doi: 10.7690/bgzdh.2018.11.009

基于量子框架和 Stateflow 模型的嵌入式系统软件设计

刘芮滦，邓　杨，史伟娜，刘　志
（中国工程物理研究院电子工程研究所，四川 绵阳 621000）

摘要：基于模型的设计是目前嵌入式系统软件设计的发展趋势，对嵌入式系统建模和根据模型自动生成代码是其关键技术。量子框架作为一种事件驱动型的基础框架，可以作为嵌入式软件运行的支撑平台。Stateflow 模型适用于描述嵌入式系统的逻辑控制功能，利用 RTW 工具可以直接从该模型自动生成 C 代码。以某飞行控制系统为应用实例设计其活动对象和事件，针对时序控制功能建立 Stateflow 模型并进行仿真，最后自动生成 C 代码与量子框架集成，从而实现飞行控制系统的软件设计。研究表明：量子框架较好地支持了 Stateflow 模型自动生成的代码，两者结合可以实现基于模型的设计在嵌入式系统软件设计中的应用。

关键词：嵌入式系统；基于模型的设计；量子框架；Stateflow；自动代码生成
中图分类号：TP391.9　　**文献标志码：**A

Embedded System Software Design Based on Quantum Frame and Stateflow Model

Liu Ruiluan, Deng Yang, Shi Weina, Liu Zhi
(*Institute of Electronic Engineering*, *China Academy of Engineering Physics*, *Mianyang* 621000, *China*)

Abstract: Model-based design is nowadays the development trend of the embedded system software design, and modeling the embedded system and auto code generation by models are the key technologies. Quantum framework, as an event driven framework, can be used as a supporting platform for embedded software. Stateflow model is suitable for describing the logic control function of embedded system, and the C code can be automatically generated from the model directly by using the RTW tool. In this paper, a flight control system is used as an example to design the active objects and events, and the Stateflow model of the sequential control function is established and simulated, finally the C code is automatically generated and integrated with the quantum framework. In this way, the software of flight control system is designed. The research shows that the quantum framework can support the codes generated by Stateflow model automatically, and the combination of them can realize the application of model-based design in embedded system software design.

Keywords: embedded system; model-based design; quantum frame; Stateflow; auto code generation

0　引言

嵌入式系统大量应用于航空航天、国防军工等安全关键领域，随着嵌入式系统软件规模和复杂度的急速增长，嵌入式系统软件设计、开发及验证面临新的挑战。基于模型的设计正在逐步替代传统的基于文档的软件开发方式。通过体系结构建模、需求建模，软件以动态可仿真运行的模型呈现并贯穿于从需求、设计到验证的整个软件开发周期；通过软件代码自动生成，保证整个嵌入式系统软件开发过程中的一致性和自动化[1]。

量子框架（quantum frame，QF）是一个事件驱动型的基础软件框架，可以作为一种轻量级的软总线连接多个活动对象（active object，AO）和硬件中断服务程序[2]。活动对象之间通过事件进行通信。Stateflow 是一个基于有限状态机的图形化建模、仿真环境，构建在 Matlab/Simulink 之上，适用于对嵌入式系统的控制逻辑进行建模和仿真[3]，同时支持时间驱动和事件驱动，利用 RTW 工具可以直接从模型自动生成 C 代码。对嵌入式系统而言，量子框架可以作为软件的支撑平台，活动对象直接调用 Stateflow 模型生成的 C 代码，由框架提供线程、事件队列、事件发送、定时等服务，从而实现基于模型的设计在嵌入式系统软件开发中的应用。

1　量子框架

量子框架由美国量子公司的 Miro Samek 博士提出，是一种基于事件驱动并发状态机的基础软件框架，基于该框架的嵌入式应用软件体系结构如图 1 所示。量子框架和量子事件处理器（quantum event processor，QEP）作为应用程序（Application）、底层硬件（Hardware）和操作系统（real-time operating

收稿日期：2018-08-18；**修回日期：**2018-09-06
基金项目：中国工程物理研究院质量与可靠性技术基础课题（S2016ZK.1）
作者简介：刘芮滦（1987—），男，四川人，工学硕士，助理工程师，从事嵌入式系统建模与软件设计研究。

system，RTOS)之间的中间件将其隔离，应用层和底层之间的信息交互主要通过量子框架传递。量子侦察器(quantum scout，QS)是一个实时轨迹追踪工具，主要提供测试功能[4]。

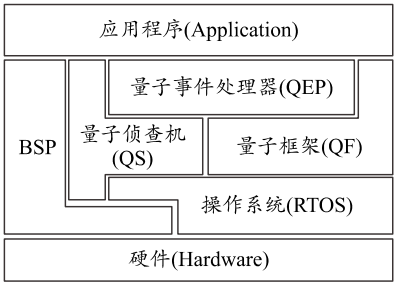

图 1　基于量子框架的应用软件体系结构

活动对象和事件是量子框架的主要元素。活动对象的本质是一个状态机，被封装成一个具有单独事件队列和执行线程的实现对象，同样，事件也被封装成对象[5-6]。它们的类图如图 2 所示。活动对象之间不共享任何数据和资源，只通过量子框架订阅和发布不同的事件实例进行交换，由框架负责事件的管理和分发，从而实现了活动对象之间的异步通信，降低了彼此间的耦合性。

活动对象包含有限层次状态机、事件队列和执行线程 3 个基本要素。有限层次状态机可以采用工具进行建模，能够清晰地表示业务逻辑。活动对象拥有自己的运行线程，支持与其他活动对象并发执行。采用事件队列和状态机中的“运行到完成”(run to complete，RTC)语义实现原子事件处理，保证了活动对象每次只处理一个事件，有效地避免了死锁问题[7]。在利用量子框架进行嵌入式系统软件设计时，只需将各个功能模块划分为不同的活动对象，再定义好各个活动对象之间通信的事件即可。

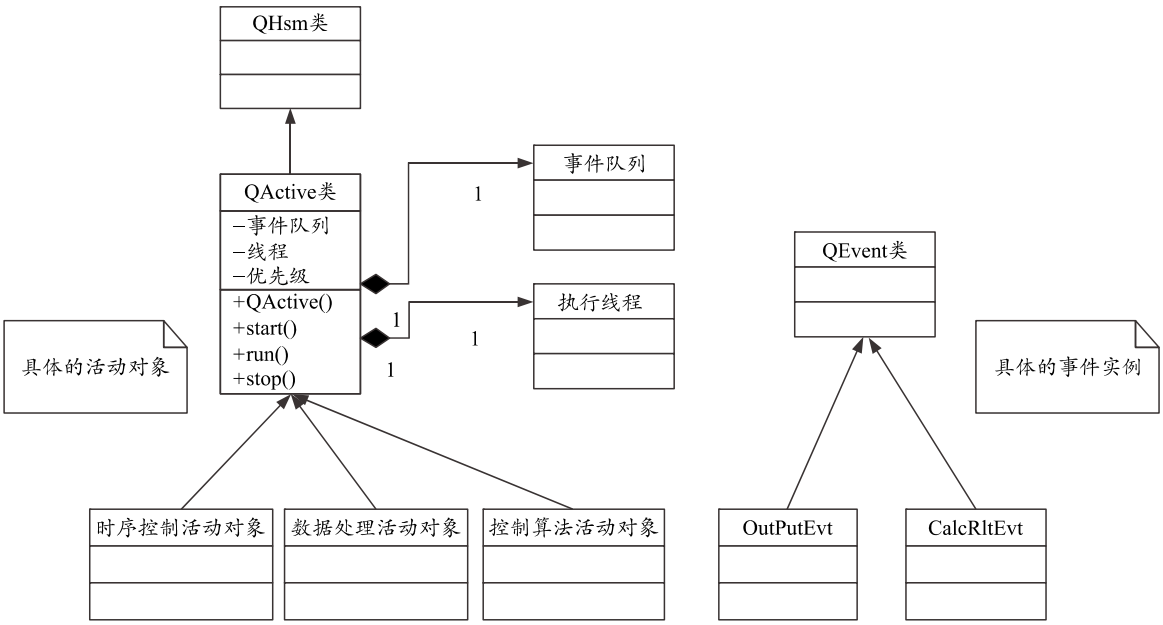

图 2　活动对象和事件的类图

2　Stateflow 模型

有限状态机是表示有限个状态以及在这些状态之间的转移和动作等行为的数学模型，是许多形式化规约、验证方法的基础模型。Stateflow[8]是有限状态机的图形化设计工具，使用层次化、可并行、有明确执行语义的元素描述复杂的逻辑系统，通过状态流程和事件驱动实现对离散事件系统的仿真。Stateflow 对事件驱动系统进行建模，状态表示系统的模态，分为激活态与非激活态，状态激活由事件驱动。Stateflow 状态图使用一条单向箭头曲线表示状态的迁移。事件在 Stateflow 中是最基本的行动元素，主要分为输入事件、本地事件和输出事件。

Real-Time Workshop(RTW)代码生成工具可以从 Stateflow 创建的模型生成高效、专业的面向特定控制器的 C 语言代码。目标语言编译器(target language compiler, TLC)是 RTW 代码生成工具的一个组成部分，合理地编写或修改 TLC 文件，可以定制生成的代码。RTW 和 TLC 协同工作，实现代码的自动生成。代码自动生成时，首先 RTW 将 Stateflow 模型生成对应的 RTW 文件，该文件包括了生成代码所需要的信息；然后 TLC 编译器读取 RTW 文件，据此选择系统 TLC 和模块 TLC 文件，进而生成 C 源代码、头文件、MK 文件(用于生成可执行文件)等[9-11]。

3　飞行控制系统软件设计实例

3.1　活动对象和事件定义

笔者以某飞行控制系统作为应用实例，设计嵌入式软件的活动对象并定义相关事件。该飞行控制系统的处理器为 ARM Cortex-M4，采用标准 CAN 总线用于内部设备通信。控制系统的主控单元通过 CAN 总线连接多个设备和传感器，通过传感器Ⅰ、Ⅱ、Ⅲ分别采集加速度数据、速度数据和高度数据。根据采集的传感器数据进行姿态识别、时序控制和飞行控制算法计算，并给出相应的控制输出。针对该飞行控制系统共设计 10 个活动对象，并为每个活动对象分配了唯一的优先级。活动对象和中断服务程序之间的事件交互关系如图 3 所示。

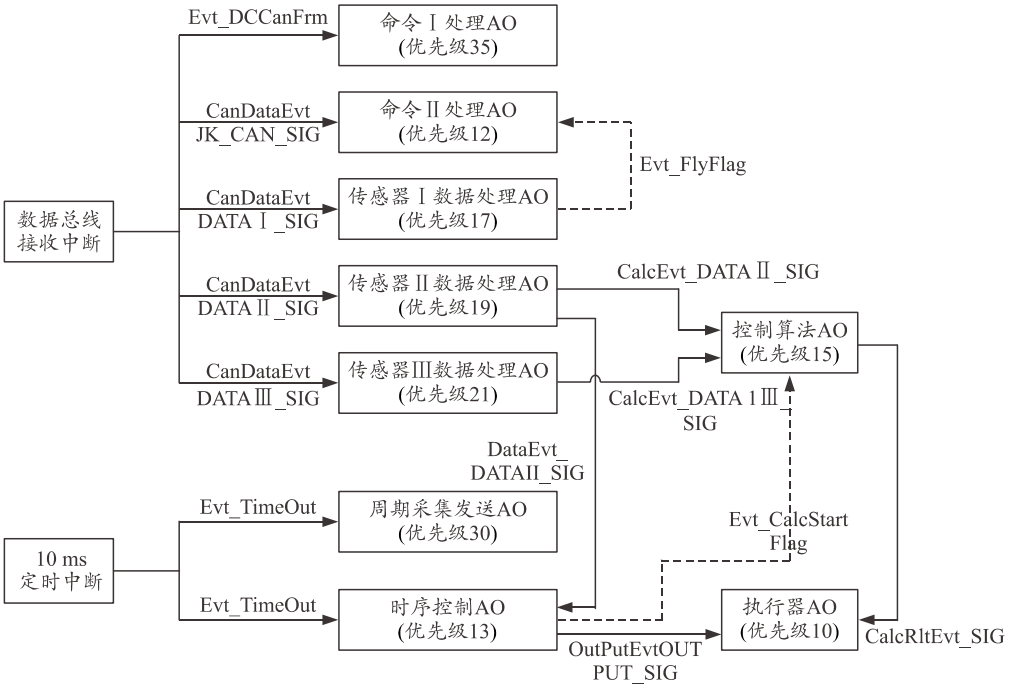

图 3　活动对象事件交互关系

在 CAN 接收中断处理程序中，根据 CANID 号将不同节点设备的数据通过事件发送给相应数据或命令处理活动对象。10 ms 定时中断处理程序发送超时事件给周期数据采集发送活动对象和时序控制活动对象，为其提供时间驱动。命令处理活动对象响应外部设备命令，如参数装订和回读。传感器Ⅰ数据处理活动对象主要处理加速度数据并对飞行姿态进行识别；时序控制活动对象主要对飞行过程进行时序逻辑控制；控制算法活动对象主要是依据传感器Ⅱ数据处理活动对象和传感器Ⅲ数据处理活动对象解析处理的速度数据和高度数据，进行控制算法计算。时序控制活动对象的控制命令和控制算法活动对象的计算结果通过事件传输给执行器活动对象，执行相关输出操作。

3.2　时序控制建模和仿真

针对时序控制活动对象，建立与业务逻辑相对应的模型。分析其详细功能，发现时序控制主要由多个独立的状态组成，通过对各种外部激励或时间激励进行决策，实现不同状态的顺序转换。整个工作过程具有离散性的特征，决定了该活动对象可以基于有限状态机理论进行建模。根据时序控制活动对象的功能需求建立相应的 Simulink/Stateflow 模型，如图 4 所示。在模型中：DataBuild 模块从数据文件中获取加速度数据、速度数据和高度数据供 Stateflow 模型使用；Parameters 模块提供控制参数，如延时时间、期望高度等；SignalBuild 模块通过 ManualSwitch 手动切换产生输入信号(输入信号从 0 变为 1)；PulseGenerator 脉冲发生器模块用于产生时钟周期(0.01 s，上升沿有效)；Scope 示波器显示输入数据和输出的时序状态；Chart 则为时序控制的 Stateflow 模型(SequenceControl)，用于实现飞行过程的控制逻辑。SequenceControl 模型中包含时序控制中的各个子状态，每个子状态对应一个标志，当标志置 1 时，时序控制活动对象向执行器活动对象发送事件，通知其执行相应的输出操作。

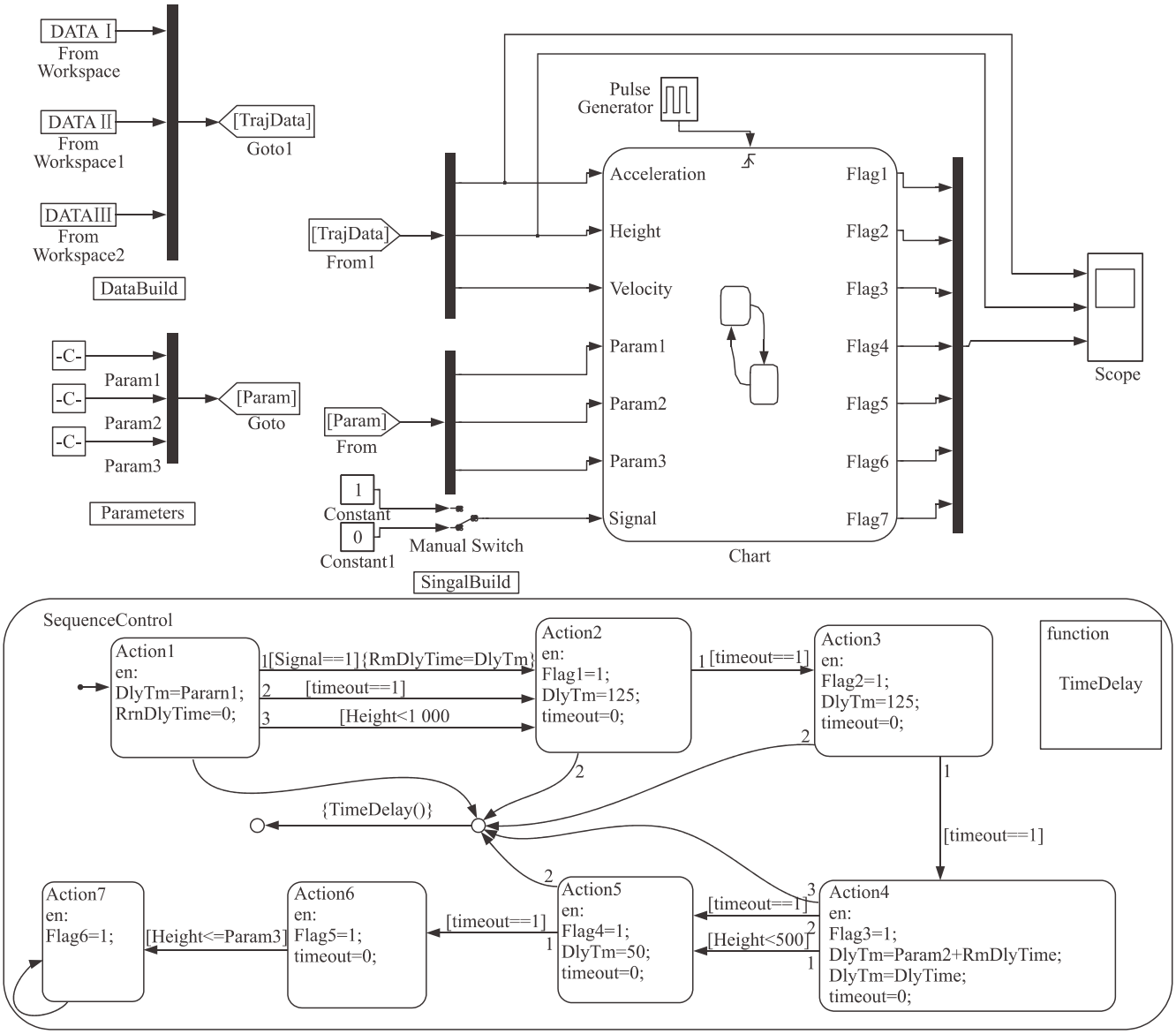

图 4　时序控制 Simulink/Stateflow 模型

Simulink/Stateflow 模型仿真时，首先运行 LoadData.m 脚本加载数据文件，读取加速度数据、速度数据和高度数据，为每个数据增加一列时间标志，时间标志的步长为 0.01 s，然后通过 From Workspace 模块从工作空间导入数据。From Workspace 模块调用工作空间的数据时遵循矩形格式，而且输入矩阵的列数必须比输入信号多一列，Simulink 会自动把首列数据当做时间向量。在 Simulink 中用 Annotation 方式定义变量，可以在 Parameters 模块中设置控制参数。对于离散系统来说，由于脉冲发生器的周期为 0.01 s，因此延时时间需要除以 0.01 转换成整型数据。在 Simulation→Configuration Parameters 设置模型仿真参数，如仿真时间、求解器步长。设置仿真时间的起止时间为 0 和 85 s。步长是求解器运算时的时间精度，步长越短，精度越高，运算量也更大。选择定步长求解器(fixed-step)，步长设置为 0.01 s。运行模型后可得到仿真结果，如图 5 所示。从图中可以看出，模型仿真运行结果和软件飞行过程中控制时序相吻合，可以通过该模型自动生成 C 代码。

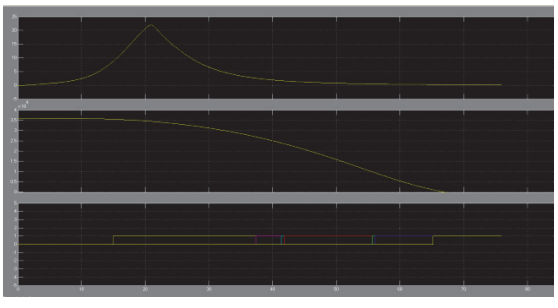

图 5　Stateflow 模型仿真结果

3.3 自动生成代码与框架集成

在模型主窗口选择菜单项 Simulation→Configuration Parameters…，在 Solver 面板中设置求解器为定步长离散求解器，步长为 0.01 s，在 Real-Time Workshop 面板设置 TLC 文件为 ert.tlc，完成设置后使用 RTW 工具生成代码和代码报告，总共生成 6 个头文件和源文件，包含 ert_main.c、SeqControl.c、SeqControl.h、SeqControl_private.h、SeqControltypes.h、rtwtypes.h。文件 SeqControl.c 包含了模型函数 SeqCtl_initialize、SeqCtl_step、SeqCtl_terminate 的入口地址及其代码。函数 SeqCtl_step 实现了模型中所有模块的功能，是最重要的函数。SeqCtl_step 函数的默认参数和返回值都为 void，属于不可重入函数，用户可以重新定义 SeqCtl_step 函数原型，自定义其参数和返回值类型。

时序控制活动对象对应的源文件为 Active_SeqCtl.c，在该文件中定义时序控制活动对象的输入、输出事件，此外还定义了 SeqCtl_OneStep 函数，在该函数中添加 SeqCtl 模型的输入和输出并调用 SeqCtl_step 函数，使得 SequenceControl 模型得到执行。时序控制活动对象的部分代码如下：

```
/*定义时序控制活动对象*/
QSTATE SeqCtrl(QEvent const *e){
switch(e->sig){
case TIMEOUT_SIG:
 /*以 10ms 为步长执行一次时序控制*/
    SeqCtl_OneStep();
 break;
case HEIGHT_DATA_SIG:
/*收到一帧高度数据后执行一次时序控制*/
    HeightDataEvt *ev = (HeightDataEvt* )e;
    s_fHeight = ev-> fHeighData;
    SeqCtl_OneStep();
break;
……
}
}
/*定义 SeqCtl_OneStep 函数，添加输入和输出*/
void SeqCtl_OneStep(void){
/*设置模型的输入数据*/
      SeqCtl_U.Height = s_fHeight;
      /*执行模型*/
      SeqCtl_step();
/*模型输出标志 1 置位后向执行器活动对象发送加电事件*/
      if((SeqCtl_Y.Flag1 == 1) {
Qevent *ev = Q_NEW(OutPutEvt,
OUTPUT_SIG);
   ev->OutputType = POWERON;
        QF::publish(ev);
      }
}
```

RTW 工具自动生成的代码结构清晰，接口明确，非常方便与量子框架集成。将自动生成的源代码和头文件拷到 IAR 工程中，和量子框架相关代码直接编译，即可生成可执行文件。程序在硬件平台运行稳定，试验结果和仿真结果一致，能够满足时序控制的功能需求。

4 结束语

量子框架为嵌入式系统软件提供基础支撑和服务，Stateflow 对嵌入式系统的业务逻辑进行建模。笔者基于量子框架和 Stateflow 模型，结合代码自动生成技术，完成了飞行控制软件时序控制功能的设计。实际应用结果表明：量子框架较好支持了 Stateflow 模型自动生成的代码，两者结合可以实现基于模型的设计在嵌入式系统软件设计中的工程应用，具有重要的研究价值和实践意义。

参考文献：

[1] 刘兴华，曹云峰. 一种模型驱动的嵌入式控制软件设计技术研究[J]. 系统仿真学报，2013，25(7)：106-110，143.

[2] 董伯麟，鞠毅. 基于量子框架的六自由度机器人控制系统研究[J]. 组合机床与自动化加工技术，2015，12(12)：85-89.

[3] 邹晖，陈万春，殷兴良. Stateflow 在巡航导弹仿真中的应用[J]. 系统仿真学报，2004，16(8)：1854-1860.

[4] 杨泉. 基于量子框架的汽车电控系统软件总线研究[D]. 济南：山东大学，2008.

[5] 李晓军，张承瑞. 基于量子框架的重卡 AMT 系统逻辑建模研究[J]. 系统仿真学报，2009，21(13)：3855-3859.

[6] 李洪斌，张承瑞. 基于量子框架的开放式汽车电控系统体系结构[J]. 吉林大学学报，2006，36(2)：166-171.

[7] 赵刚. 面向嵌入式系统中量子框架的状态机代码自动生成技术的研究与实现[D]. 西安：西安电子科技大学，2012.

[8] 崔松，韩裕生，朱守中. 基于 Stateflow/Simulink 的末制导系统仿真[J]. 兵工自动化，2009，28(5)：53-54.

[9] 刘杰，翁公羽，周宇博. 基于模型的设计——MCU 篇[M]. 北京：北京航空航天大学出版社，2011：245-250.

[10] 孙忠潇. Simulink 仿真及代码生成技术入门到精通[M]. 北京：北京航空航天大学出版社，2015：369-374.

[11] 路引，陈睿璟，王道波. 某型无人机飞行动力学仿真软件设计[J]. 兵工自动化，2016，35(2)：94-96.