doi: 10.7690/bgzdh.2015.09.001

# 高机动性装备半物理仿真系统关键技术研究

王国庆，薛鸿臻，殷周扬
（长安大学道路施工技术与装备教育部重点实验室，西安 710021）

**摘要**：为了解决目前国内高机动性装备缺少具有自主知识产权的半物理仿真系统的现状，提出一种具有完全自主知识产权的高机动性装备控制器开发半物理仿真系统架构。以某型号坦克的火控控制器自主知识产权开发与测试环境为研究对象，重点研究高机动性装备动力学模型的嵌入，通过数据驱动的方式解决控制器与设备动力学系统联动过程的后处理虚拟现实演示 2 个关键问题，设计开发一个原型半物理仿真系统，并进行测试验证。测试结果表明：该架构能实现火控控制器的开发与应用，大幅提高开发效率。

**关键词**：半物理仿真；虚拟现实；动力学模型；数据驱动；后处理
**中图分类号**：TJ811 **文献标志码**：A

# Key Technology Research of High Mobility Equipment Hardware-in-loop Simulation System

Wang Guoqing, Xue Hongzhen, Yin Zhouyang
(*Key Laboratory of Highway Construction Technique & Equipment of Ministry Education, Chang'an University*, *Xi'an* 710021, *China*)

**Abstract:** For solving lack of independent intellectual property rights hardware-in-loop simulation system of high mobility equipment, put forwards hardware-in-loop simulation system frame of high mobility equipment controller with independent intellectual property rights. Taking certain type tank fire controller development and test environment as research object, focus on analyzing embedded kinetic model of high mobility equipment, solve 2 key problems of post processing virtual reality in linkage process of controller and equipment kinetic system through data driving. Design and develop a prototype hardware-in- loop simulation system, then carry out test verifying. The results show that, the frame can realize fire controller development and application with high efficiency improvement.

**Keywords:** hardware-in-loop simulation; virtual reality; kinetic models; data driven; post-processing

## 0 引言

随着高性能装备智能化的不断提高，半物理仿真在装备控制器的开发与应用中的高效性显得尤为重要。目前已有的开发模式有：文献[1]基于 Matlab/RTW 的半物理仿真方法对四足机器人控制模型进行了仿真分析；文献[2]开发了用于对飞控计算机进行全面测试的基于嵌入式操作系统 RT Linux 及仿真工具软件 Matlab 的半物理仿真系统；文献[3]采用 LabVIEW 对船用双轴燃气轮机进行仿真建模。大多数仿真系统都是在国外软硬件的环境下搭建起来的，因此系统没有自主知识产权，难以进行扩展开发。为此，笔者通过对以上方案的分析研究，在文献[4]通用机械半物理仿真软件 V1.0 和文献[5]用于半物理仿真的信号转换器专利等成果基础上，提出了一个具有完全自主知识产权的用于高机动性装备火控控制器开发与测试应用的半物理仿真系统架构，主要解决了如何将设备动力学模型嵌入到半物理仿真系统中实现与控制器联动的问题，并完成了整个半物理仿真过程中的实时虚拟现实后处理演示的研究，最后以此为基础开发了半物理仿真原型系统，并进行了初步测试。

## 1 半物理仿真架构

高机动性半物理仿真系统主要由上位机软件系统和硬件系统组成，如图 1 所示。硬件部分包含设备火控控制器和信号转换器，实现控制器与设备动力学模型之间的信号交互；软件部分包含半物理仿真主逻辑调度模块、设备动力学模型管理模块、实时虚拟现实后处理模块和人机界面 4 个部分，实现信号交互流程的管理和可视化表达。

整个半物理仿真系统的信号交互过程如下：用户通过配置人机交互界面提供控制目标、干扰等参数的设定，以 300 μs 为周期，启动设备动力学模型运算过程，在周期中设备动力学模型完成一次迭代并将状态信号(实际系统的传感器信号)通过数字总线接口传递给硬件部分中的信号转换器，信号转换器将这些数字信号通过自身的 DAC 或者数字信号模块转换为物理量，传输给设备控制器，设备控制器通过其信号采集模块完成信号采集，并根据控制律进行结算，最后发出控制命令，该控制命令为物理量并通过信号转换器的采集系统转换为数字量，

**收稿日期**：2015-06-02；**修回日期**：2015-07-04
**作者简介**：王国庆(1972—)，男，陕西人，工学博士，教授，从事机器人动力学与控制系统研究。

再次通过数字总线传递给上位机中的动力学模型作为下一次计算的初始值，周而复始，直至完成控制率的测试，系统达到稳态为止。

在整个信号交互的过程中，上位机的实时虚拟现实后处理模块以状态信号和控制信号为基础对设备的虚拟现实模型进行驱动，从而实现可视化表达。

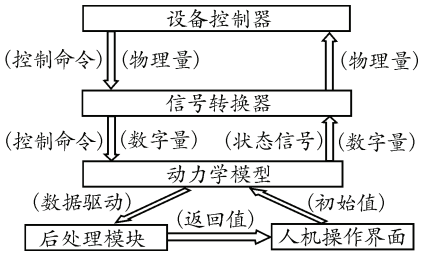


图1 高机动性设备控制器半物理仿真开发系统架构

由此可见，要实现该架构的正确运行，如何建立设备的动力学模型并将该模型嵌入到架构中是关键点，其后如何对数据进行可视化表达也至关重要。

## 2 设备动力学模型的嵌入

设备动力学模型的建立可以有多种技术路径：一是基于多体动力学方法完全采用通用计算机语言开发；二是采用商用多体软件进行模型建立。第1种方法，由于设备的复杂性，模型开发周期很长；第2种方法，如ADAMS、Recurdyn等软件都可以建立模型，但是模型的代码是不对外开放的，仅和Matlab等软件具备接口。因此要实现自主知识产权的半物理仿真系统是快速的，并且能输出模型代码的软件是第1选择。

从目前来看：Maplesim软件中对移动式装备进行整体建模并添加驱动进行仿真，输出动力学模型C代码，对该C代码进行封装成动态链接库DLL后，即可将该DLL嵌入到前述的半物理仿真系统中。下面以实例说明这种方法的建立过程。

### 2.1 建立虚拟样机

以某型号履带式战车坦克为研究对象。它主要由坦克车身、炮塔及炮筒组成，通过控制左右履带驱动、炮塔驱动，以不同转速来驱动左右履带从而实现坦克的各种运动形态以及驱动炮塔的旋转运动，最终建立虚拟样机[6-7]三维CAD模型如图2。

在三维模型基础上，采用Maplesim设计与火控系统有关的动力学虚拟样机模型如图3所示，包含了2个平移运动副和1个转动运动副，其中包括沿$Y$轴的横向运动、沿$X$轴的纵向运动以及绕$Z$轴的旋转运动，由于是对火控系统进行测试，因此机构传动系统进行了简化，其余部分由于与火控关系较小也予以忽略。

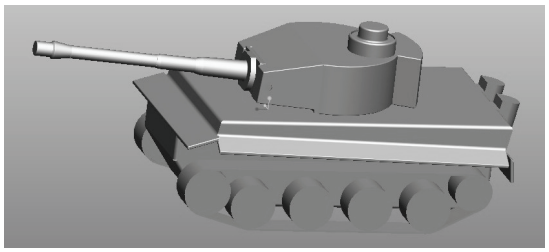

图2 某型号坦克实物模型

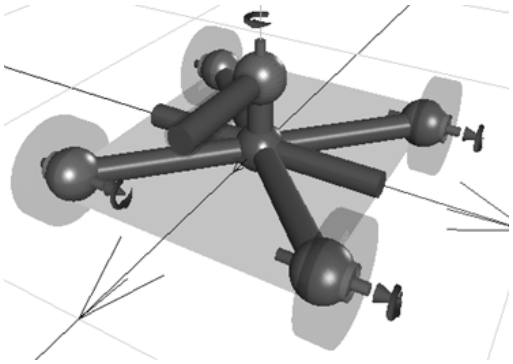

图3 坦克的动力学模型图

### 2.2 动力学模型的DLL转换

创建好的动力学模型仿真结束后，首先创建为C代码。部分C代码如图4所示。

```
*************************************************/
#ifdef WMI_WINNT
#define EXP __declspec(dllexport)
#else
#define EXP
#endif
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <mplshlib.h>
static MKernelVector kv;
EXP ALGEB M_DECL SetKernelVector(MKernelVector kv_in, ALGEB args)

/*************************************************
* Variable Definition for System:

* State variable(s):
*    x[ 0] = `Main.DFPSubsyslinst.s_C1`(t)
*    x[ 1] = diff(`Main.DFPSubsyslinst.s_C1`(t),t)
*    x[ 2] = `Main.DFPSubsyslinst.s_C3`(t)
*    x[ 3] = diff(`Main.DFPSubsyslinst.s_C3`(t),t)
*    x[ 4] = `Main.DFPSubsyslinst.s_C5`(t)
*    x[ 5] = diff(`Main.DFPSubsyslinst.s_C5`(t),t)
*    x[ 6] = `Main.DFPSubsyslinst.s_C7`(t)
*    x[ 7] = diff(`Main.DFPSubsyslinst.s_C7`(t),t)
*    x[ 8] = `Main.DFPSubsyslinst.theta_C1`(t)
*    x[ 9] = diff(`Main.DFPSubsyslinst.theta_C1`(t),t)
*    x[10] = `Main.DFPSubsyslinst.theta_C3`(t)
*    x[11] = diff(`Main.DFPSubsyslinst.theta_C3`(t),t)
*    x[12] = `Main.DFPSubsyslinst.theta_C5`(t)
*    x[13] = diff(`Main.DFPSubsyslinst.theta_C5`(t),t)
*    x[14] = `Main.DFPSubsyslinst.theta_C7`(t)
*    x[15] = diff(`Main.DFPSubsyslinst.theta_C7`(t),t)
```

图4 部分C代码生成图

在生成C代码后，采用Win32 Dynamic-Link Library方式创建Non-MFC DLL动态链接库[8]，采用缺省的DLLMain()函数就可以方便、正确地初始化所要调用的动态链接库(DLL)文件。在生成DLL过程中，首先确定导出函数关键字_declspec(dllexport)来创建Non-MFC DLL动态链接库，简要过程如下：

建立Win32 DLL空项目TankeDLL后，使用导

出函数关键字_declspec(dllexport)来创建 DLL 文件。

在 TankeDLL.h 中输入：

extern "C" _declspec(dllexport)　函数名 1；

extern "C" _declspec(dllexport)　函数名 2；

……

在 TankeDLL.cpp 中将 C 代码放入，并在头文件中加入：

#include <TankeDLL.h>

动态链接库编译成功后，就可以生成 TankeDLL.dll 和 TankeDLL.lib 文件。

完成 DLL 的生成后，就可以在主逻辑管理模块中使用该动态链接库完成对动力学模型的调用。

## 3　后处理设备虚拟现实展示

半物理仿真中，系统的各个状态参数和控制参数都是用户关心的对象，实时展示这些动态数据既可以采用实时曲线的方式，又可以采用基于虚拟现实的可视化展示模式。由于实时曲线的直观性不足，笔者采用 Unity3D 设计独立的设备虚拟现实模型[9]，并建立与前述上位机的数据通信联系，以实时数据驱动 VR 模型的运动，从而大大增加仿真的可视性。

### 3.1　设备虚拟现实模型的建立流程

首先通过 3DMAX 对坦克建模，并且以 FBX 格式导入到 Unity3D 场景中，再通过对基础场景的建立，完善场景的开发与脚本的编写并调试，图 5 为建立好的场景。

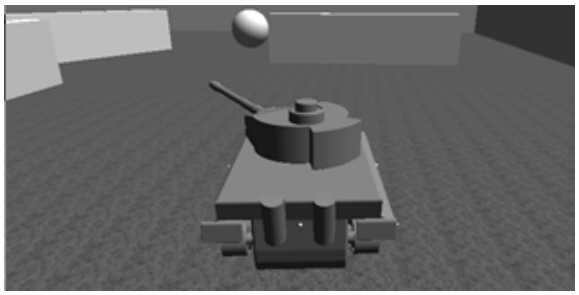

图 5　Unity3D 场景

### 3.2　设备 VR 模型的数据驱动

数据驱动的原理是在仿真系统计算过程中，设备控制器每次发出的控制向量都会通过信号转换器的转换与动力学模型的计算得到数字量的值，并实时保存到数据库中。这样，模型的每一个关节在每一个时刻都会对应到一个驱动数据，从而保证了虚拟现实的真实性与实时性。通过模型脚本的编写在 Update 函数(JavaScript 语言)中每一帧都调用数据库中的参数，周而复始，就形成了模型的虚拟动画演示。脚本中的部分代码如图 6 所示。

```
void Update()
{
  for (int t = 0; t < table[0].row[t][5].length; t++)
  {
     if (table[0].row[t][8] = 1)
     {
      dog[table[0].row[t][1]].transform.RotateAround(table[0].row[t][2].transform.position,
        Vector3.table[0].row[t][3], Time.deltaTime * (float)Convert.ToDouble(ds.Tables[0].Rows[t][4]));

     }
     if (table[0].row[t][8] = 0)
     {
       dog[table[0].row[t][1]].transform.Translate(Vector3.table[0].row[t][3] * Time.deltaTime *
        (float)Convert.ToDouble(ds.Tables[0].Rows[t][4]));
     }
  }
}
```

图 6　脚本部分代码

## 4　原型系统设计

### 4.1　原型系统的基本组成

高机动性装备半物理仿真系统的主要设计目标是实现操作人员通过 PC 操作平台的各个控制按钮完成需仿真装备的运动曲线图，同时可以直观地看到运动曲线背后真实的动画演示。高机动性装备半物理仿真系统的软件部分包括工程配置、控制器配置与 Unity3D 配置等 3 大部分组成，用户在人机交互模块操作可以得到仿真装备的运动曲线以及实时状态下的运动动画的再现。根据移动式装备半物理仿真系统设计要求，移动式装备上位机软件平台总体框架图如图 7 所示。

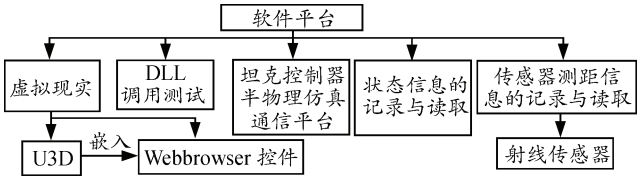


图 7　软件平台总体框架图

### 4.2　人机操作界面设计

根据所要实现的功能要求，用户界面总体结构主要由虚拟现实、DLL 调用测试、坦克控制器半物理仿真通信平台、状态信息的记录与读取、传感器测距信息的记录与读取 5 部分组成。其中虚拟现实主要是将 Unity3D 中建立的场景打包发布成 Web 网页版，再利用 Webbrowser 控件把网页嵌入到 WPF 应用程序的主窗口下。传感器测距采用射线传感器(代码控制)，其他部分都是以 Form 窗体实现相应的功能。软件平台的用户界面如图 8 所示。

## 5　结束语

笔者以某型号坦克的火控控制器自主知识产权开发与测试环境为研究对象，提出了高机动性装备火控控制器半物理仿真系统框架，并完成了原型系统的设计开发，采用商用 CAE 软件快速建立动力学模型 C 代码并以动态链接库的方式解决了设备动力学模型的嵌入问题；

(下转第 9 页)