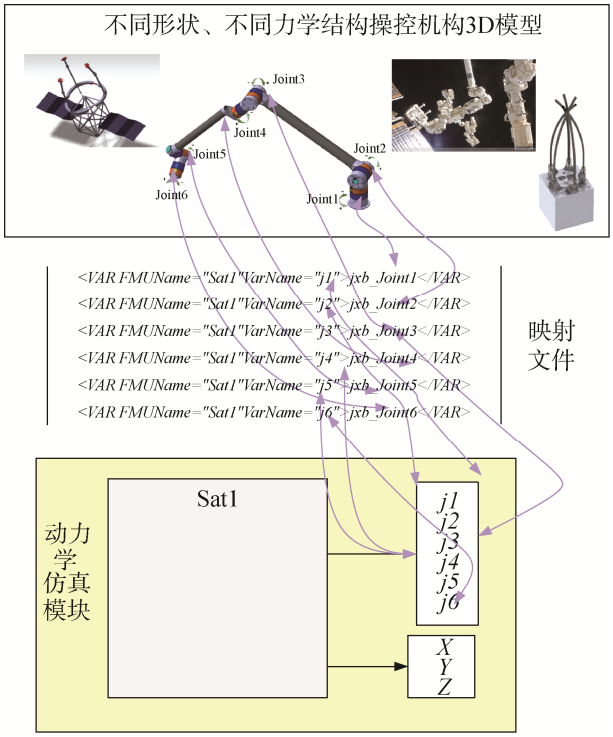

图 9 映射文件与 3D 模型及仿真模块接口之间的关系
Fig. 9 Mapping file links 3D models and dynamic output

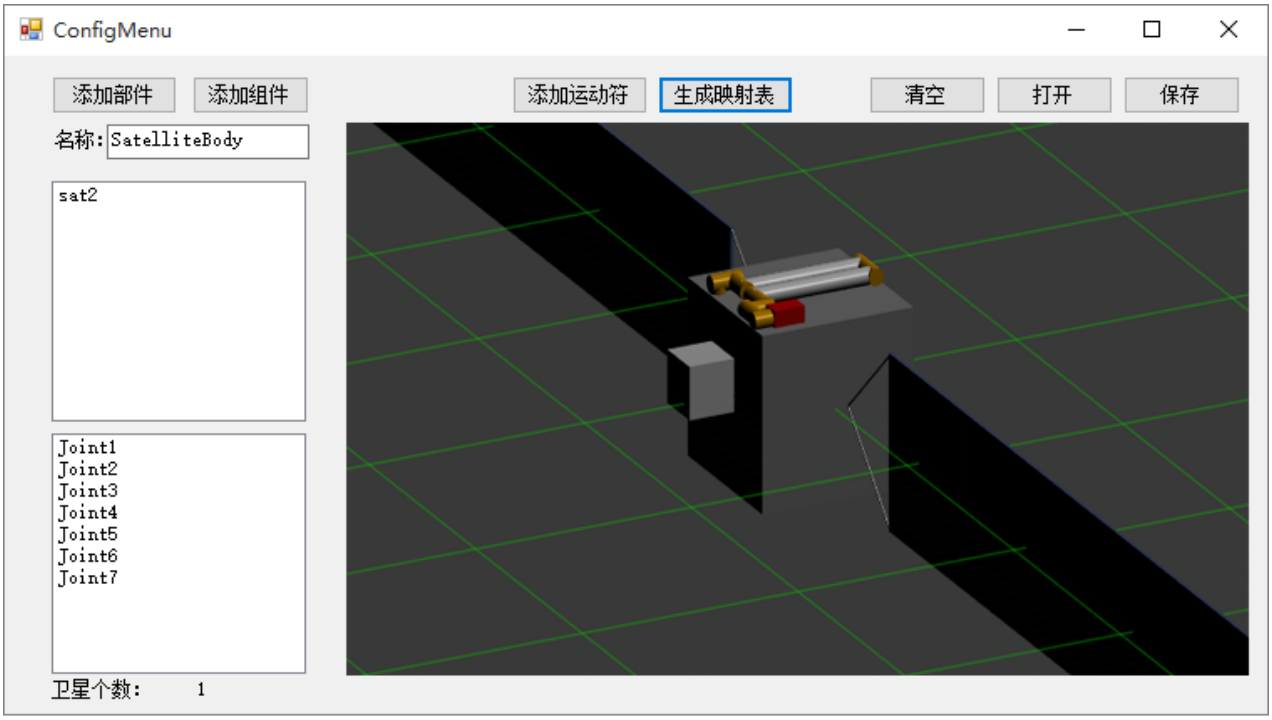

图 10 3D 模型接口配置工具界面
Fig. 10 Interface of 3D models configuration tool

https://dc-china-simulation.researchcommons.org/journal/vol30/iss9/18
DOI: 10.16182/j.issn1004731x.joss.201809018

3 空间操控典型任务演示验证

在这个验证任务中，主星带两颗子星，飞行过程中释放子星进行三星编队，然后飞向目标飞行器进行交会，利用机械臂完成对接。其任务过程如下：

T=1 000 s：释放子星 a;

T=2 000 s：释放子星 b;

T=3 000 s：按照逼近策略，开始小推力点火，逼近目标飞行器;

T=5 000 s：小推力点火结束;

T=6 000 s：机械臂展开，经历时间 100 s;

T=6 120 s：进行机械臂抓捕路径规划;

T=6 200 s：控制机械臂抓住目标飞行器特定部位进行操作;

T=300 s：与目标飞行器完成对接。

该试验的 3D 场景通过配置工具建立。任务控制流程及仿真参数通过 Python 脚本加载，采用 Simulink、SimulationX 和 C 实现的动力学模型被封装成 FMU，导入仿真平台进行仿真，与 Python 脚本一起运行，精确完成了整个仿真过程。图 11 为主星与目标飞行器相对位置变化曲线，图 12 为主星机械臂位置变化曲线，图 13~18 为视景显示平台显示的任务仿真过程近景视图画面。

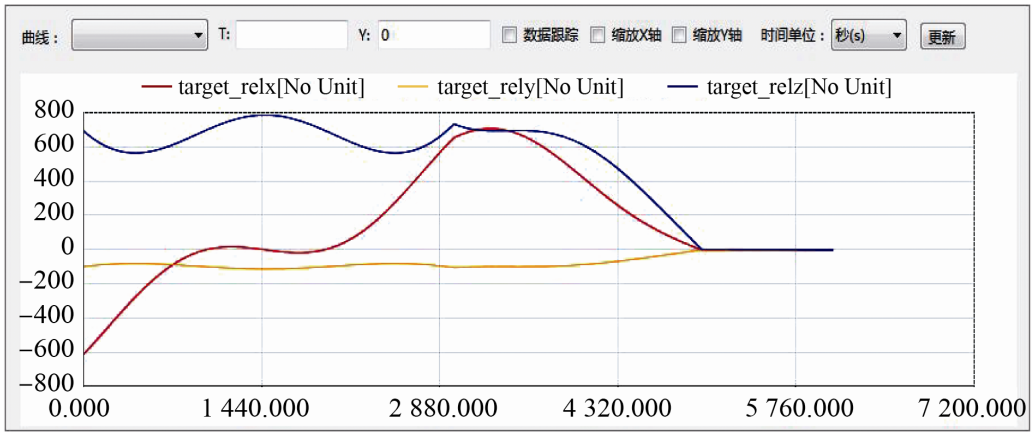

图 11 主星与目标星相对位置曲线

Fig. 11 Relative position between main Sat and Target

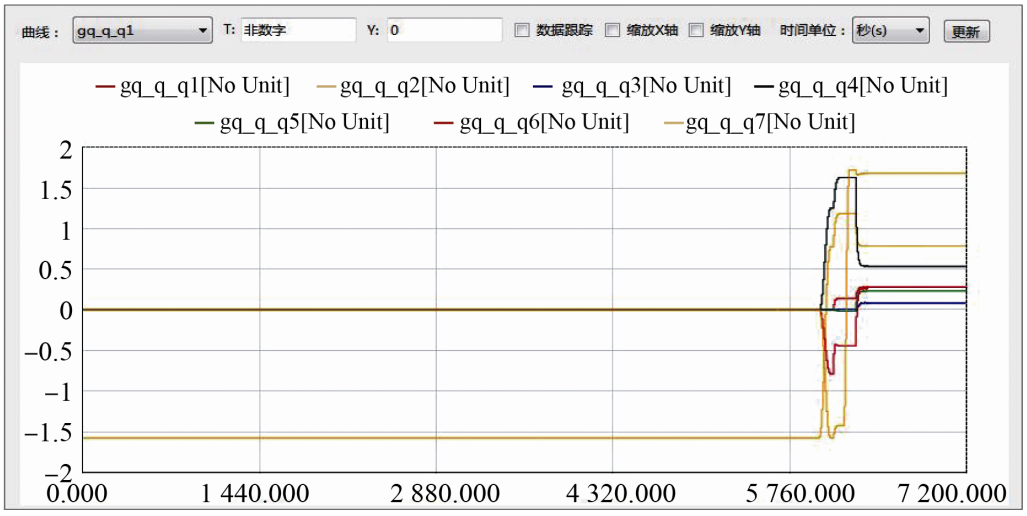

图 12 主星机械臂位置变化曲线

Fig.12 Curve graph of main Sat robot position

图 13 开始阶段，携带二个子星初始飞行
Fig. 13 Initial flight with 2 sub Sats

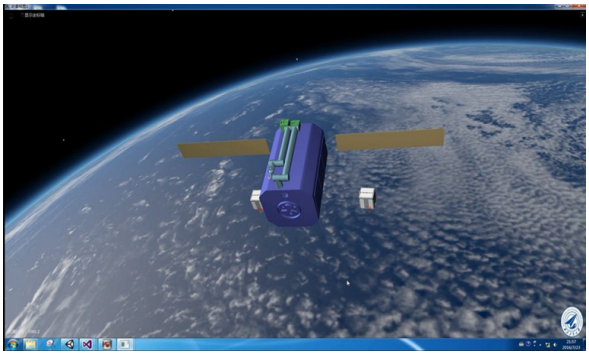

图 14 1 000 s 第 1 颗子星分离
Fig. 14 1 th sub Sat separates at 1000s

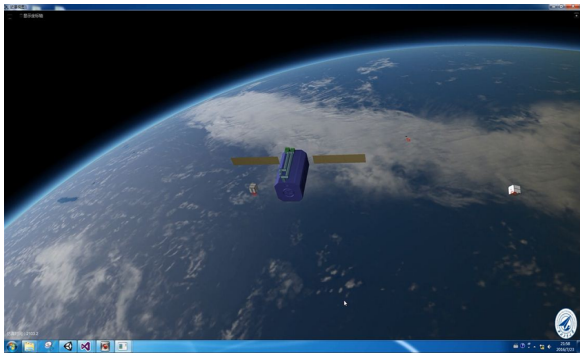

图 15 2 000 s 第 2 颗子星分离，3 颗星开始编队飞行
Fig. 15 2nd sub Sat separates at 2 000 s, the three Sats to format

图 16 5067s 主星与目标星保持相对位置和姿态不变
Fig. 16 Relative position and attitude of main sat and target sat keep stable at 5067s

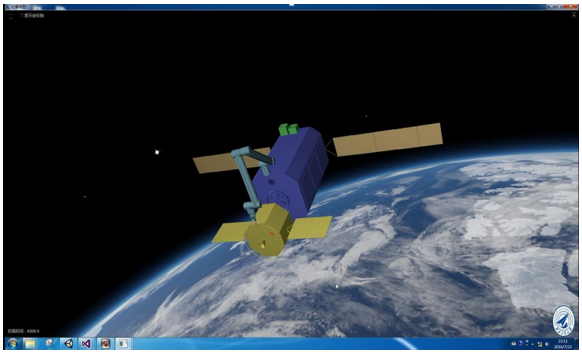

图 17 6 049 s 主星机械臂抓住目标星挂环
Fig. 17 The robot on main Sat seizes the ring on the target Sat at 6049s

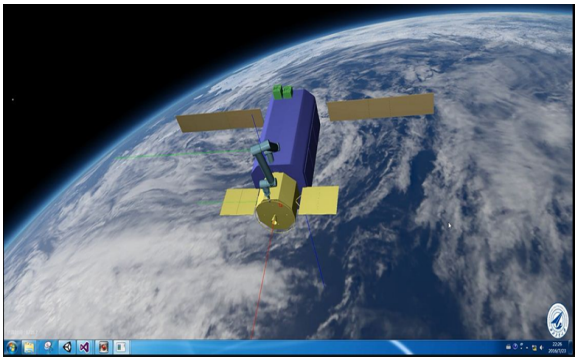

图 18 6 300 s 与目标飞行器完成对接
Fig.18 Completion of rendezvous and docking with target aircraft at 6300s

4 平台性能测试结果

4.1 精度比较

经测试，模型加载后由平台输出的变量与模型独立运算的输出变量之间的误差在 10^{-8} 以下，可以认为平台不影响模型的精度。考虑到任务仿真精度主要取决于模型和算法，选择了平台最常用的带 J2 项的轨道外推算法与 GMAT 工具(General Mission Analysis Tool 是NASA与商业公司合作研制的、专用于轨道任务分析的高精度工具)进行比较，分别针对低轨太阳同步卫星、地球同步卫星、中高轨大倾角大椭圆三种较有代表性的轨道进行比较，其轨道参数见表 1。

https://dc-china-simulation.researchcommons.org/journal/vol30/iss9/18
DOI: 10.16182/j.issn1004731x.joss.201809018

第 30 卷第 9 期 系统仿真学报 Vol. 30 No. 9
2018 年 9 月 Journal of System Simulation Sep., 2018

表 1 航天器轨道参数(历元时间 2015-01-09T04:00:00.0)
Tab. 1 Initial orbit element(2015-01-09T04:00:00.0)

Sat	半长轴/km	偏心率	倾角/°	升交点赤经/°	近点点幅角/°	真近点角/°
1	7178.1	0	98.608	10.886	0	0
2	42166.3	0	0	0	0	110
3	12 578.1	0.461	63.435	260	270	90

注：表中 Sat1、Sat2、Sat3 分别为低轨太阳同步卫星、地球同步卫星、中高轨大倾角大椭圆卫星。

太阳同步轨道比对结果见图 19~20，仿真 24 h，最大位置误差 0.51 km，速度误差 5.3X10^{-4} km/s。

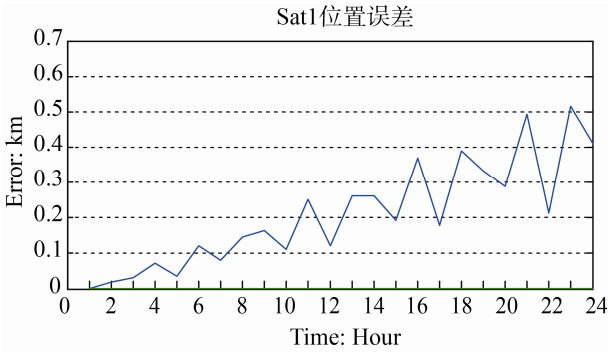

图 19 Sat1 卫星位置误差
Fig. 19 Position error of Sat1

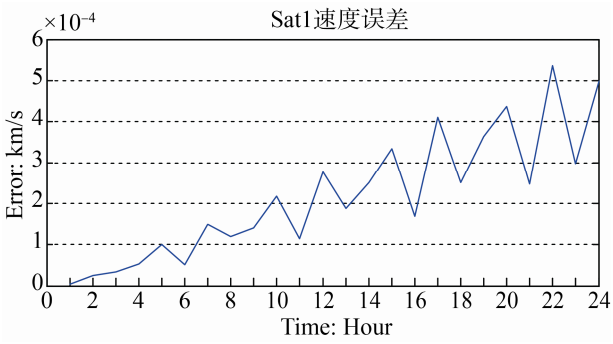

图 20 Sat1 速度误差
Fig. 20 Velocity error of Sat1

地球同步轨道比对结果如图 21~22 所示，仿真 24 h，最大位置误差 0.014 4 km，速度最大误差 6.58 ×10^{-6} km/s。

中高轨大倾角大椭圆轨道比对结果如图 23~24 所示，仿真 24 h，最大位置误差 0.69 km，速度最大误差 4.5 ×10^{-4} km/s。该结果表明，仿真平台的常用外推算法能够满足航天器正常飞行、编队、交会对接所需初始轨道的精度要求。

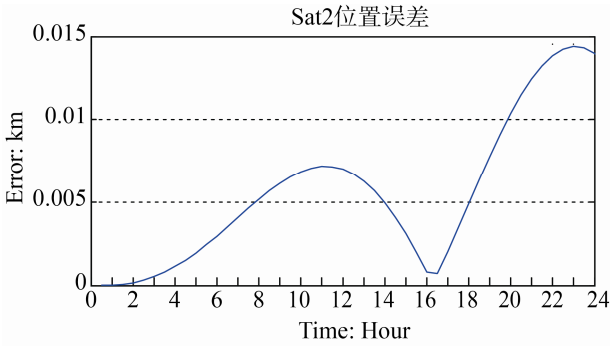

图 21 Sat2 位置误差
Fig. 21 Position error of Sat2

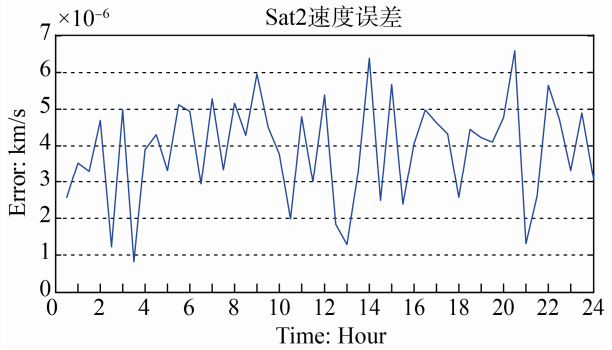

图 22 Sat2 速度误差
Fig. 22 Velocity error of Sat2

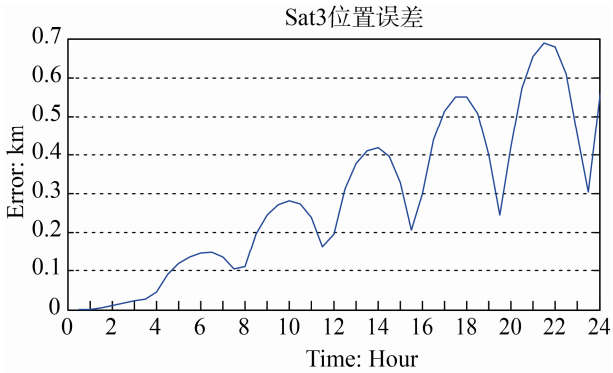

图 23 Sat3 位置误差
Fig. 23 Position error of Sat3

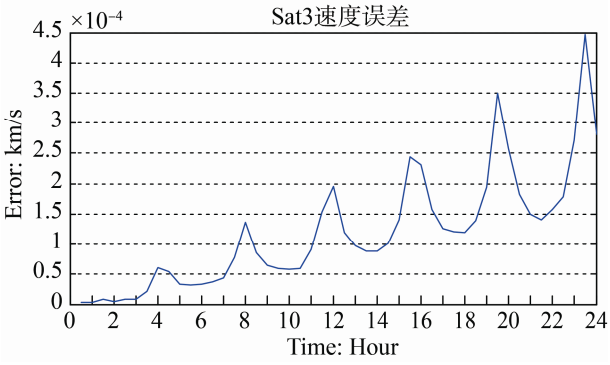

图 24　Sat3 速度误差
Fig. 24　Velocity error of Sat3

4.2 时间性能测试

此仿真平台是按照准实时平台的标准设计，没有采用硬件时钟进行管理，但要求软件的延时不能超过一个积分周期。为此，设计几种典型应用，对每积分步长的 CPU 消耗进行测试。测试环境为：1 台 Dell 工作站，CPU E5-2620VS(2.4G、双 CPU)，内存 32G。得到如下测试结果。

case 1： 一个积分器进行运行轨道外推，每步长大约 60 个加乘运算，积分步长为 50 ms，每步长 CPU 使用情况如图 25 所示。此时每步时间周期内的 CPU 消耗均值为 100 ns，峰值为 500 ms。

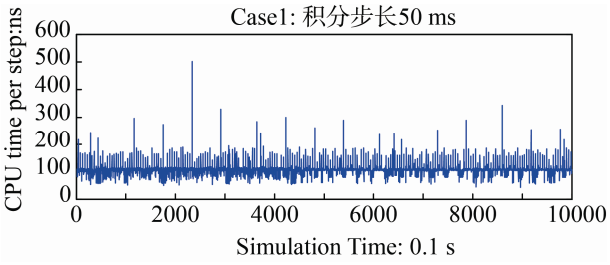

图 25　case1 计算时间散点图
Fig. 25　CPU time of case1

case 2：4 个积分器进行运行轨道和姿态计算，每步长大约 800 个加乘运算及 20 个逻辑运算，积分步长为 100 ms. 每步长 CPU 使用情况如图 26 所示。此时每步时间周期内的 CPU 消耗均值为 1200 ns，即 1.2 ms。

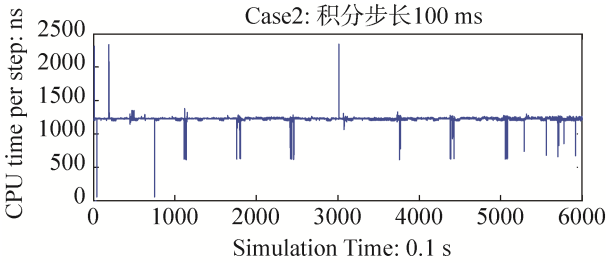

图 26　case2 计算时间散点图
Fig. 26　CPU time of case2

case 3：8 个积分器进行运行轨道和姿态计算、轨道控制计算，机械臂 7 自由度控制计算，每步长大约 5 000 个加乘运算及 200 个逻辑运算，积分步长为 100 ms. 每步长 CPU 使用情况如图 27 所示。此时每步时间周期内的 CPU 消耗均值为 83 ms，峰值 88 ms。

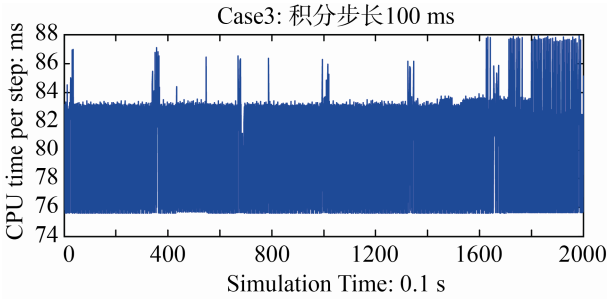

图 27　case3 计算时间散点图
Fig. 27　CPU time of case3

从以上结果看，在目前的计算机配置下，对于中等复杂的模型，每积分步长包含的运算在 5 000 个以下时，系统基本能够满足准实时性需求。

5　结论

本文介绍的航天器空间操控任务控制仿真平台，由航天器任务仿真平台、空间态势信息计算平台、视景显示平台和任务操控平台组成。航天器仿真平台基于 FMI 标准开发，支持多学科模型耦合；基于 FMI 标准开发了 API 接口，通过运行 Python 脚本调用 API 接口操控仿真模型，以实现不同的任务策略仿真。同时，仿真引擎采用多线程调度算法，保证了系统的高效性和实时性。针对空间操控模型结构多样性的特点，通过装载航天器及其部件

https://dc-china-simulation.researchcommons.org/journal/vol30/iss9/18
DOI: 10.16182/j.issn1004731x.joss.201809018

第 30 卷第 9 期 系统仿真学报 Vol. 30 No. 9
2018 年 9 月 Journal of System Simulation Sep., 2018

的 3D 模型，后台建立虚拟场景，配置场景对象与仿真计算结果的映射关系，实现了仿真平台的通用性和可扩展性。通过对航天器特定操控任务进行任务想定、流程设计、仿真运行、视景显示以及任务评估，验证了在该平台上能够通过对航天器、空间环境、任务策略等仿真参数进行自由配置 ，按照所配置的空间操控任务策略对航天器空间操控任务进行仿真实验；通过测试平台所支持典型算法的计算精度以及仿真过程的实时性，验证了平台的主要性能指标。

参考文献：

[1] NASA. On-orbit satellite servicing study: Project report. National Aeronautics and Space Administration. Goddard Space Flight Center, 2010.

[2] 安昊, 屈桢深, 王常虹. 基于 Matlab 的交会对接全数字仿真系统[J]. 系统仿真学报, 2015, 27(6): 1227-1234.
An Hao, Qu Zhenshen, Wang changhong. Rendezvous and Docking Digital Simulation System Based on Matlab[J]. Journal of System Simulation, 2015, 27(6): 1227-1234.

[3] Wang Mingming. A DDS Based Real-Time Simulation Architecture For Space Robotic Tele-Operation, IAC-13-D1.4.4

[4] 周剑勇, 蒋自成, 王跃峰. 飞行器机动飞行分布式仿真系统研究[J]. 系统仿真学报, 2011, 23(11): 2391-2394.
ZHOU Jian-yong, JIANG Zi-cheng, WANG Yue-feng. Research on the Aerocraft Maneuver Flighting Distributed Simulation System[J]. Journal of System Simulation, 2011, 23(11): 2391-2394.

[5] 郭漼, 龚光红, 韩亮，等. 飞行控制系统建模与仿真研究[J]. 系统仿真学报, 2013, 25(增 1): 46-51
GUO Chan, GONG Guang-hong, HAN Liang, et al. Research on Modeling and Simulation of Flight Control System[J]. Journal of System Simulation, 2013, 25(S1): 46-51.

[6] 李永林, 曹克强, 胡良谋, 等. 基于 Modelica 的飞机液压系统热力学建模与仿真[J]. 系统仿真学报, 2014, 26(12): 2828-2833.
Li Yonglin, Cao Keqiang, Hu Liangmou, et al. Thermal-hydraulic Modeling and Simulation of Aircraft Hydraulic System Based on Modelica[J]. Journal of System Simulation, 2014, 26(12): 2828-2833.

[7] 赵杨杨. 基于 FMI 的一体化仿真平台及其在航天工程中的应用[D]. 哈尔滨: 哈尔滨工业大学, 2013: 1-70.
Zhao Yangyang. Integrated Simulation Platform Based On Functional Mockup Interface And Its Application On Aerospace Engineering[D]. Harbin: Harbin Institute of Technology, 2013: 1-70.

[8] MODELISAR. Functional Mock-up Interface for Co-Simulation. ITEA 2-07006 2010: 20-33.

[9] MODELISAR. Functional Mock-up Interface for Model Exchange. ITEA 2-07006 2010: 17-23.

[10] Christian A, Johan A, Claus F. Import and Export of Functional Mock-up Units in Modelica.org[D]. Lund University, Sweden, 2011: 1-5.

(上接第 3365 页)

[4] 张敏. 基于离散事件系统仿真的曲轴生产线仿真与优化[D]. 长沙: 湖南大学, 2013.
Zhang Min. Simulation and Optimization of Crankshaft Production Line Based on Discrete System Simulation [D]. Changsha: Hunan University, 2013.

[5] 邵明珠，周泓. 数控加工流程的 Arena 仿真建模与优化研究[J]. 成组技术与生产现代化, 2010, 27(2): 1-5.
Shao Mingzhu, Zhou Hong. Research on Arena Simulation Modeling and Optimization of CNC Machining Process[J]. Group Technology & Production Modernization, 2010, 27(2): 1-5.

[6] 潘燕春, 周泓, 冯允成. 基于 Arena 的车间作业排序问题建模方法及其仿真优化系统设计[J]. 计算机集成制造系统, 2006, 12(3): 389-394.
Pan Yanchun, Zhou Hong, Feng Yuncheng. Modeling & Simulation Optimization Systems Design for Job Shop Scheduling Based on Arena[J]. Computer Integrated Manufacturing System, 2006, 12(3): 389-394.

[7] 于新. 多品种、小批量离散制造企业生产能力研究[D]. 长春: 吉林大学, 2007.
Yu Xin. Research of Publication Capability Based on the Discrete Manufacturing Enterprise of Multi-categories and Small-lot[D]. Changchun: Jilin University, 2007.

