

基于 RTW 的实时多任务无人机仿真系统设计

陈朋, 李春涛

(南京航空航天大学自动化学院, 南京 211106)

摘要: 针对无人机仿真系统对高实时性、多任务的需求以及传统手工代码开发设计难度大、开发周期长等问题,提出了一种使用 RTW 结合 C 代码进行 VxWorks 平台的多任务设计方法。使用 Simulink 仿真环境开发无人机仿真系统;将系统任务划分为系统管理任务、模型解算任务、调度管理任务、接口任务并分别进行详细设计;最后使用 RTW 将 Simulink 模型文件及 C 代码文件统一编译生成为可执行文件,完成无人机仿真系统构建。半物理仿真结果表明:该无人机仿真系统仿真功能正确并满足实时性要求,与传统的手工代码开发方式相比,RTW 与 C 代码联合开发方式降低了系统设计难度、加速了半物理仿真平台的研制。

关键词: 无人机; 仿真系统设计; 快速原型设计; Real Time Workshop; 多任务设计

中图分类号: TP391.9 **文献标志码:** A **文章编号:** 1671-637X(2016)10-0090-06

RTW Based Design of a Real-Time and Multi-Task UAV Simulation System

CHEN Peng, LI Chun-tao

(College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China)

Abstract: In view of the high real-time and multi-task design requirements to UAV simulation system and the problems of complicated manual code development and long design cycle, a design method using RTW and C code on VxWorks platform is proposed. First, Simulink tool is used for developing the UAV simulation system. Then missions of the simulation system are divided into system administration, model solving, scheduling management and interfacing task, which are designed respectively. Then, Real Time Workshop (RTW) is used to compile the Simulink and C code files into an executable file, and thus to implement the construction of the UAV simulation system. Semi-physical simulation results show that: 1) The UAV simulation system has proper simulating function and can meet the real-time requirements; and 2) Compared with the traditional manual code development mode, the method based on RTW together with C code reduces system design complexity and accelerates the simulation platform development.

Key words: UAV; simulation system design; rapid prototyping; Real Time Workshop; multi-task design

0 引言

无人机仿真系统作为验证机载飞行控制系统的重要一环,能够充分验证飞行控制系统逻辑功能的正确性,加快无人机开发研制进程,减少试飞次数。

传统的无人机仿真系统开发采用手工代码编写,由于无人机仿真系统数学模型复杂,通常是由一系列线性、非线性微分方程组成,数据处理量巨大,需要进行大量的数据插值计算,导致手工代码编写工作量巨

大,开发周期长,不利于仿真系统的维护和升级。

RTW(Real Time Workshop)^[1-2]是 Matlab 图形建模仿真的重要组件,可将 Simulink 模型直接生成为嵌入式 C 代码,省去了数学模型至可执行文件的传统手工代码开发过程。RTW 在国内外应用广泛,美国 Lockheed Martin 公司利用 RTW 技术建立了 F-35 战机的控制律模型并直接生成为嵌入式代码^[3];NASA 将 RTW 技术应用于 X-43A 高超声速飞行器推进系统的控制律设计,并生成基于 Honeywell H-746 嵌入式处理器的实时仿真软件;国内将 RTW 技术应用于分布式仿真系统^[4]。

虽然 RTW 在无人机仿真领域应用广泛,但还存在

着一些不足。在一些复杂、庞大的无人机仿真系统中包含大量的功能模块,如果仅仅单独使用 RTW 技术,生成的整个系统往往并不高效,主要表现在:1)需将硬件驱动、外围接口程序进行 Simulink 图形化封装,再调用 RTW 生成嵌入式代码,增加了系统开发难度和复杂性;2)系统任务由 RTW 根据模块采样率生成,导致任务调度不明确,不便于系统任务管理和系统性能提升。

本文根据无人机仿真系统实时、多任务设计需求,利用 Matlab/Simulink 环境实现无人机模型的快速开发,使用 C 代码设计编写无人机仿真系统任务,应用 RTW 快速生成技术将 Simulink 模型文件及手工 C 代码文件统一编译为目标文件,完成仿真系统开发。

1 样例无人机仿真系统构成

飞行仿真环境包括飞行控制系统和无人机仿真系统两大部分(见图 1)。飞行控制系统主要实现无人机航路规划和目标导航等功能,使无人机能够按照预定轨迹飞行。无人机仿真系统模拟实际飞机运动学和动力学特性,对飞控系统的输出做出实时响应,反馈飞机当前状态信息。

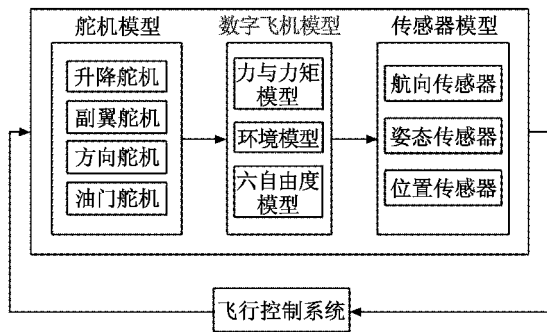


图 1 无人机仿真环境组成

Fig. 1 The composition of UAV simulation environment

无人机仿真系统包含舵机模型、数字飞机模型和传感器模型等模型。仿真系统在接收到飞行控制系统的控制信号后,驱动无人机六自由度模型进行实时解算,获取无人机飞行状态信息。这些状态信息进入到传感器模型环节,按照规定的通信协议发送给飞行控制系统。飞行控制系统根据当前飞行状态信息,输出新的控制信号,并通过舵机环节传输给数字飞机模型,完成一次完整的仿真循环。

样例无人机机载传感器和舵机数量众多,涉及大量的模型解算且均有严格的时序要求,因此仿真系统实时运行过程中任务多、解算工作量大。为减小无人机仿真系统的负荷,硬件结构采用核心板加功能板组合方式搭建(见图 2)。核心板负责无人机数字飞机模型、舵机模型以及传感器模型的实时解算,功能板卡实

现传感器和舵机环节的电气接口模拟。功能板通过 CAN 总线与核心板进行数据通信,这样就将仿真系统划分为解算功能和接口功能,实现低速外设和 CPU 快速计算能力的协同运行,有效减小系统延迟,降低 CPU 负荷。

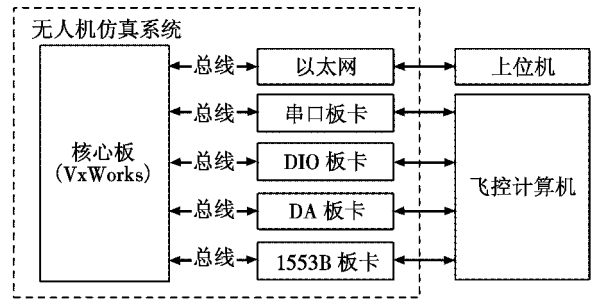


图 2 无人机仿真系统硬件组成

Fig. 2 Hardware components of UAV simulation system

2 无人机 Simulink 仿真模型建立

无人机本体模型主要由无人机六自由度模型、舵机、传感器等模型组成。无人机六自由度模型复杂,可由式(1)中十二阶非线性微分方程来描述^[5],即

$$\begin{cases} \dot{x} = f(t, x, u) \\ y = cx \end{cases} \quad (1)$$

状态量 x^T 和输入量 u^T 分别为

$$x^T = [u, v, w, \phi, \theta, \psi, p, q, r, x_e, y_e, h] \quad (2)$$

$$u^T = [\delta_r, \delta_e, \delta_a, \delta_r] \quad (3)$$

式(2)中,输入项分别表示北向速度、东向速度、天向速度、滚转角、俯仰角、偏航角、滚转角速率、俯仰角速率、偏航角速率、北向距离、东向距离、高度。式(3)中,输入项分别表示油门舵机开度、升降舵偏角、副翼舵偏角、方向舵偏角。

样例无人机的舵机模型采用典型的二环节,即

$$G(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (4)$$

式中: ω_n 为舵机的自然频率; ξ 为阻尼比。

传感器模型将样例无人机本体模型中解算出的无人机状态信息按照实际传感器帧格式进行输出。样例无人机传感器模型众多,包括惯导(INS)、速率陀螺(GYRO)、大气数据(Atmos)、GPS等。以GPS模型为例,通过无人机当前运动距离和地球半径解算出无人机当前的经纬度信息,即

$$\begin{cases} L_{lon} = L_{lon,0} + \frac{Y}{(R+H) \cos L_{lat}} \\ L_{lat} = L_{lat,0} + \frac{X}{(R+H)} \end{cases} \quad (5)$$

式中: $L_{lon,0}, L_{lat,0}$ 为无人机初始经纬度; X, Y 分别为无人

机在地理坐标系正北方向和正东方向的运动距离; H 为无人机所处的海拔高度; R 为地球半径。

Simulink 仿真环境提供了大量的航空数值仿真模块,例如六自由度模型、舵机模型、大气环境等,利用这些模块可以实现无人机仿真系统的快速搭建。

3 无人机仿真系统多任务设计

3.1 软件结构设计及任务划分

无人机仿真系统软件由系统管理层、应用层、接口服务层和操作系统层构成(见图3)。所有软件层次均运行于核心板,由接口服务层通过总线与各功能板卡交互。

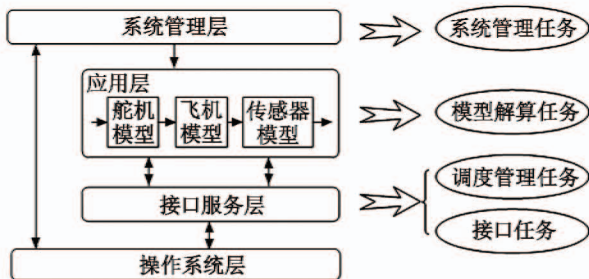


图3 仿真软件结构

Fig.3 Simulation software structure

图中:系统管理层主要实现系统初始化、时钟中断挂接等功能,并控制仿真系统的执行过程;应用层执行舵机模型、飞机模型、传感器等模型解算任务;接口服务层完成舵机、传感器等仿真对象与功能板卡的数据交互;操作系统层提供软件运行平台,负责任务调度、任务同步管理等功能。

为保证仿真系统运行的实时性能,无人机仿真系统采用 VxWorks 作为软件运行平台。RTW 提供了对 VxWorks 平台的支持,能够将 Simulink 模型直接生成成为运行于 VxWorks 的可执行程序。因此,结合 RTW 在 Simulink 模型代码生成的优势和 C 代码的灵活性,对无人机仿真系统任务进行设计,如表 1 所示。

表 1 仿真系统任务

Table 1 Simulation system task

任务名称	任务功能	设计方式
系统管理任务	系统初始化、创建系统时钟、仿真过程控制、任务创建销毁	RTW + C 代码
模型解算任务	解算 Simulink 模型任务,由系统时钟调度	RTW 创建
调度管理任务	调度数据发送任务、实现 CPU 负载均衡	C 代码
接口任务	数据收发管理、与功能板卡交互	C 代码

接口任务涉及到传感器/舵机模型的编解码操作,直接用 C 代码创建更加方便;调度管理任务用于接口任务调度,若采用 RTW 技术需要先将其封装为 Simu-

link 模型,再使用 RTW 生成,导致调度管理任务设计更加复杂,因此直接用 C 代码创建;模型解算任务主要涉及无人机六自由度动力学和运动学解算,Simulink 环境提供完善的数值仿真模块,使用 RTW 可直接生成运行于 Vxworks 环境的代码;系统管理任务涉及到系统初始化、时钟管理等操作,由 RTW 自行完成创建,但为了支持对接口任务、调度管理任务等手工创建任务的控制,需要对其进行优化调整,来满足仿真过程控制的需求。

3.2 基于 VxWorks 的实时仿真任务设计

VxWorks^[6-7]是一套具有高实时性、可裁剪实时操作系统,能够提供高精度定时时钟,并可实现基于优先级的抢占式任务调度,非常适合应用于需要高实时性、细粒度、多任务仿真的无人机仿真系统。

系统任务的调度关系如图 4 所示。

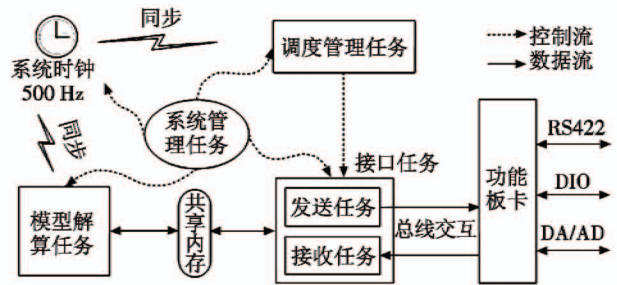


图4 任务调度关系

Fig.4 Task scheduling relationship

任务之间进行数据流和控制流交互。其中,系统管理任务处于任务控制核心,控制着系统的仿真过程,实现任务创建销毁、系统时钟挂接等功能。RTW 为系统提供基准时间信号,并通过时钟中断释放信号量实现对模型解算任务、调度管理任务的调度。

VxWorks 采用基于优先级的抢占式调度,保证优先级高的任务优先执行。模型解算任务解算频率为 500 Hz,需设置最高的优先级。接口任务模拟传感器、舵机真实的物理发送频率,设置为较低的优先级,并由高优先级的调度管理任务进行任务调度。此外,高优先级、高解算频率的模型解算任务与低优先级、低解算频率的接口任务通过共享内存方式进行数据交互。

3.2.1 系统管理任务设计

系统管理任务主要由 RTW 创建维护,入口函数为 `rt_main()`。`rt_main()`是 RTW 提供的系统主函数,实现模型初始化、系统时钟设置、中断函数挂接、任务创建、错误检查、内存管理等功能。

通过修改 `rt_main()` 函数可实现对 C 代码创建任务的初始化、创建、销毁等执行过程控制,达到 RTW 与 C 代码结合的设计目的。此外,为实现仿真开始、仿真结束等仿真过程控制,在 `rt_main()` 函数内部添加同步

信号量 `simStart` 和 `simEnd`, 通过对 `simStart` 和 `simEnd` 信号量的释放, 实现仿真过程控制。系统管理任务伪代码设计过程如下所示。

```

Int_T rt_main( <形参> )
{
0: <其他代码>
1: 错误检查、挂接时钟中断等;
2: semTake(simStart); //同步信号量
3: 模型初始化; //RTW 自动创建
4: 模型解算任务创建; //RTW 自动创建
5: 调度管理任务创建; //人工创建
6: 接口任务创建; //人工创建
7: <其他代码>
8: semTake(simEnd); //同步信号量
9: 模型解算任务销毁; //RTW 自动创建
10: 调度管理任务销毁; //人工创建
11: 接口任务销毁; //人工创建
12: <其他代码>
13: 系统管理任务自启 //人工创建
}

```

系统管理任务是事件触发任务, 上位机触发仿真开始指令并释放 `simStart` 信号量, 系统管理任务获取 `simStart` 信号量后挂接系统时钟、创建模型解算任务、接口任务等仿真任务, 系统进入图 4 所示正常的仿真状态, 此时, 系统管理任务执行到第 8 行因等待 `simEnd` 信号量进入阻塞状态。

当上位机发出仿真结束指令后, 系统管理任务获得 `simEnd` 信号量并退出阻塞状态, 销毁模型解算任务、接口任务等所有任务, 并释放内存。

为实现仿真结束后系统能够进行再次仿真, 在 `rt_main()` 函数末尾添加系统管理任务自启代码。仿真结束后重启系统管理任务, 重新初始化系统并进入等待 `simStart` 信号量阶段, 实现仿真开始、仿真结束状态的平滑切换。

3.2.2 模型解算任务设计

模型解算任务处于系统结构中的应用层, 是由 RTW 创建生成的 Simulink 仿真模型解算任务。RTW 根据无人机 Simulink 模型自动创建任务、挂接系统时钟、执行任务调度。模型解算任务的设计只需要解决该任务与接口任务之间的数据交互问题。VxWorks 系统下任务之间的数据交互有消息队列、管道、共享内存等方式, 而共享内存实施简单方便, 通过修改无人机仿真系统的 Simulink 模型, 使其对共享内存进行读写即可实现与接口任务之间的数据交互。

Matlab 提供了 Stateflow^[8] 机制实现 Simulink 环境下的仿真数据和 C 代码之间的数据交互。Stateflow 是

Matlab 提供的基于有限状态机和流程图的建模和仿真环境, 在 Stateflow 可以以子函数调用方式实现用户代码向仿真系统的嵌入, 同时 Stateflow 建立的模型也可以由 RTW 自动转化为 C 代码。

图 5 所示为模型解算任务数据流向, 通过向 Simulink/Stateflow 嵌入读取共享内存的用户代码, 实现模型解算任务和接口任务之间的数据交互。

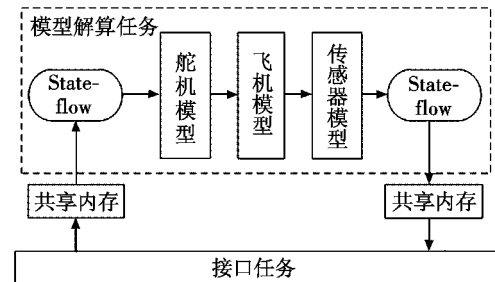


图 5 模型解算任务执行流程

Fig. 5 Model solving task execution process

共享内存在使用时会产生因读写任务竞争而导致的数据一致性问题。为解决该问题, 可采用双缓冲区方式同时创建读内存和写内存, 读任务只能读取读内存中的数据, 而写任务只能将数据写入写内存。写任务在完成一次数据写入后将读写内存置换, 即读内存成为写内存, 写内存成为读内存。通过内存置换使读任务能够读取写任务写入的最新数据。

双缓冲区方式将共享内存的读写分割在不同的物理内存上进行, 避免了因读写冲突导致的数据一致性问题。

3.2.3 接口和调度管理任务设计

接口任务负责仿真系统与外界系统的数据交互, 包括发送任务和接收任务两大部分。数据发送任务按照实际帧格式进行组帧, 再通过总线发往功能板卡, 由功能板卡模拟电气接口并发送数据。为模拟真实传感器的信号发送频率, 以及避免过多传感器数据同时发送而产生瞬时负载过大问题, 数据发送任务由调度管理任务进行统一调度。

调度管理任务首先获取系统时钟中断(中断频率 500 Hz)并进行中断 tick 数累加, 累加和分别对 5, 10, 20 等进行求余计算, 根据余数释放相应信号量, 数据发送任务获取信号量后执行一次数据发送, 从而实现 100 Hz, 50 Hz, 25 Hz 等频率的任务调度。

数据接收任务完成外界仿真数据的接收, 并将其传输至模型解算任务, 该任务通过系统管理任务创建, 通过总线数据中断触发进行任务调度, 其任务流程如图 6 所示。

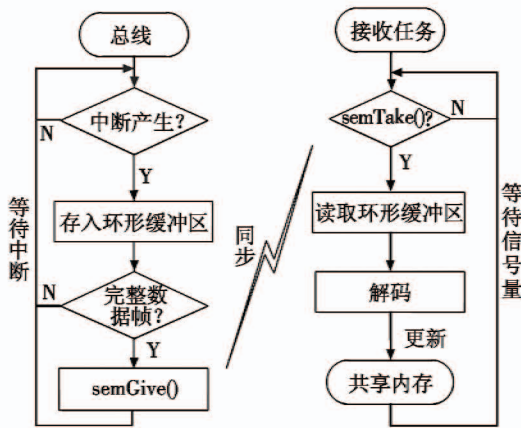


图6 接收任务执行流程

Fig. 6 Execution process for task receiving

首先将中断数据存入环形缓冲区,并在每次中断检查数据帧完整性。接收完整后释放信号量给数据接收任务,数据接收任务获得信号量后读取缓冲区并进行数据解码,最后将数据存入全局变量供模型解算任务读取。

3.2.4 实时代码生成

接口任务等C代码文件需加入到RTW中进行联合编译。编译方式通常有两种:1)将RTW生成的代码文件和手工代码文件移植到第三方编译环境,进行联合编译,生成可执行文件^[9];2)通过修改RTW提供的模板联编文件(tornado.tmf),添加手工代码文件路径,并指导第三方编译器进行编译链接,直接生成为目标文件^[10]。

方式2)相比方式1)更加快速高效,实现从模型到可执行代码的直接生成。RTW首先通过RTW Build将模型文件转化为model.rtw中间文件,该文件包含了模型内部各个模块的输入输出信息、状态信息等信息汇总。RTW Build同时将定制的tornado.tmf文件生成Makefile文件:model.mk。接着,RTW调用目标语言编译器tornado.tlc将中间文件编译为model.c,model.h等一系列模型代码文件,最后,在Makefile文件的编译下将生成的模型文件、用户代码文件统一编译链接为可执行文件model.out。

4 仿真与验证

该部分主要验证系统的实时性和功能的正确性。其中:实时性测试主要验证系统任务是否实时可调度;功能测试主要验证仿真系统逻辑功能正确性。

4.1 实时性验证

Tornado开发环境提供了一个专门用于分析任务状态的工具WindView,它以图形化方式详细记录了任

务调度周期、CPU占用量等信息。通过WindView工具对仿真系统任务做出如表2所示的统计信息。

表2 任务执行时间测试结果

Table 2 Test results of task execution time

任务名	频率/Hz	执行时间/ μ s	CPU 占用/%	任务类别
my_rt_main	触发	2300		系统管理任务
tsingleRate	500	890.1	45.00	模型解算任务
taskManage	500	20.4	1.02	调度管理任务
INS_FCC	50	31.5	0.16	接口任务
GPS_FCC	20	36.1	0.072	接口任务
Atmos_FCC	20	32.4	0.065	接口任务
Gyro_FCC	50	21.2	0.106	接口任务
FCC_Servo	50	8.7	0.044	接口任务
FCC_INS	25	16.1	0.040	接口任务

表2中:my_rt_main为触发任务,根据系统仿真开始、仿真结束等指令执行系统启动、销毁等过程操作,因此执行时间较长;tsingleRate是系统的模型解算任务,为仿真主体任务并且占用较长解算时间,约占CPU利用率的45%;调度管理任务和接口任务执行时间普遍较短。为验证系统的可调度性,将表2测试结果代入式(6)所示的可调度任务利用率算式,计算出系统平均可调度利用率为0.46,小于最大可调度利用率0.72($N=8$),说明任务是可调度的,满足实时性要求。

$$U = \sum_{i=1}^N \frac{t_i}{T_i} \leq N(2^{1/N} - 1) \quad (6)$$

式中: U 为可调度利用率; N 为任务个数; t_i 为各任务的运行时间; T_i 为各任务的运行周期; $N(2^{1/N} - 1)$ 为理论最大可调度利用率。

其次,Tornado提供了Spy命令动态监测系统CPU利用率,获得表3所示的CPU利用率的统计结果。从测试结果发现系统CPU空闲资源充足,并且负载均衡。

表3 CPU利用率测试结果

Table 3 Test results of CPU utilization %

类别	最大利用率	最小利用率	平均利用率
CPU 利用率	50.4	44.4	46.0

4.2 功能性验证

为验证仿真系统功能的正确性,在相同初始条件下,将半物理仿真环境下的仿真数据与Matlab/Simulink数字仿真数据进行对比,以验证Matlab模型与RTW生成代码程序的等效性。

选取仿真过程中起飞拉起阶段的高度信号、俯仰角信号和V尾舵机偏转信号作为观测对象,测得如图7所示的对比曲线(设定高度1000m),从中可以发现,仿真数据和半物理数据一致性良好,验证了仿真功能的正确性。

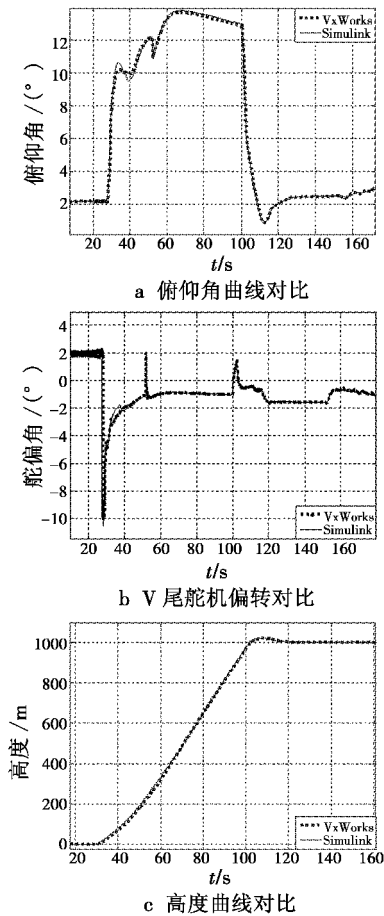


图 7 测试结果

Fig. 7 Test results

5 结论

RTW 提供了多种灵活多变的 Matlab 与 C 代码结合的方式,使开发人员可以方便快速地进行仿真任务设计,进而实现仿真系统的构建。综合利用 RTW 技术的快速性与 C 代码的灵活性进行联合开发,缩短了无人机仿真系统的研制周期,大大减小了仿真人员的开发难度和工作量,并且利于后期维护升级。

参考文献

- [1] 任传俊,蒋志文. Real-Time Workshop 实时仿真研究与应用[J]. 计算机仿真,2007, 24(8):268-271. (REN C J, JIANG Z W. Research and application of real-time simulation on Real-Time Workshop[J]. Computer Simulation, 2007, 24(8):268-271.)
- [2] 马慧敏,章慧君. 导弹制导控制系统快速原型设计技术[J]. 电光与控制,2012, 19(5):78-81. (MA H M, ZHANG

H J. Rapid prototyping technologies for missile guidance and control system [J]. Electronics Optics & Control, 2012, 19(5):78-81.)

- [3] NIXON D W. Flight control law development for the F-35 joint strike fighter[Z]. 2004.
- [4] 冯磊. 基于 RTW 的分布式实时仿真系统研究与实现[D]. 长沙:国防科学技术大学,2005. (FENG L. Research and implementation of distributed real time simulation system based on RTW[D]. Changsha: National University of Defense Technology, 2005.)
- [5] 李军伟,袁冬莉. 基于 VxWorks 的无人机半物理仿真研究[J]. 测控技术,2008, 27(9):92-94. (LI J W, YUAN D L. Research on UAV semi-physical simulation based on VxWorks[J]. Measurement & Control Technology, 2008, 27(9):92-94.)
- [6] 尹超. 基于 MATLAB/RTW 和 Vxworks 的飞行模拟器实时仿真研究[J]. 软件导刊,2010, 9(12):86-87. (YIN C. Research on real-time flight simulator based on MATLAB/RTW and Vxworks[J]. Software Guide, 2010, 9(12):86-87.)
- [7] 李立哲,吕伟,樊丁. 基于 VxWorks 的航空发动机控制系统软件设计[J]. 电光与控制,2010, 17(9):86-89. (LI L Z, LYU W, FAN D. Design of the aero-engine control system software based on VxWorks[J]. Electronics Optics & Control, 2010, 17(9):86-89.)
- [8] 施嵘. Simulink/Stateflow 仿真原理和实现的研究[D]. 成都:电子科技大学,2011. (SHI R. Research on the principle and implementation of Simulink/Stateflow[D]. Chengdu: University of Electronic Science and Technology, 2011.)
- [9] 刘婕. Matlab/Simulink 与 VC++ 混合编程的实现[J]. 科学技术与工程,2012, 12(16):4005-4007. (LIU J. Implementation of hybrid programming in Matlab/ Simulink and VC++ [J]. Science Technology and Engineering, 2012, 12(16):4005-4007.)
- [10] 齐振恒,孙中杰,李涛. RTW 嵌入式代码自动生成机制与代码结构分析[J]. 计算机测量与控制,2010, 18(3):639-642. (QI Z H, SUN Z J, LI T. RTW embedded code auto generation mechanism and code structure analysis [J]. Computer Measurement & Control, 2010, 18(3):639-642.)