www.scichina.com info.scichina.com

SCIENTIA SINICA Informationis

中国科学 信息科学

第 45 卷 第 3 期 2015 年 3 月 CN 11-5846/TP ISSN 1674-7267 eISSN 2095-9486

中国科学院
国家自然科学基金委员会 主办

国家自然科学基金资助

中国科学 信息科学
SCIENTIA SINICA Informationis
(ZHONGGUO KEXUE XINXI KEXUE)

中国科学院科学出版基金资助

第 45 卷 第 3 期 2015 年 3 月

目 次

评述

电子装备机电耦合研究的现状与发展 …… 299

段宝岩

网格模型上的离散测地线 …… 313

赵俊莉，辛士庆，刘永进，王醒策，武仲科，周明全，贺英

论文

客观信息的模型和度量研究 …… 336

许建峰，汤俊，马雪峰，徐斌，沈艳丽，乔永杰

相控阵雷达同时多波束 SAR 成像模式 …… 354

温晓杨，匡纲要，胡杰民，张军

基于 MAPC-RISR 的 MIMO 雷达距离—角度二维超分辨率成像算法 …… 372

王伟，马跃华，郝燕玲

数值模拟领域并行编程模型的要素与实例研究 …… 385

莫则尧，张爱清，刘青凯，曹小林

面向临床心电图分析的深层学习算法 …… 398

金林鹏，董军

中国高速铁路列控系统的形式化分析与验证 …… 417

郭丹青，吕继东，王淑灵，唐涛，詹乃军，周达天，邹亮

帕金森状态的慢变量反馈模糊控制 …… 439

刘晨，王江，邓斌，魏熙乐，于海涛，李会艳

Volume 45 Number 3 March 2015

Contents

REVIEW

Review of electromechanical coupling of electronic equipment ····· 299

DUAN BaoYan

A survey on the computing of geodesic distances on meshes ····· 313

ZHAO JunLi, XIN ShiQing, LIU YongJin, WANG XingCe, WU ZhongKe, ZHOU MingQuan & HE Ying

RESEARCH PAPER

Research on metrics and models for objective information ····· 336

XU JianFeng, TANG Jun, MA XueFeng, XU Bin, SHEN YanLi & QIAO YongJie

Simultaneous multi-beam SAR imaging mode using phased array radar ····· 354

WEN XiaoYang, KUANG GangYao, HU JieMin & ZHANG Jun

High-resolution MIMO radar range-angle 2D imaging algorithm based on MAPC-RISR ····· 372

WANG Wei, MA YueHua & HAO YanLing

Research on the components and practices for domain-specific parallel programming models for numerical simulation ····· 385

MO ZeYao, ZHANG AiQing, LIU QingKai & CAO XiaoLin

Deep learning research on clinical electrocardiogram analysis ····· 398

JIN LinPeng & DONG Jun

Formal analysis and verification of Chinese train control system ····· 417

GUO DanQing, LÜ JiDong, WANG ShuLing, TANG Tao, ZHAN NaiJun, ZHOU DaTian & ZOU Liang

Closed-loop fuzzy control of Parkinsonian state based on slow variable ····· 439

LIU Chen, WANG Jiang, DENG Bin, WEI XiLe, YU HaiTao & LI HuiYan

中国科学: 信息科学 2015 年 第 45 卷 第 3 期: 417–438
www.scichina.com info.scichina.com
《中国科学》杂志社
SCIENCE CHINA PRESS

论 文

中国高速铁路列控系统的形式化分析与验证

郭丹青[①], 吕继东[②], 王淑灵[①], 唐涛[③], 詹乃军[①*], 周达天[②], 邹亮[①]

① 中国科学院软件研究所计算机科学国家重点实验室, 北京 100190
② 北京交通大学轨道交通运行控制国家工程中心, 北京 100190
③ 北京交通大学轨道交通控制与安全国家重点实验室, 北京 100190
* 通信作者. E-mail: znj@ios.ac.cn

收稿日期: 2014–01–21; 接受日期: 2014–05–08; 网络出版日期: 2015–01–21
国家重点基础研究发展计划 (973)(批准号: 2014CB340700)、国家高技术研究发展计划 (863)(批准号: 2012AA112801)、国家自然科学基金 (批准号: 91118007, 61100061, 61304185) 和中国科学院国家外专局创新国际合作伙伴计划资助项目

摘要 高速铁路列控系统的安全与否直接涉及人民的生命财产安全, 对高速铁路列控系统进行严格的形式化验证具有重要意义. 但是随着高速铁路列控系统软件以及硬件规模的不断增大, 系统的复杂性有了很大的提高, 直接对高速铁路列控系统进行形式化验证已经变得越来越困难. 另一方面, 由于图形化建模和仿真表现方式直观且易于理解, 在工程实践中已经得到了广泛的应用. 因此, 为了更好地保证铁路系统的安全, 对系统进行仿真, 排除部分安全隐患显得尤为重要. 本文通过使用 Simulink/Stateflow 建模工具对高速铁路列控系统的行车许可, 等级升级及部分模式转换场景进行了建模. 该模型具有普适性, 通过修改参数信息, 可以对不同的等级转换和模式转换的组合情况进行仿真. 本文使用该模型对 10 种组合情况进行了仿真, 发现在某些情况下可能会出现不正常停车或者等级转换失败的现象. 类似于测试, 仿真仅仅能够发现错误, 如未发现错误, 也不能证明系统是正确的. 因为仿真的这种不完备性, 对仿真辅助形式验证在安全攸关系统设计中非常必要. 为此, 取其中一个不正常停车的场景进行了形式验证, 验证结果证明在任何情况下都不能正常停车.

关键词 中国高速铁路列控系统 Simulink/Stateflow 仿真 模式转换 等级转换 形式化验证

1 引言

高速铁路具有节能、环保、大运量、安全舒适等明显优势, 是交通运输体系中最具可持续性和环境友好性的运输模式, 客运高速化已成为世界潮流. 我国高速铁路建设列为拉动内需的 “火车头”, 重点发展高速铁路, 将建成以北京为中心, 2 小时大中城市圈为主节点, 8 小时快速铁路交通为主干的高速铁路客运网络. 可以预见, 以高速铁路为核心的快速铁路交通网的建成, 将使我国经济社会发展步入高速铁路时代. 高速铁路是庞大复杂的系统工程, 集成了多学科、多领域的高新技术. 其中, 中国高速铁路列控系统是保证列车高速、高效、安全运行的坚实基础, 所以确保高速铁路列控系统的安全性有非常重要的意义.

列车运行控制系统是高速铁路的核心技术之一, 是一个深度融合了计算、通信和控制的系统, 同时又是安全攸关的系统, 即系统的任何错误都可能导致灾难性后果, 是一种典型的信息物理融合系统

引用格式: 郭丹青, 吕继东, 王淑灵, 等. 中国高速铁路列控系统的形式化分析与验证. 中国科学: 信息科学, 2015, 45: 417–438, doi: 10.1360/N112014-00017

CPS(cyber-physical systems). 例如, 2011 年 7 月 23 日发生在温州的事故, 导致 40 多位乘客失去生命. 它通过 3C(computation, communication, control) 技术的有机融合与深度协作, 实现列车运行过程与信息交互系统的实时感知、动态控制和信息服务. 随着列车运行速度的不断提高 (运行时速将超过 500 公里/小时), 地面、轨道网络与列车之间、列车与列车之间的交互作用极其复杂, 使得列车运行控制系统建模、性质刻画和性能抽取变得复杂, 最终必然导致对系统安全性分析和验证的复杂程度急剧增加. 列车运行安全控制标准是列车运行控制系统的技术标准, 是实现列车安全运行的技术保障之一. 因此必须确保制定的列车运行安全控制标准是安全可靠的, 即首先保证内部不存在互相矛盾的地方; 其次, 要保证根据标准列车能够安全运行在给定路段上; 再次, 一旦发生事故, 根据标准能够采取相应的应对措施不至于发生灾难性后果.

1.1 本文主要贡献

本文主要研究如何利用信息物理融合系统方面最新结果, 特别是我们最近几年在混成系统建模、分析和验证方面的结果, 提出一套中国高速铁路列控系统图形建模、仿真、形式建模和形式验证于一体的方法, 从而提高中国高速铁路列控系统的可靠性. 具体讲:

Simulink/Stateflow[1,2] 是 Matlab 中一个重要的商业组件, 拥有 Matlab 的强大计算能力支持, 在工业界使用广泛, 拥有深厚的用户基础. 它是一种图形化的建模工具, 对模型的描述比较直观, 并且支持仿真, 可以检查模型是否与预计行为相符, 以便排除部分早期的设计失误, 已经成为一种事实上的工业标准. 同时, 它还可以将建立的模型转化为对应的 C 语言, 有助降低开发成本. 而过去, 很少有使用 Simulink/Stateflow 对列控系统进行建模和仿真的工作. 基于此, 我们首先针对目前中国高速铁路列控系统的规范, 提出一种一般性 Simulink/Stateflow 图形模型框架. 对高速铁路列控系统的行车许可、等级升级及部分模式转换场景建立了 Simlink/Stateflow 模型, 并且使用该模型对二级到三级与模式转换相结合的部分情况进行了仿真, 发现在部分情况下会出现不正常停车以及等级转换失败的现象.

另一方面, 因为仿真仅仅能够根据环境的有穷输入在有限时间内观察系统行为是否满足要求, 而环境的输入通常有无穷多种, 并且这些系统的运行可能没有时间限制, 所以类似于测试, 仿真仅仅能够发现系统错误, 不能够证明系统没有错误. 特别对于安全攸关系统, 仅仅仿真无法保证系统的正确性. 因而对 Simulink/Stateflow 图形模型进行形式验证, 作为仿真的一种补充, 是非常必要的. 形式验证 Simulink/Stateflow 图形模型的前提是需要提供一个形式语义以及针对该形式语义的验证技术. 在文献 [3,4] 中, 我们考虑如何将 Simulink/Stateflow 图形模型转换成 HCSP[5,6] 形式模型, 并使用 HHL[7] 及其定理证明器 [8] 来形式验证转化后的形式模型. 本文的另一贡献是: 我们以其中某个场景为例, 说明如何使用上述方法将其转换为形式模型并严格验证, 验证结果表明在任何输入下均不能正常停车.

1.2 相关工作

为了支持混成系统建模, 人们提出很多建模语言, 例如: Modelica[9], HybridUML[10], 时间自动机 [11], 混成自动机 [12], Esterel[13], 等等. Modelica[9] 与 Simulink 同是图形化的建模语言. 它是一种开源语言, 支持用户自定义修改. 它同时支持图形建模和代码生成, 表达能力很强. 但是它没有类似于 Stateflow 这样的控制流图建模工具, 所以对控制逻辑建模支持比较薄弱. 虽然 Modelica 使用算法和函数对控制逻辑进行建模, 但是由于这种建模方式是基于文本的, 表现不够直观, 不利于对高层模型进行建模. HybridUML[10] 是一种对混成系统进行图形化建模的语言, 它是基于传统 UML 的混成扩展. UML 在工业界有广泛的使用, 用户基础较好. 但是 HybridUML 不能进行仿真, 也不能进行验证, 使用

上有很大的限制. 混成自动机 [12] 是自动机在混成系统方面的拓展, 基于可达集计算的各种验证工作在学术界研究深入, 应用广泛. 基于自动机的建模技术直观, 易于理解, 但是它没有系统结构方面的信息, 组合性差, 不适合大型系统的建模. 特别是, 没有图形化工具支持, 不能进行仿真, 形式验证技术也不适用于复杂大型系统. Esterel[13] 是一种基于时间同步模型的建模语言. 与常见的同步异步通讯方式不同, 当 Esterel 中信号被发送之后, 该时间点上其他并发进程都可以将该信号接收任意有限次. Esterel 拥有成熟的商业工具 Scade[14] 对其进行支持, 该工具生成的代码可适用于实际系统, 减少了后期开发的工作量. 总之, Simulink/Stateflow 是更实用的混成系统建模, 分析和仿真工具.

国内外有许多关于高速铁路列控系统形式建模和验证的工作, 例如: Platzer 等在文献 [15] 利用一种基于微分不变量的微分动态逻辑等技术对欧洲高速铁路列控系统进行建模、分析和验证, 从欧洲高速铁路列控系统的非形式描述中发现了许多不严格的地方, 之后他们还进一步通过设计一个可以严格证明安全性质的形式模型, 以改进欧洲高速铁路列控系统的设计. 国内也有很多科研工作者尝试对中国高速铁路列控系统进行形式建模和验证, 例如: 在文献 [16] 中, 作者结合 UML 在业界的广泛应用及 SMV 模型检验工具的优点, 提出了一套列控系统规范的建模与验证方法. 而在文献 [17] 中, 作者使用 Petri 网对高速铁路列控系统信道模型和数据传输的时间特性进行建模和分析, 为 CTCS-3 级列控系统规范中的相关参数设计提供了前提和基础.

综上分析, 目前还没有一套针对高速铁路列控系统的集图形建模、仿真、形式建模和验证于一体的方法. 本文填补了这方面的空白.

2 背景知识

本节将介绍高速铁路列控系统, Simulink/Stateflow 和 HCSP 以及 HHL 的一些背景知识, 因为我们关心等级转换和模式转换相结合的场景, 所以使用第 2.1 和第 2.2 小节分别介绍高速铁路列控系统等级和列车在不同等级下的不同模式, 第 2.3 小节介绍 Simulink/Stateflow, 第 2.4 小节介绍 HCSP 和 HHL.

2.1 高速铁路列控系统等级概述

中国列车控制系统 CTCS(Chinese train control system), 根据总体原则, 从国情、路情实际出发, 共划分为 5 级 [18], 其中我们关心如下两个等级:

(1) CTCS-2 (C2) 级

基于轨道传输信息的列车运行控制系统, 面向提速干线和高速新线, 采用车、地一体化设计. 适用于各种限速区段, 地面可不设通过信号机, 机车凭车载信号行车. 地面子系统中增加列控中心, 根据列车占用情况及进路状态, 计算行车许可及静态列车速度曲线并传送给列车. 点式信息设备用于向车载设备传输定位信息、进路参数、线路参数、限速和停车信息等.

(2) CTCS-3 (C3) 级

基于无线传输信息, 采用轨道电路等方式检查列车占用的列车运行控制系统. 面向提速干线、高速新线或特殊线路, 基于无线通信的固定闭塞或虚拟自动闭塞. CTCS-3 在 CTCS-2 的基础上有了很多改进. 主要体现在控制中心和信息交互方式的改变上.

2.2 CTCS-2 级和 CTCS-3 级下模式概述

每个等级下列车都有不同的模式, 其中我们只关心 C2 下的完全监控模式 FS(full supervision

mode)、目视行车模式 OS(onsight mode)、部分监控模式 PS(partial supervision mode) 模式, 以及 C3 下的 FS, OS, 引导模式 CO(calling on mode) 和冒进模式 TR(trip mode).

(1) CTCS-2 下模式介绍

完全监控模式 (FS): 在完全监控模式下, 列控车载设备应能判断列车位置和停车位置, 在保证列车速度满足线路固定限速、车辆构造速度、停车位置、临时限速等条件的前提下, 生成目标距离连续速度控制模式曲线并连续监控列车速度, 与模式速度比较, 自动输出紧急制动或常用制动命令, 同时, 应能通过 DMI 显示列车实际速度、允许速度、目标速度和目标距离等信息.

目视行车模式 (OS): 车载设备显示停车信号或位置不确定时, 在停车状态下司机按压专用按钮可使车载设备转入目视行车模式. 在该模式下, 列控车载设备生成 NBP(normal brake profile)[1)]为 20 km/h 的模式曲线.

部分监控模式 (PS): 由于应答器信息接收异常导致线路数据缺失, 或者由于其他原因列控车载设备无线路数据, 以及侧线接车和在车站办理引导接车时, 列控车载设备的工作模式都定义为部分监控模式.

(2) CTCS-3 下模式介绍

完全监控模 (FS): 当车载设备具备列车控制所需的全部基本数据 (包括列车数据、行车许可和线路数据等) 时, 车载设备生成目标距离连续速度控制模式曲线, 并通过 DMI 显示列车运行速度、允许速度、目标速度和目标距离等信息, 监控列车安全运行.

目视行车模式 (OS): 车载设备显示停车信号或位置不确定时, 在停车状态下司机按压专用按钮可使车载设备转入目视行车模式. 目视行车模式下, 车载设备按固定限制速度 40 km/h 监控列车运行, 列车每运行一定距离司机需确认一次.

引导模式 (CO): 当开放引导信号进行接发车时, 车载设备生成目标距离连续速度控制模式曲线, 并通过 DMI 显示列车运行速度、允许速度、目标速度和目标距离等, 车载设备按固定限制速度 40 km/h 监控列车运行, 司机负责在列车运行时检查轨道占用情况.

冒进模式 (TR): 列车执行冒进防护时, 车载设备进入冒进模式. 这个模式下列车会紧急制动, 直至完全停止.

2.3 Simulink/Stateflow 基本概念

2.3.1 Simulink 介绍

Simulink[1] 是 MATLAB 最重要的组件之一, 它提供一个图形化的动态系统建模、仿真和综合分析的集成环境. 在该环境中, 无需大量书写程序, 而只需要通过简单直观的鼠标操作, 就可构造出复杂的系统. Simulink 是用于动态系统和嵌入式系统的多领域仿真和基于模型的设计工具, 拥有丰富的可扩充预定义模块库. 对各种实时系统, 包括通讯、控制、信号处理、视频处理和图像处理系统, Simulink 提供了交互式图形化环境和可定制模块库来对其进行设计、仿真、执行和测试.

Simulink 模型是由一系列的模块和连接这些模块的边组成的. 每个模块可以是一个 Simulink 模块库中的基本模块, 表达输入输出之间的一个数学关系; 也可以是由若干模块组成的子系统; 连接边反应了两个被连接模块之间的关系.

Simulink 模型通常由三部分组成: 输入信号源 (source)、系统 (system) 以及接收模块 (sink).

(1) 输入信号源模块库 (source)

1) 常用制动介入曲线.

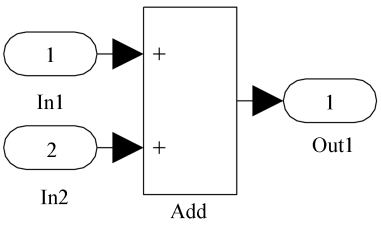


图 1 Simulink 图表

Figure 1 A Simulink diagram

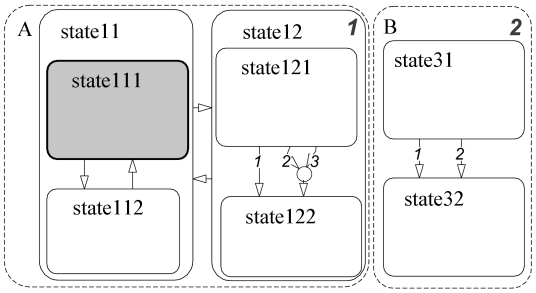


图 2 Stateflow 图表

Figure 2 A Stateflow diagram

输入信号源模块库用来向模型提供输入信号, 没有输入口, 但是至少有一个输出口. 主要有: Constant, Step, Ramp, Sine Wave, Signal Generator, From File, From Workspace, Clock, In.

(2) 系统模块库 (system)

系统模块主要包括连续模块、离散模块、数学模块、逻辑模块、信号处理模块、系统模块. 连续模块主要有: Integrator, Derivative, State-Space, Transfer Fcn, Zero-Pole, Transport Delay. 离散模块主要有: Discrete-time Integrator, Discrete Filter, Discrete State-Space, Discrete Transfer-Fcn, Discrete, Zero-Pole, First-Order Hold, Zero-Order Hold, Unit Delay. 数学模块主要有: Sum, Product, Dot Product, Gain, Math Function, MinMax, Abs, Sign. 逻辑模块主要有: Compare To Zero, Compare to Constant, Inteval Test, Relational Operator, Logical Operator, Bit Set, Bitwise Operator, Bit Clear. 信号处理模块主要有: IC, Bus Creator, Bus Selector, Mux, Demux, Switch, Multiport Switch, Manual Switch, Selector. 系统模块主要有: SubSystem, Triggered Subsystem, Enabled Subsystem, Function-Call Subsystem, If Action Subsystem.

(3) 接收模块库 (sink)

接收模块是用来接收模块信号的, 没有输出口, 但是至少有一个输入口. 主要有: Scope, Display, XY Graph, To File, To Workspace, Stop Simulation, Out.

Simulink 模块从其行为模式上分, 可以分为连续模块和离散模块两种. Simulink 模型中的每个模块都有一个样本时间, 它的取值范围是 -1 或者非负浮点数. 如果这个模块的样本时间是 -1, 表示其样本时间由其来源模块的样本时间计算而来; 如果这个模块的样本时间是 0, 表示这个模块是连续模块, 其计算行为具有连续性; 如果是一个正值 x, 表示该模块为离散模块, 其每隔 x 时间长度重新计算一次.

大多数 Simulink 模块都包含配置参数, 用户可以通过修改这些参数来获得预想的功能. 比如说常数模块可以设置输出常数值, 积分模块可以设置初始值, Switch 模块可以设置操作符号类型和阈值.

图 1 描述了一个 Simulink 模型, 其中的 Add 模块从 In1, In2 接收了两个输入, 通过加法运算产生结果, 并通过 Out1 输出. 该模型中 In1, In2, Out1 和 Add 分别对应于输入信号源, 输出信号源与系统.

2.3.2 Stateflow 介绍

Stateflow[2] 是 Simulink 里面的一个工具箱, 它是一个基于状态机和流程图来构建组合和时序逻辑决策模型并进行仿真的环境. Stateflow 可以将图形表示和表格表示 (包括状态转换图、流程图、状态转换表和真值表) 结合在一起, 针对系统对事件, 基于时间的条件以及外部输入信号的反应方式进

行建模. 用户可以在使用 Simulink 仿真时, 使用这种图形化的工具实现各个状态之间的转换, 对复杂的监控逻辑进行建模.

Stateflow 模型主要由事件变量集合、状态、结点和状态迁移组成:

(1) 事件变量集合

事件变量集合是记录 Stateflow 模型中使用的广播事件和变量的集合. 广播事件和变量均分为输入输出和局部三种, 分别表示从外部环境输入的广播事件或变量, 向外部输出的广播事件或变量和内部使用的广播事件或变量.

(2) 状态

状态代表了系统现在处在的情况. 在 Stateflow 下, 状态有两种行为: 活动的 (active) 和非活动的 (inactive). 可以对一个状态进行标记, 包括给这个状态规定状态名以及这个状态在进入、退出和处于激活状态下接收到一个事件所应执行的动作, 一般表示如下:

name

entry: entry actions

during: during actions

exit: exit actions

bind: data and events

on event_name: on event_name actions.

入口动作 (entry: entry actions) 是表示发生状态迁移, 激活了该状态时需要执行的动作. 中间动作 (during: during actions) 表示原处于激活的状态受到一个事件的触发, 不存在从这个状态发出的状态迁移时, 此状态仍处于激活状态需要执行的动作. 出口动作 (exit: exit actions) 表示存在由此状态发出的有效状态迁移时, 该状态退出时执行的动作. 数据事件绑定动作 (bind: data and events) 将数据和事件绑定在此状态上. 绑定的数据只能在此状态或其子状态内被改写, 其他状态只能读取此数据. 绑定的事件由此状态或其子状态广播. 特定事件发生动作 (on event_name: on event_name actions). event_name 规定一个特定的事件; on event_name actions 表示当该状态是激活状态且 event_name 规定的事件发生时需要执行的动作.

(3) 结点

结点用于描述状态迁移过程中的迁移信号的分离和汇合. 通过使用结点, 状态和状态之间的迁移不仅仅是边的简单连接, 而且是一个复杂的迁移网络.

(4) 状态迁移

状态迁移[2)]一般用于连接两个状态或者结点, 它可以被标记, 该标记的一般形式为如下四元组:

Event Trigger[Condition]Condition Action/Transition Action.

触发事件 (evnet trigger) 是事件集合中的一个元素, 作为迁移条件的一部分. 条件 (condition) 是一个布尔表达式, 与触发事件共同组成状态迁移的迁移条件. 触发事件 (evnet trigger) 为空或者与当前广播事件一致, 同时条件 (condition) 为真, 则该状态迁移被触发. 条件动作 (condition action) 和迁移动作 (transition action) 是使用 Matlab 的 action language 书写的代码片段. 条件动作在迁移被触发时被立即执行, 而迁移动作仅当迁移终点为状态时被立即执行, 否则被寄存在一个队列中, 若该迁移最终达到一个状态, 队列中的迁移动作将被顺序执行.

Stateflow 模型具有层次化的特性, 任意 Stateflow 状态均可嵌入一个 Stateflow 子模型来丰富该状

2) 默认迁移是一种特殊的迁移, 它仅有一个目标状态, 表示该状态为默认激活状态.

态的行为. 同一层次中, 所有的状态是互斥 (OR) 或者并行 (AND) 的, 互斥状态之间可以使用迁移进行连接, 而并行状态之间不能使用迁移连接.

Stateflow 通讯采用广播机制, 当一个事件被广播时, 并行状态按照其预先定义的顺序先后接收到该广播事件, 互斥状态仅有激活状态收到该广播事件. 当状态接收到一个广播事件之后, 进行如下操作: 按照预先定义的顺序, 对状态的所有外部出口迁移进行检查, 如果该迁移网络能成功到达某个状态, 则执行对应的迁移动作, 并完成该轮事件广播; 否则, 按照预定的顺序对内部出口迁移进行检查, 如果该迁移网络能成功到达某个状态, 则执行对应的迁移动作, 并完成该轮事件广播; 如果上述两项均不成功, 则执行中间动作. 然后把广播事件对其包含的子图进行广播. 当状态迁移成功时, 需要找到源状态和目的状态之间的共同路径, 从内至外, 执行源状态出口动作至共同路径, 然后从外向内, 执行目标状态的入口动作.

图 2 描述了一个 Stateflow 的模型, 该图中有两个并行的状态 A 与 B, 同时处于激活状态. 状态 A 和状态 B 中分别包含六个和两个互斥状态, 它们层次结构如图 2 所示. 图 2 中的迁移边包含三种特殊情况: 以实心黑点开头的迁移边为默认迁移边, 表明该箭头所指状态为默认激活状态; 状态 S11 到 S2 的迁移为层间迁移; 状态 S21 到 S22 的迁移为迁移网络, 该迁移网络包含一个结点.

2.4 混成 CSP 简介

2.4.1 混成 CSP

HCSP(hybrid communicating sequential process)[5,6] 在 CSP(communicating sequential process) 的基础上引入了微分方程以描述在混成系统中的连续动态行为, 并同时引入各种中断机制 (条件中断和通讯中断) 来中断一个连续的微分行为, 进而使连续微分行为和离散控制行为交错运行. HCSP 能同时刻画系统的连续行为和控制逻辑, 并具有 CSP 高可组合性的特性, 是一种很好的用来描述大型控制系统的形式化建模语言.

HCSP 的语法如下所示:

$$\begin{aligned} P ::=\ & \text{skip} \mid x := e \mid \text{ch}?x \mid \text{ch}!e \mid P; Q \mid B \to P \mid P \sqcup Q \mid P^* \\ & \mid \langle \mathcal{F}(\dot{s}, s) = 0 \& B \rangle \mid \langle \mathcal{F}(\dot{s}, s) = 0 \& B \rangle \trianglerighteq [\!]_{i \in I} (\text{io}_i \to Q_i) \\ S ::=\ & P \mid S \| S. \end{aligned}$$

上式中 P, Q, Q_i, S 均为 HCSP 进程, x 和 s 为进程变量, ch 为通道名, io_i 为通信事件 (输入事件 ch?x 或输出事件 ch!e), B 和 e 为布尔表达式和算术表达式. 以上语法结构的非形式化语义表示如下: skip 不执行任何操作, 立即终止; $x := e$ 将表达式 e 的值赋给 x; ch?x 从通道 ch 中得到一个值, 并将其赋给 x; ch!e 将 e 的值发送到通道 ch; $P; Q$ 先执行进程 P, 当进程 P 终止时执行进程 Q; $B \to P$ 在 B 为真时执行 P, 否则立刻终止; $P \sqcup Q$ 由系统随机选择是执行进程 P 还是 Q; P^* 表示有限次重复执行进程 P; $\langle \mathcal{F}(\dot{s}, s) = 0 \& B \rangle$ 代表微分动态过程, 该微分过程相关的变量取值必须使得布尔表达式 B 始终取真值, 否则该语句立即终止; $\langle \mathcal{F}(\dot{s}, s) = 0 \& B \rangle \trianglerighteq [\!]_{i \in I} (\text{io}_i \to Q_i)$ 与 $\langle \mathcal{F}(\dot{s}, s) = 0 \& B \rangle$ 执行过程基本一致, 但是当通信事件 io_i 发生时会立刻执行对应进程 Q_i, 否则它会一直执行到微分过程终止为止; $S_1 \| S_2$ 中 S_1 和 S_2 为并发进程, 在不需要通讯时各自独立运行, 当需要时通讯双方需要同步执行该通讯过程, 并发进程之间不能存在共享变量或是共享输入或者输出通道.

2.4.2 混成 Hoare 逻辑

文献 [7] 通过在经典的 Hoare 逻辑中引入历史公式构建了 HCSP 的证明系统. 之后, 文献 [8] 将

该逻辑在 Isabelle/HOL 中实现, 并利用该工具验证了中国铁路控制系统中的一个实际案例. 历史公式由一个时段验算公式 [19,20] 表达, 用于描述系统在执行时间区段内满足的性质. 在混成 Hoare 逻辑 (HHL) 中, 顺序进程 P 的性质描述由三元组 $\{\text{pre}\}P\{\text{post};\text{HF}\}$ 来表达, 其中 pre, post 和 HF 分别表示前置条件、后置条件和历史公式. 前置条件和后置条件由一阶逻辑公式表达, 历史公式由时段验算公式表达. 对于并发进程其性质描述如下表示:

$$\{\text{pre}_1,\ldots,\text{pre}_n\}P_1 \parallel \cdots \parallel P_n\{\text{post}_1,\ldots,\text{post}_n;\text{HF}_1,\ldots,\text{HF}_n\},$$

其中 pre_i, post_i 和 HF_i 分别表示第 i 个进程的前置条件、后置条件和历史公式.

3 场景描述及其建模过程

我们将高速铁路列控系统在等级转换和模式转换的两个场景中涉及的主体抽象为由列控系统、列车与司机组成. 列控系统分为多个等级, 只考虑列车在 C2 级和 C3 级下面的行为, 所以只涉及 C2 级列控系统与 C3 级列控系统, 分别是 TCC(train control center), RBC(radio block center), 其主要作用是根据车载子系统、地面子系统等提供的列车状态、轨道占用、临时限速命令、联锁进路状态、灾害防护等信息产生其控制范围内各个列车的行车许可等控制信息, 并且传输给车载设备以控制列车的运行. 列车运行的过程中, 司机要监控列车的运行, 需要经常对不同的情况作出反应.

图 3 是表示整个列控系统的 Stateflow 模型, 其中 C2 级控制器、C3 级控制器、列车和司机四个部分分别由 TCC, RBC, Train 和 Driver 四个状态表示, 这四个部分需要并行执行, 所以这四个状态之间的关系为并行. 列车在运行过程中需要一个动力系统来计算速度, 由于动力系统是连续的, 因此我们在 Stateflow 外使用一个子系统来表示, 如图 4 所示, 这个子系统的输入是加速度, 输出是速度和距离, 使用了两个积分模块, 分别代表加速度的积分为速度, 速度的积分为距离.

运营场景是对运营中系统工作方式的简要描述, 在这个模型中, 我们考虑三个运营场景: 行车许可 MA(movement authority) 场景, 控制器会对列车的运行过程进行监控, 并且通过 MA 授权控制列车的行为; 等级转换场景, 列车在运行中可以从 C2 级转换到 C3 级, 图 5 是等级升级场景的一个示意图; 模式转换场景, 列车在不同的模式下面也会有不同的行为模式, 不同等级下面不同的模式之间的转换有不同的行为模式. 下面将详细介绍这三个场景. 因为从文档中推测等级转换和模式转换相结合可能会产生问题, 因此我们将模拟等级转换和模式转换相结合的场景, 其中等级转换点和模式转换点是同一点.

3.1 行车许可

MA 是列车安全运行的行车凭证. MA 终点是指列车被授权运行到的位置. 一个行车许可包括多个连续的区段, 每个区段由一个五元组表示, 具体如下:

MA 区段 ={等级、模式、目标距离、紧急制动介入速度、常用制动介入速度}, 从之前的描述中可以知道, 列车的行为模式均由列车当时处在的等级和模式所决定, 即五元组的第一项和第二项. 模型只涉及 C2 级与 C3 级两个等级, 分别用 1 和 2 表示, 并且只考虑 FS, PS, CO, OS, TR 五个模式, 分别用 1, 2, 3, 4, 5 表示.

列车在运行过程中会不断地更新 MA 列表. 我们假定列车在运行完一个区段之后, 会向 TCC(C3 级下为 RBC) 申请新的区段, 发出 ch_ma2 (ch_ma3) 信号, TCC(RBC) 接到该申请之后, 将查看列车

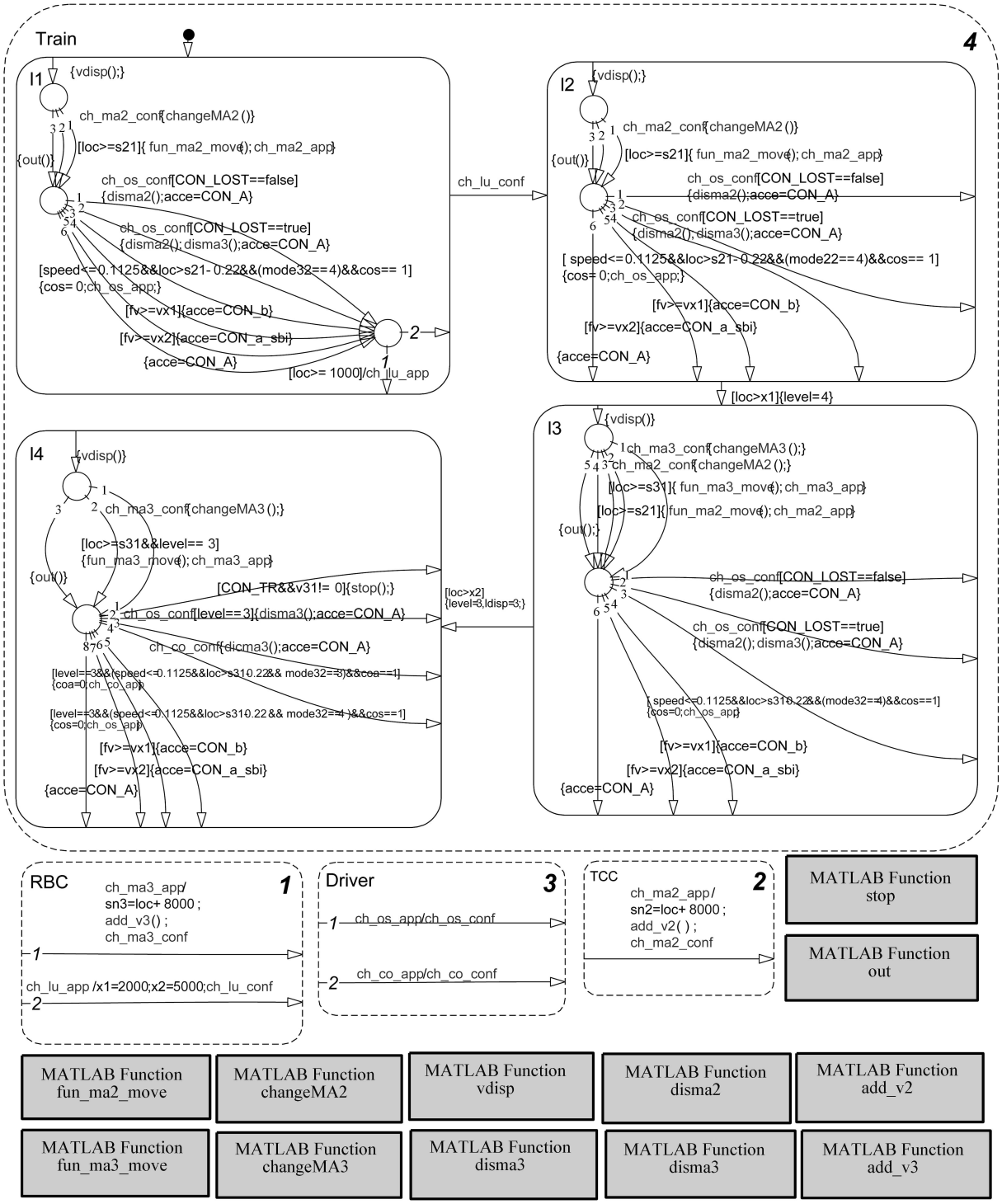


图 3 列控系统

Figure 3 The train control system

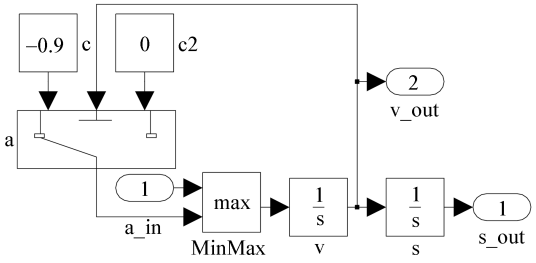


图 4　动力系统

Figure 4　Speed system

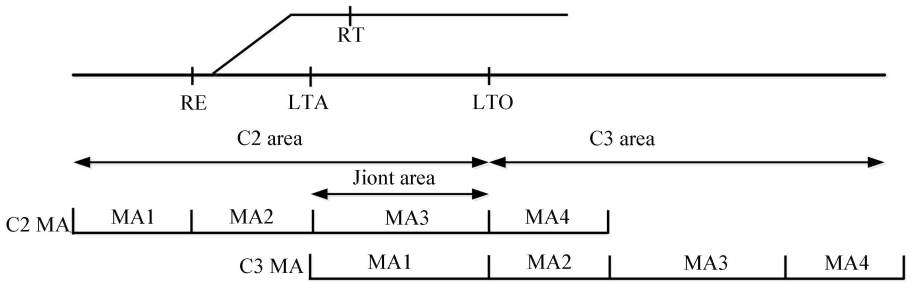


图 5　等级转换场景

Figure 5　The scene of level change

允许的位置生成新的 MA, 并且发送 ch_ma2_conf (ch_ma3_conf) 和新的 MA 给列车, 如 TCC 中的迁移和 RBC 中的迁移 1 所示. 列车接收到新的 MA 之后将覆盖原有 MA.

列车会不断根据目前的 MA 计算列车有效加速度范围, 计算主要依据列车的两类曲线:

(1) 静态速度曲线 SSP(static speed profile)

由每个区段的所有最高速度限制组成, 包括紧急制动介入速度和常用制动介入速度两个速度限制.

(2) 动态速度曲线 DSP(dynamic speed profile)

包括紧急制动介入曲线和常用制动介入曲线, 由 MA 的目标距离和静态速度曲线计算而来. 列车在运行的过程中任意时刻需保证不能超越静态速度曲线和动态显示曲线.

(3) 紧急制动介入曲线 EBI(emergency brake intervention)

是根据目标距离和紧急制动介入速度计算而来的速度曲线, 如果列车当前速度超出了紧急制动介入曲线, 列车就要以最大减速度进行紧急制动.

(4) 常用制动介入曲线 SBI(service brake intervention)

是根据目标距离和常用制动介入速度计算而来的速度曲线, 如果列车目前的速度超出了常用制动介入曲线, 那么列车要以常用制动减速度进行制动.

如果列车的当前速度没有超出常用制动介入曲线, 那么就把列车的加速度设置为常用制动减速度以及最大加速度之间的某个值.

在模型中我们使用 out 函数计算列车当前是否超过 EBI 或者 SBI. 如果超过了 EBI, 则变量 eout 的值设为 1, 如果超过了 SBI, 则变量 sout 的值设为 1, 在之后的迁移中分别查看 eout 和 sout 的值设置加速度值 a. 列车在运行中的每一刻都要计算拥有的所有 MA 区段的常用制动介入曲线和紧急制动介入曲线, 每一点均取所有 MA 区段的紧急制动介入曲线对应位置的最小值作为当前的紧急制动介

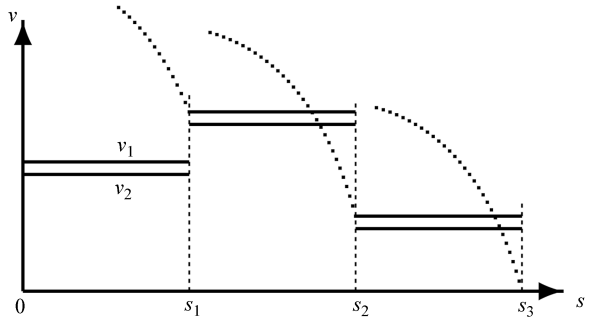


图 6 MA 示例
Figure 6 Example of MA

入速度, 取所有 MA 区段的常用制动介入曲线对应位置的最小值作为当前的常用制动介入速度, 这样叠加形成该区段的紧急制动介入曲线和常用制动介入曲线, 也即列车运行中的每一刻都不可以超越目前拥有的任何一个 MA 区段计算得到的紧急制动介入曲线和常用制动介入曲线, 这样确保如果列车在这个区段正常运行, 就不会违反下个区段起点的速度上限.

图 6 是一个有三个 MA 区段的例子, 由点 s_1, s_2, s_3 分开. 这个例子中, 我们假设最后一个区段的限速曲线终点速度为零, 因此之后如果不继续拓展列车的 MA 区段列表, 列车就需要在 s_3 点停下. 图中的六条直线分别代表三个 MA 区段的静态速度曲线, 六条曲线 (最后一段的紧急制动介入曲线和常用制动介入曲线重合, 因为末点的紧急制动介入速度和常用制动介入速度都是 0) 分别代表三个 MA 区段各自根据上述公式计算得到的紧急制动介入曲线和常用制动介入曲线. 根据如下公式进行计算动态曲线 $v^2 + 2b\,s < \text{next(seg)}.v_1^2 + 2b\,\text{seg}.s$, 这里的 next(seg) 代表列车下一个要运行的 MA 区段, b 代表紧急制动加速度, seg.s 代表这一段 MA 区段的终点位置.

3.2 等级转换

等级转换是列控系统的主要功能之一, 列车由 C2 级线路进入 C3 级线路, 需要在固定地点进行转换. 等级转换前, 应完成的工作包括注册到 GSM-R(global system mobile-railway) 网络, 与 RBC 建立通信会话, 从 RBC 获得系统配置参数和行车许可等信息. 直至 C3 级控车条件具备后, 车载设备自动转入 C3 级工作. 如图 5 所示. 主要过程如下:

(1) 列车与 GSM-R 建立链接, 与 RBC 建立通信会话

从 C2 级控车转换为 C3 级控车首先应建立车载设备与 GSM-R 的通信链接. 车载设备与 GSM-R 网络建立链接后, 从 C2 级转换为 C3 级应建立车载设备与 RBC 的通信会话, 为此在转换区入口处设置 RBC 连接应答器组 (RE), 该应答器组应向接近的列车发送用于建立通信会话的命令. 列车之后可能进入不通往 C3 级的分支, 这个时候应该设置一个命令列车关闭通信会话的应答器组 (RT).

(2) 获得行车许可

C3 级系统应只向真正进入 C3 级区域的列车提供行车许可. C3 级系统在至转换边界前唯一进路的区域设置用于向车载设备提供准确进路的级间转换预告应答器组 LTA(level transition announcement). 当列车前端通过预告应答器组 (LTA) 时, 车载设备向 RBC 报告列车位置, RBC 据此确定列车接近的准确进路, 然后根据 C3 级控制区域联锁进路信息, 向车载设备提供包括线路参数的行车许可 (MA) 及级间转换命令. 列车在到达转换边界前始终由 C2 级系统控车, 车载设备将储存此前获得的 C3 级

行车许可 (MA) 并在其通过 C2/C3 级边界并转换到 C3 级系统控车时使用. 在获得行车许可到等级转换执行这段期间内, 列车会受到 C2 和 C3 级行车许可的共同控制.

(3) 执行等级转换

当列车前端通过 C2/C3 级边界, 根据 RBC 的命令并具备 C3 级系统控车条件时, 车载设备将自动转为 C3 级控车. 在 C2/C3 级间转换边界设置转换执行应答器组 LTO(level transition operation) 向列车发送消息. 为保持在等级转换边界列车能以 C2 级最高允许速度运行, C2 级系统的行车许可必须越过 C2/C3 级边界至少一个完整常用制动距离. 之后在 C3 级系统控车过程中, C2 级车载设备控制单元处于后台工作状态, 不对列车有控制作用.

图 3 中 Train 模块展示了等级转换场景, 该模块中的四个状态分别代表列车在运行至 RE 之前、RE 到 LTA 之间、LTA 到 LTO 之间、与越过 LTO 之后四个阶段. 第一个阶段中列车正常启动运行, 直到到达 RE 建立连接. 第二个阶段中, 列车已经与 RBC 通过 GSM-R 网络建立了连接, 列车向 RBC 报告了目前位置, 并且收到了 RBC 发送的等级转换预备点和等级转换执行点位置, 列车继续在 C2 级 MA 控制下, 直到列车到达 LTA. 第三个阶段中, 列车已经接收到 RBC 发来的 MA, 列车开始受到 C2 级和 C3 级的共同控制, 直到到达 LTO. 如果列车越过等级转换点则标志着等级转换成功, 列车进入 C3 级运行.

3.3 模式转换

C2 级与 C3 级的模式转换有一些不同的行为方式, 这一小节将分别详细介绍 C2 级下面和 C3 级下面的模式转换.

(1) C2 级模式转换

当列车缺少线路数据, 便可进入 PS 模式. 当列车接收到的行车许可模式为 OS 时, 列车便可以转入 OS, 转入的条件是列车在非 OS 区段末停车, 显示目视行车键, 当司机按压了目视行车键后, 便可以转入 OS 模式. 我们使用列车模块向司机模块发出 ch_os_app 信号, 司机模块在收到之后向列车模块返回 ch_os_conf 信号来模拟 OS 模式的目视行车键请求过程. 列车在引导区域可以进入引导区的 PS 模式, 这个时候行为模式与 PS 模式一致, 但是上限速度比 PS 模式要低. 进入这种模式列车不需要进行确认, 只需要速度降低到该模式允许速度. 当线路数据完整时, 列车只要接收到 FS 模式的行车许可就可以转入 FS 模式. 图 3 中 I1, I2, I3 均实现了这些模式转换.

(2) C3 级模式转换

当列车接收到的行车许可模式为 CO 时, 列车便可以转入 CO, 转入的条件是列车速度降低到 CO 模式的允许速度, 并且司机进行确认. 我们使用列车模块向司机模块发出 ch_co_app 信号, 司机模块在收到之后向列车模块返回 ch_co_conf 信号来模拟 CO 模式的目视行车键请求过程. 当列车接收到的行车许可模式为 OS 时, 列车便可以转入 OS, 转入的条件是列车在非 OS 区段末停车, 显示目视行车键, 当司机按压了目视行车键后, 便可以转入 OS 模式, 确认行为的模拟方式与 C2 级一致. 当线路数据完整时, 列车接收到 FS 模式的行车许可便可以转入 FS 模式. 当列车运行越过等级转换点的同时接收到 RBC 发送的缩短 MA 命令, 自动转到 C3 级下的 TR 模式, 这时列车进行紧急制动, 最后停在等级转换点之后的某一点. I4 实现了这些模式转换.

4 模拟结果

上述模型包含了第 3 章节描述的所有场景, 通过修改模型参数, 可以模拟不同的组合情况. 需要

表 1 仿真常数列表

Table 1 List of simulation parameters

v21, v22, v23, v24	The limit speeds of emergency braking of the 4 MA segments in C2
vr21, vr22, vr23, vr24	The limit speeds of normal braking of the 4 MA segments in C2
s21, s22, s23, s24	The distances of the targets of the 4 MA segments in C2
mode21, mode22, mode23, mode24	The modes of the 4 MA segments in C2
v31, v32, v33, v34	The limit speeds of emergency braking of the 4 MA segments in C3
vr31, vr32, vr33, vr34	The limit speeds of normal braking of the 4 MA segments in C3
s31, s32, s33, s34	The distances of the targets of the 4 MA segments in C3
mode31, mode32, mode33, mode34	The modes of the 4 MA segments in C3
CON_LOST	Mark whether or not the confirming message of OS mode in C2 controller has delivered to C3 controller
CON_TR	Mark whether or not the train has received the command to shorten the MA in C3

修改的参数包括如下两个方面: 通过修改 MA 初始值来修改轨道参数信息, 从而对不同的组合情况进行仿真; 通过修改仿真常数 (表 1 为仿真常数列表) 将模型的行为确定化.

C2 级 4 个初始 MA 区段和 C3 级 4 个初始 MA 区段位置并不一致, 参照图 5 中显示. C2 级 MA 从 RE 之前开始, 但是 C3 级 MA 从经过 LTA 之后开始, 这表示只有在经过 LTA 之后列车才接收到 C3 级 MA 并且对列车的运行起到控制作用.

利用建立好的模型, 我们对部分组合情况进行了仿真, 结果如图 7 所示. 图 7 由 12 幅子图构成, 每幅子图描述了一个组合情况的运行结果. 每幅子图横轴均表示时间, 纵轴为仿真信号的数值结果, 图中粗实线 (绿线) 表示列车所在位置 (单位为 150 m) 随时间的变化情况, 细实线 (蓝线) 表示列车运行速度 (单位为 m/s) 随时间的变化情况, 稀疏虚线 (紫线) 表示列车的常用制动介入曲线, 紧密虚线 (红线) 表示列车所处等级 (10 表示处于 C2 级, 15 表示处于 C3 级) 随时间变化情况.

(1) C2→C3&&OS→TR

为了模拟该组合情况, 我们将 C2 级和 C3 级 MA 模式均设置为 OS, 为了表示在列车越过等级转换点的同时接收到缩短 MA 命令, 我们将 CON_TR 设置为 true.

仿真结果如图 7(a) 所示, 列车在等级转换点之前依照 C2 级 OS 模式运行, 列车越过等级转换点的同时接收到缩短 MA 的消息, 立即进入 TR 模式, 列车紧急刹车, 停在等级转换点之后某处.

(2) C2→C3&&OS→CO

为了模拟该组合情况, 我们将 C2 级 MA 前 3 段模式设置为 OS, 第 4 段的模式设置为 PS; 同时将 C3 级 MA 第 1 段模式设置为 OS, 后 3 段模式设置为 CO.

仿真结果如图 7(b) 所示, 列车在等级转换点之前依照 C2 级 OS 模式运行, 依照 MA 设定之后转

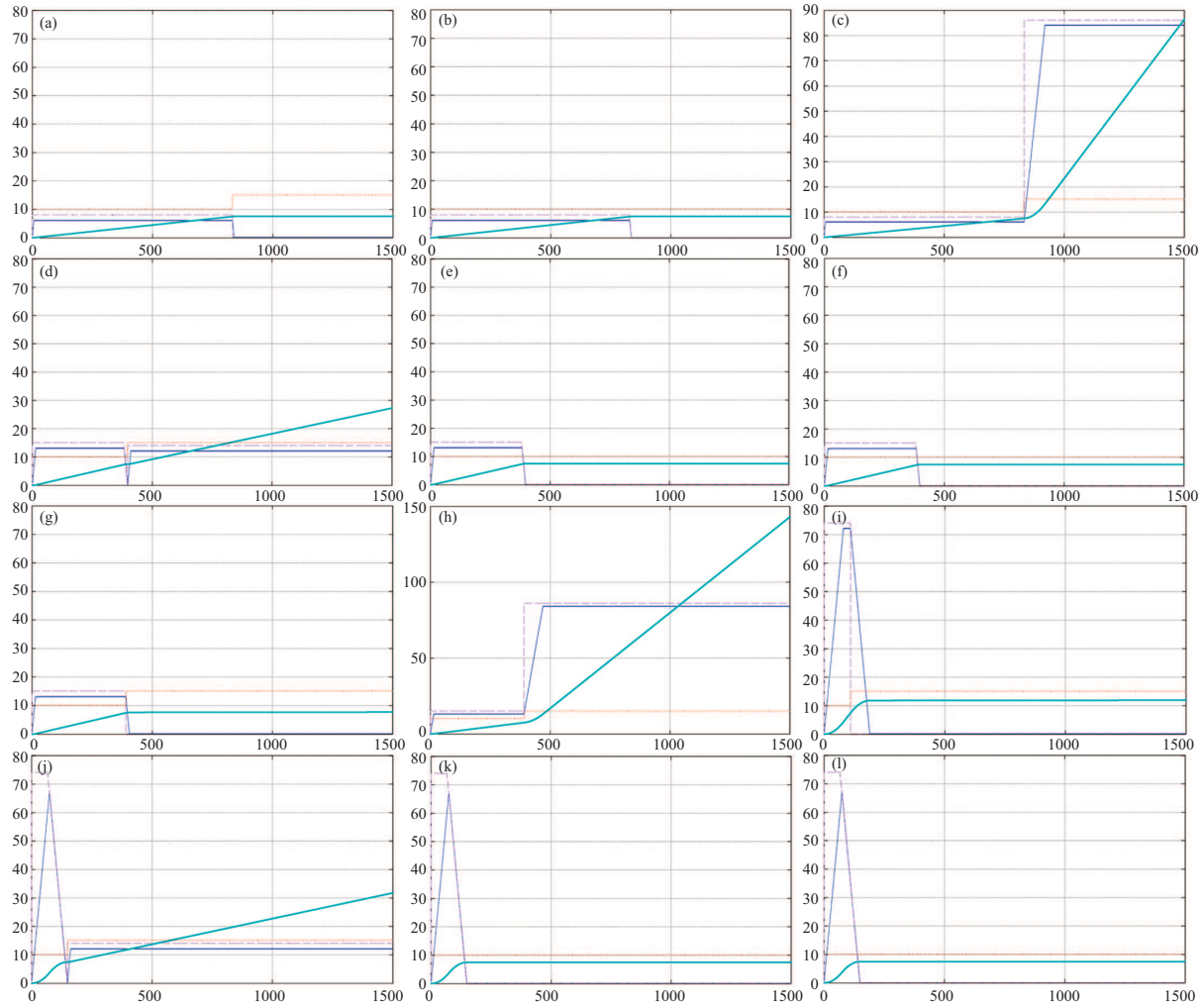


图 7　仿真结果

Figure 7　Simulation result. (a) C2→C3&&OS→TR; (b) C2→C3&&OS→CO; (c) C2→C3&&OS→FS; (d) C2→C3&&PS→OS success; (e) C2→C3&&PS→OS fail; (f) C2→C3&&PS→CO; (g) C2→C3&&PS→TR; (h) C2→ C3&&PS→FS; (i) C2→C3&&FS→TR; (j) C2→C3&&FS→OS success; (k) C2→C3&&FS→OS fail; (l) C2→ C3&&FS→CO

入 C3 级 CO 模式, 但是在 C2 级下无法显示列车在 C3 级转入 CO 的确认键信息, 所以司机无法对进入 C3 级 CO 模式进行确认, 因此转入 CO 模式失败, 列车在等级转换点处停车.

(3) C2→C3&&OS→FS

为了模拟该组合情况, 我们将 C2 级 MA 前 3 段模式设置为 OS, 第 4 段模式设置为 FS; C3 级 MA 第 1 段模式设置为 OS, 后 3 段模式设置为 FS.

仿真结果如图 7(c) 所示, 列车在等级转换点之前依照 C2 级 OS 模式运行, 越过等级转换点之后转入 C3 级依照 FS 模式运行.

(4) C2→ C3&&PS→ OS

为了模拟该组合情况, 我们将 C2 级 MA 前 3 段模式设置为 PS, 第 4 段模式设置为 OS; C3 级 MA 第 1 段模式设置为 FS, 后 3 段模式设置为 OS. 为了模拟目视行车键确认消息从 C2 级传递到 C3

级成功的情况, 设置 CON_LOST 为 true, 为了模拟目视行车键确认消息从 C2 级传递到 C3 级失败的情况, 设置 CON_LOST 为 false.

目视行车键确认消息从 C2 级传递到 C3 级成功的仿真结果如图 7(d) 所示, 列车在等级转换点之前依照 C2 级 PS 模式运行, 在等级转换点停车, 显示目视行车键, 司机确认后, 成功转入 C3 级 OS 模式, 列车继续运行. 目视行车键确认消息没有从 C2 级传递到 C3 级下的情况如图 7(e) 所示: 列车在等级转换点之前依照 C2 级 PS 模式运行, 在等级转换点停车, 显示目视行车键, 司机的确认后, 确认消息返回给 C2 级控制器, 但是由于故障, 确认消息未能传递给 C3 级控制器, 模式转换不成功, 停在等级转换点.

(5) C2→ C3&&PS→ CO

为了模拟该组合情况, 我们将 C2 级 MA 前 3 段模式设置为 PS, 第 4 段模式设置为 CO; C3 级 MA 第 1 段模式设置为 FS, 后 3 段设置为 CO.

该场景仿真结果如图 7(f) 所示, 列车在等级转换点之前依照 C2 级 PS 模式运行, 依照 MA 设定之后要转入 C3 级 CO 模式, 但是在 C2 级下无法显示列车在 C3 级转入 CO 的确认键信息, 所以司机无法对进入 C3 级 CO 模式进行确认, 因此转入 CO 模式失败, 列车在等级转换点处停车.

(6) C2→ C3&&PS→ TR

为了模拟该组合情况, 我们将 C2 级 MA 模式均设置为 PS; C3 级 MA 模式均设置为 FS, 为了表示在列车越过等级转换点的同时接收到缩短 MA 命令, 我们将 CON_TR 设置为 true.

结果如图 7(g) 所示, 列车在等级转换点之前依照 PS 模式运行, 越过等级转换点之后立即进入 TR 模式, 列车紧急制动, 停在等级转换点之后某处.

(7) C2→ C3&&PS→ FS

为了模拟该组合情况, 我们将 C2 级 MA 模式前 3 段设置为 PS, 第 4 段设置为 FS; C3 级 MA 模式均设置为 FS.

结果如图 7(h) 所示, 列车在等级转换点之前依照 C2 级 PS 模式运行, 越过等级转换点之后转入 C3 级依照 FS 模式运行.

(8) C2→C3&&FS→TR

为了模拟该组合情况, 我们将 C2 级 MA 模式均设置为 FS; C3 级 MA 第 1 段模式设置为 FS, 为了表示在列车越过等级转换点的同时接收到缩短 MA 命令, 我们将 CON_TR 设置为 true.

结果如图 7(i) 所示, 列车在等级转换点之前依照 FS 模式运行, 越过等级转换点之后立即进入 TR 模式, 列车紧急制动, 停在等级转换点之后某处.

(9) C2→C3&&FS→OS

为了模拟该组合情况, 我们将 C2 级 MA 前 3 段模式设置为 FS, 第 4 段模式设置为 OS; C3 级 MA 第 1 段模式设置为 FS, 后 3 段模式设置为 OS. 为了模拟目视行车键确认消息从 C2 级传递到 C3 级成功的情况, 我们设置 CON_LOST 为 true, 为了模拟目视行车键确认消息从 C2 级传递到 C3 级失败的情况, 我们设置 CON_LOST 为 false.

目视行车键确认消息从 C2 级传递到 C3 级成功的仿真结果如图 7(j) 所示, 列车在等级转换点之前依照 C2 级 FS 模式运行, 在等级转换点停车, 显示目视行车键, 司机确认后, 成功转入 C3 级 OS 模式, 列车继续运行. 目视行车键确认消息没有从 C2 级传递到 C3 级下的情况如图 7(k) 所示: 列车在等级转换点之前依照 C2 级 FS 模式运行, 在等级转换点停车, 显示目视行车键, 司机的确认后, 确认消息返回给 C2 级控制器, 但是由于故障, 确认消息未能传递给 C3 级控制器, 模式转换不成功, 停在等级转换点.

表 2 组合情况结果

Table 2 Result of the combination

C2→C3&&OS→TR	Stop normally
C2→C3&&OS→CO	Stop abnormally
C2→C3&&OS→FS	Run normally
C2→C3&&PS→OS	Maybe stop abnormally
C2→C3&&PS→CO	Stop abnormally
C2→C3&&PS→TR	Stop normally
C2→C3&&PS→FS	Run normally
C2→C3&&FS→TR	Stop normally
C2→C3&&FS→OS	Maybe stop abnormally
C2→C3&&FS→CO	Stop abnormally

(10) C2→C3&&FS→CO

为了模拟该组合情况, 我们将 C2 级 MA 前 3 段模式设置为 FS, 第 4 段模式设置为 CO; C3 级 MA 第 1 段模式设置为 FS, 后 3 段模式设置为 CO.

结果如图 7(l) 所示, 列车在等级转换点之前依照 C2 级 FS 模式运行, 依照 MA 设定之后要转入 C3 级 CO 模式, 但是在 C2 级下无法显示列车在 C3 级转入 CO 的确认键信息, 所以司机无法对进入 C3 级 CO 模式进行确认, 因此转入 CO 模式失败, 列车在等级转换点处停车.

综上所述, 我们通过对上述 10 个情景进行仿真, 发现部分情景中会出现不正常停车的情况, 具体如表 2 所示.

5 形式验证

由表 2 可知, 在部分组合情况下会出现停车现象. 然而, 由于仿真仅能选择一种系统的可能运行情况进行测试, 并不能表明系统的所有运行情况. 例如, 在我们的模型中, 司机的行为具有很明确的确定性, 即永远选择所有可能加速度里面最大的加速度进行行车. 从某种意义上说, 可以猜测这种选择是火车最危险的行车方式, 也是火车最不可能停车的情况. 由于仿真与生俱来的不完备性, 我们对其中 PS 到 CO 的情况的模型转化为 HCSP 进程, 并利用其定理证明器 HHL Prover 进行了证明. 通过证明我们确认在这种组合场景下任何情况都会发生不正常停车. 在证明过程中, 由于我们仅关心 PS 到 CO 模式转化的情况, 在我们验证的模型中将与此不相关的部分内容进行了简化处理.

5.1 形式转换

通过使用工具 Sim2HCSP[3,4], 我们获得了该模型的 HCSP 的进程, 它包括 7 个文件, 分别对应于 Simulink 和 Stateflow 的变量定义、进程定义、断言定义和最终的证明目标, 这些文件一共有 1351 行, 他们分别是: controlPDef.thy, controlADef.thy, controlVarDef.thy, varDef.thy, assertionDef.thy, processDef.thy 和 goal.thy. 其中最后三个文件主要用于形式证明, 将在下一章节介绍. 下面, 我们简要介绍前四个文件内容:

processDef.thy 主要结构如下, 主要定义了转化后的 HCSP 模型的整体框架, 该框架包含了 Stateflow 图对应的 HCSP 进程 Pcontrol.

```
theory varDef
      imports "HHL"
begin
(*Define channel names.*)
definition Ch_plant_v_1_1::cname where
  "Ch_plant_v_1_1=="Ch_plant_v_1_1""
...
(*Define receiving variables.*)
definition plant_v_1_1 :: exp where
      "plant_v_1_1 ==
      RVar "plant_v_1_1""
...
(*Local and sending variables.*)
definition plant_v_1::exp where
      "plant_v_1==RVar "plant_v_1""
definition plant_s_1 :: exp where
      "plant_s_1 == RVar "plant_s_1""
...
end
```

```
theory processDef
  imports "controlPDef"
begin
(*Define continuous processes*)
...
definition PC1_init :: proc where
"PC1_init == PC1_1;assertion2;PC1_2"
definition PC1_rep :: proc where
  "PC1_rep == PC1_3;assertion3;PC1_4"
definition PC1 :: proc where
  "PC1 == PC1_init;assertion4;(PC1_rep)*"
(*Define the whole process.*)
definition P :: proc where
"P == PC1||Pcontrol"
end
```

controlVarDef.thy 主要结构如下, 主要定义了 Stateflow 图对应的 HCSP 进程中使用的变量.

```
theory controlVarDef
    imports "assertionDef"
begin

definition BR_1 :: cname where
    "BR_1 == "BR_1""
definition BO_1 :: cname where
    "BO_1 == "BO_1""
definition VO1 :: cname where
    "VO1 == "VO1""
definition VI1 :: cname where
    "VI1 == "VI1""
definition vBO1 :: exp where
    "vBO1 == SVar "vBO1""
...
(*Define event variables assistent.*)
consts eL :: "exp list"
consts nL :: "exp list"
(*Define event variables.*)
definition E1 :: exp where
(*Define channel names.*)
definition BC_1 :: cname where
    "BC_1 == "BC_1""

...
    "E == SVar "E""
definition num :: exp where
    "num == RVar "num""
definition EL :: exp where
    "EL == List eL"
definition NL :: exp where
    "NL == List nL"
(*Define local and sending variables.*)
definition s :: exp where
    "s == RVar "s""
definition v :: exp where
    "v == RVar "v""
definition a :: exp where
    "a == RVar "a""
definition CONFR :: exp where
```

```
    "E1 == SVar "E1""                    "CONFR == RVar "CONFR""
definition done1 :: exp where            ...
    "done1 == RVar "done1""              end
```

controlPDef.thy 主要结构如下, 主要定义了 Stateflow 图对应的 HCSP 进程, 该进程作为一个进程使用于整体 HCSP 模型 processDef 中. 其中, 进程 Pcontrol7 定义了一个控制进程, 用于描述 Stateflow 图的语义. Pcontrol11, Pcontrol14, Pcontrol86 和 Pcontrol89 分别定义了 RBC, TCC, 车控系统和司机的行为.

```
theory controlPDef
    imports "controlADef"
begin
(*Define the processes for MATLAB fuctions.*)
definition Fcontrol1 :: proc where
"Fcontrol1 ==
e31 := e32;assSF1;
e32 := e33;assSF2;
e33 := (e33 [+] (Real 32000));assSF3;
v311 := v321;assSF4;
v312 := v322"
...
definition FMA3 :: proc where
    "FMA3 == Fcontrol1;assSF10;Fcontrol2;assSF11;Fcontrol3"
...
definition Pcontrol7 :: proc where
    "Pcontrol7 == ((num:=(Real 0);assSF92;E:=(String " ");assSF93;(a:=(Real 0)));
      assSF94;Ch_control_1_0!!a);assSF95;(Pcontrol1;assSF96;Pcontrol2;assSF97;
      Pcontrol3;assSF98;Pcontrol4;assSF99;Pcontrol5;assSF100;Pcontrol6)*"
definition Pcontrol8 :: proc where
    "Pcontrol8 == IF ((done2[=](Real 0))[&]E2[=](String "LUA"))
      (actRBC:=(Real 0);assSF101;actRBC:=(Real 1);
      assSF102;BR_2!!(String "LU");assSF103;done2:=(Real 1))"
definition Pcontrol9 :: proc where
    "Pcontrol9 ==
      IF ((done2[=](Real 0))[&]E2[=](String "MAA3"))
      (actRBC:=(Real 0);assSF104;actRBC:=(Real 1);
      assSF105;BR_2!!(String "MAA3c");assSF106;done2:=(Real 1))"
definition Pcontrol10 :: proc where
    "Pcontrol10 == done2:=(Real 0)"
definition Pcontrol11 :: proc where
    "Pcontrol11 == Pcontrol10;assSF107;
      (BC_2??E2;assSF108;(Pcontrol8;assSF109;Pcontrol9;
```

```
        assSF110;done2:=(Real 0));assSF111;BO_2!!(String ""))*"
definition Pcontrol12 :: proc where
    "Pcontrol12 == IF ((done3[=](Real 0))[&]E3[=](String "MAA2"))
      (actTCC:=(Real 0);assSF112;actTCC:=(Real 1);
      assSF113;BR_3!!(String "MAA2c");assSF114;done3:=(Real 1))"
definition Pcontrol13 :: proc where
    "Pcontrol13 == done3:=(Real 0)"
definition Pcontrol14 :: proc where
    "Pcontrol14 == Pcontrol13;assSF115;(BC_3??E3;assSF116;
      (Pcontrol12;assSF117;done3:=(Real 0));assSF118;BO_3!!(String ""))*"
...
definition Pcontrol86 :: proc where
    "Pcontrol86 == Pcontrol85;assSF355;
      (BC_1??E1;assSF356;(Pcontrol84;assSF357;
      done1:=(Real 0));assSF358;BO_1!!(String ""))*"
definition Pcontrol87 :: proc where
    "Pcontrol87 == IF ((done4[=](Real 0))[&]E4[=](String "CONFR"))
      (actDriver:=(Real 0);assSF359;actDriver:=(Real 1);
      assSF360;BR_4!!(String "CONF");assSF361;done4:=(Real 1))"
definition Pcontrol88 :: proc where
    "Pcontrol88 == done4:=(Real 0)"
definition Pcontrol89 :: proc where
    "Pcontrol89 == Pcontrol88;assSF362;(BC_4??E4;assSF363;
    (Pcontrol87;assSF364;done4:=(Real 0));assSF365;BO_4!!(String ""))*"
definition Pcontrol :: proc where
    "Pcontrol == Pcontrol7||Pcontrol11||Pcontrol14||Pcontrol86||Pcontrol89"
end
```

5.2 形式验证

在 HHL Prover 中, 我们证明的结论如下:

lemma goal: {T,T,T,T,T,T} P

{plant_s_1 ⩽ 4000,T,T,T,T,T; (l = 0) | (high (plant_s_1⩽4000)),T,T, T,T,T}.

从该结论中, 我们可以得出火车的位置 plant_s_1 始终不能超过 4000 米, 即火车会停车.

由于文件 assertionDef.thy 和 controlADef.thy 仅提供插入断言以辅助证明, 我们在以下介绍中将忽略这两个文件. 以下为证明文件的主要结构:

```
(*Goal for the whole process.*)
lemma goal : "WTrue,WTrue,WTrue,WTrue,WTrue,WTrue P
    {(plant_s_1[<=](Real 4000)),WTrue,WTrue,WTrue,WTrue,WTrue;
    (l [=] Real 0) [||] (high (plant_s_1[<=](Real 4000))),
    WTrue,WTrue,WTrue,WTrue,WTrue}"
```

证明过程分为以下三步:

(1) 将证明目标划分为初始化证明过程和重复计算两个过程;

(2) 证明初始化过程满足性质要求:

重复使用并发规则, 顺序组合规则和赋值语句规则证明该性质.

(3) 证明重复过程满足性质要求:

(i) 使用重复语句 (repetition statement) 的规则消去证明目标中的重复部分.

(ii) 使用并发通讯规则消去证明目标中的通讯.

(iii) 集中证明剩余的算术性质, 该部分确保火车不会超过目标点.

由于版面所限, 我们仅给出步骤 1 的相关代码, 全部代码见 "http://lcs.ios.ac.cn/ zoul/casestudies/psco.rar".

```
apply (simp add: P_def)
apply (simp add: PC1_def Pcontrol_def)
apply (simp add: Pcontrol7_def Pcontrol11_def
    Pcontrol14_def Pcontrol86_def Pcontrol89_def)
apply (simp add: assertion4_def assSF95_def
    assSF107_def assSF115_def assSF355_def assSF362_def)
apply (cut_tac Ha="HisP1"and Hb="WTrue"and Hc="WTrue"
    and Hd="WTrue" and He="WTrue" and Hf="WTrue" in ParallelSeq6, auto)
defer defer
apply (simp add: HisP1_def)
apply (rule impR, rule LL3a, rule basic)
apply (rule Trans,auto)+
defer
```

6 总结与未来的工作

在这篇文章中, 我们提出一套高速铁路控制系统图形建模、仿真、形式建模与验证的方法, 包括: 使用 Simulink/Stateflow 对高速铁路列控系统的进行图形建化 Simulink/Stateflow 图形模型转换成 HCSP 形式模型; 利用 HHL 定理证明器进行形式验证. 本文以行车许可、等级升级及部分模式转换场景为例说明上述方法的有效性. 特别通过上述方法, 我们发现并验证原来规范中存在的一些错误. 以上工作将有助于指导改进高速铁路列控系统规范.

今后我们希望在该模型中加入 RBC 切换等场景, 使模型更加完善, 从而可以对更加复杂的高速铁路列控系统运行情况进行仿真, 并进而能够对所有的可能存在问题的场景进行形式验证. 并且会进一步完善验证工具, 特别是利用 Simulink/Stateflow 模块定制一个面向高速列车控制系统的图形化建模语言.

参考文献

1 The Mathworks. Simulink User's Guide Version 7, 2009

2 The Mathworks. Stateflow User's Guide Version 6, 2006

3 Zou L, Zhan N J, Wang S L, et al. Verifying simulink diagrams via a hybrid hoare logic prover. In: Proceedings of EMSOFT 2013, Washington, 2013. 1–10

4 Zou L, Zhan N J, Wang S L, et al. Formal Verification of Simulink/Stateflow Diagrams. SKLCS of ISCAS Technical Report ISCAS-SKLCS-13-07. 2013
5 He J F. From CSP to hybrid systems. In: Roscoe A W, ed. A Classical Mind. Hertfordshire: Prentice Hall International (UK) Ltd., 1994. 171–189
6 Zhou C C, Wang J, Ravn A P. A Formal Description of Hybrid Systems. In: Proceedings of Hybrid Systems III, Lecture Notes in Computer Science 1066. Berlin: Springer, 1996. 511–530
7 Liu J, Lü J D, Quan Z, et al. A calculus for hybrid CSP. In: Proceedings of APLAS 2010, Lecture Notes in Computer Science 6461. Berlin: Springer, 2010. 1–15
8 Zou L, Lü J D, Wang S L, et al. Verifying Chinese Train Control System Under a Combined Scenario by Theorem Proving. In: Proceedings of VSTTE 2013, Lecture Notes in Computer Science 8164. Berlin: Springer, 2013. 262–280
9 Mattsson S E, Elmqvist H, Otter M. Physical system modeling with modelica. Control Eng Pract, 1998, 6: 501–510
10 Berkenkötter K, Bisanz S, Hannemann U, et al. The HybridUML profile for UML 2.0. Int J Softw Tools Technol Tran, 2006, 8: 167–176
11 Alur R, Dill D L. A theory of timed automata. Theor Comput Sci, 1994, 126: 183–235
12 Henzinger T A. The theory of hybrid automata. In: Proceedings of LICS 1996, Washington, 1996. 278–292
13 Berry G, Gonthier G. The esterel synchronous programming language: design, semantics, implementation. Sci Comput Program, 1992, 19: 87–152
14 Berry G. Synchronous design and verification of critical embedded systems using SCADE and Esterel. In: Proceedings of FMICS 2007, Lecture Notes in Computer Science 4916. Berlin: Springer, 2007. 2
15 Platzer A, Quesel J D. European Train Control System: a Case Study in Formal Verification. In: Proceedings of ICFEM 2009, Lecture Notes in Computer Science 5688. Berlin: Springer, 2009. 246–265
16 Tang T, Gao C H. Analysis of ETCS and CTCS. Electr D Locomot, 2005, 6: 1–3 [唐涛, 郜春海. ETCS 系统分析及 CTCS 的研究. 机车电传动, 2005, 6: 1–3]
17 Xu T H, Zhao H L, Tang T. Reliability analysis of wireless communication of ETCS using colored Petri Net. Rail Sci, 2008, 30: 38–42 [徐田华, 赵红礼, 唐涛. 基于有色 Petri 网的 ETCS 无线通信可靠性分析. 铁道学报, 2008, 30: 38–42]
18 Zhang S G. CTCS-3 Technology Specification. Beijing: China Railway Publishing House, 2008 [张曙光. CTCS-3 级列控系统总体技术方案. 北京: 中国铁道出版社, 2008]
19 Zhou C C, Hoare C A R, Ravn A P. A calculus of durations. Inform Process Lett, 1991, 40: 269–276
20 Zhou C C, Hansen M R. Duration Calculus: A Formal Approach to Real-Time Systems. Berlin: Springer, 2004

Formal analysis and verification of Chinese train control system

GUO DanQing[1], LÜ JiDong[2], WANG ShuLing[1], TANG Tao[3], ZHAN NaiJun[1*], ZHOU DaTian[2] & ZOU Liang[1]

1 *State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing* 100190*, China*;
2 *National Engineering Research Center of Rail Transportation Operation and Control Systems, Beijing* 100190*, China*;
3 *China State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing* 100190*, China*
*E-mail: znj@ios.ac.cn

Abstract It is essential that the correctness of high-speed train control systems be guaranteed by formal verification because they are safety-critical. However, as these systems become increasingly more complicated, achieving this goal in practice is difficult, and may even be infeasible. On the other hand, it is more convenient to graphically model a complicated system. A graphical model is fairly intuitive, which has resulted in such models being used extensively in industry. Thus, to improve the reliability of a high-speed train control system, constructing a

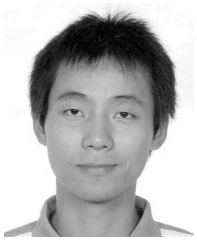
graphical model for the system and then detecting its bugs by simulation should be very effective. In this paper, we first show how to use Simulink/Stateflow to build graphical models for various combined scenarios of Chinese train control systems (CTCS), in which mode conversion and level upgrade take place simultaneously. This modeling approach can be easily adapted to model other scenarios in CTCS by simply modifying the corresponding parameters. Then, we analyze these graphical models via simulations and show that under some circumstances the trains will stop abnormally. Finally, in order to avoid the inherent incompleteness of simulation, we show how to translate these graphical models into formal models given in HCSP-a formal modeling language for hybrid systems that extends CSP-and subsequently formally prove that abnormal stops can happen in many of the cases in one of the combined scenarios. Formal verification of Simulink/Stateflow diagrams complements simulation and improves the reliability of systems being developed.

Keywords Chinese train control system (CTCS), simulink/stateflow, simulation, mode transition, level change, formal verification

GUO DanQing was born in 1989. She received a B.S. degree in software engineering from Beijing University of Posts and Telecommunications in 2011. Currently, she is a master student at the Institute of Software, Chinese Academy of Sciences. Her research interests include modeling and verification of hybrid systems.

ZHAN NaiJun was born in 1971. He received a Ph.D. in computer science from the Institute of Software, Chinese Academy of Sciences in 2000. He is currently a research professor in the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences. His research interests include formal techniques for design of real-time and hybrid systems, program verification, concurrent computation models, modal and temporal logics, and object and component systems. Prof. ZHAN is a member of ACM, a senior member of CCF, and a member of the editorial board of the Journal of Computer Research and Development.

LÜ JiDong was born in 1981. He received a Ph.D. from Beijing Jiaotong University in 2011. He is currently a lecturer at Beijing Jiaotong University. His research interests include formal modeling and verification of hybrid system, and model-based test case generation of train control systems.



ZOU Liang was born in 1986. He received a B.S. degree from Central China University of Science and Technology in 2009. He is currently a Ph.D. candidate at the Institute of Software, Chinese Academy of Sciences. His research interests include hybrid systems, formal verification, language transformation, and statistical model checking.

征 稿 简 则

简介：《中国科学: 信息科学》(英文名称: SCIENTIA SINICA Informationis) 是中国科学院和国家自然科学基金委员会共同主办、《中国科学》杂志社出版的学术刊物. 本刊力求刊载信息科学领域最高学术水平的中文文章, 及时报道计算机科学与技术、控制科学与控制工程、通信与信息系统、电子科学与技术等领域基础与应用研究方面的原创性成果, 推动信息科学技术发展, 搭建理论与技术应用的桥梁, 促进与各学科、各行业的交叉融合. 月刊, 每月 20 日出版.

收录情况：《中国科学: 信息科学》被《中文核心期刊要目总览》、《中国科学引文数据库》、《中国期刊全文数据库》、《中国科技论文与引文数据库》等收录.

栏目：《中国科学: 信息科学》设有以下 4 个栏目.

评述: 综述信息科学领域的代表性研究成果和最新进展, 提出作者的独到见解和未来的研究方向. 长度在 20 页左右. 一般由编委邀请, 有意撰写评述的专家也可向有关编委提议.

论文: 重点发表报道最新研究成果及其科学思想、意义、创新点、实验和理论依据及应用前景的概述性论文, 长度不超过 15 页.

快报: 简要介绍信息科学领域最新研究成果的核心内容. 长度不超过 4 页. 优先发表.

学术介绍: 主要介绍从事信息科学研究的院校、实验室和一些重大的研究课题、研究成果及其核心产品.

投稿：请使用在线投稿系统投稿. 访问本刊网站 www.scichina.com 或 info.scichina.com, 注册一个“作者账户”, 按照提示填写投稿信息并将稿件全文(PDF 格式文件)上传到数据库服务器. 如果不能在线投稿, 请与编辑部联系. 本刊受理的稿件要求用 LaTeX 排版, 模板可从本刊网站下载. 作者在投稿时只能选择一个语种投稿, 且在评审录用后, 不能再翻译成另一语种发表. 请您在投稿时注意, 认真阅读本刊投稿指南, 选择好刊物.

审稿：稿件由主编负责组织编委和审稿专家进行评审, 并根据评审意见确定录用与否. 评审结束后, 编辑部将及时向作者转达评审意见和结果, 作者若在 90 天内没有收到编辑部有关稿件的具体意见, 在通知编辑部后, 可改投他刊. 本刊不受理“一稿多投”之稿件.

文章署名：通讯作者应保证稿件内容经全体作者认可并同意署名. 投稿后, 任何署名的改变要有全体原作者签名同意的书面材料.

录用：稿件录用后, 全体作者应当签署“著作权转让声明书”, 将该论文(各种语言版本)的复制权、发行权、信息网络传播权、翻译权、汇编权在全世界范围内转让给《中国科学: 信息科学》的出版单位《中国科学》杂志社.

出版：本刊对录用的稿件收取版面费, 出版后向作者免费提供一本样刊. 作者可以购买抽印本和更多的期刊. 作者可以在本刊网站上免费注册下载本刊论文.

地　　址：北京东黄城根北街 16 号　　**邮政编码**：100717

电　　话：(010) 64015683 (编辑部)　　**电子信箱**：informatics@scichina.org (编辑部)

(010) 64019709 (发行部)　　sales@scichina.org (发行部)

(010) 64008316 (广告部)　　ads@scichina.org (广告部)

传　　真：(010) 64016350

中国科学 信息科学

SCIENTIA SINICA Informationis

第 45 卷　第 3 期　2015 年 3 月出版

主　　管	中国科学院	**出　　版**	《中国科学》杂志社
编　　辑	中国科学院《中国科学》编辑委员会	**印刷装订**	北京中科印刷有限公司
		总发行处	北京报刊发行局
主　　编	李　未	**订 购 处**	全国各邮电局. 《中国科学》杂志社发行部

刊号: ISSN 1674-7267 / CN 11-5846/TP　　代号: 国外 M568 / 国内邮发 80-948　　每期定价: 145.00 元　全年定价: 1740.00 元

广告经营许可证: 京东工商广字第 0429 号

主管：中国科学院 | 主办：中国科学院 国家自然科学基金委员会 | 主编：李未 | 出版：《中国科学》杂志社 SCIENCE CHINA PRESS

ISSN 1674-7267
CN 11-5846/TP

中国科学：信息科学 （月刊）
SCIENTIA SINICA Informationis

定位：发表信息领域最高学术水平的中文文章，包括计算机科学与技术、控制科学与控制工程、通信与信息系统、电子科学与技术、生物信息学等领域的理论、工程技术和应用研究方面的原创性成果. 推动信息科学技术发展，搭建理论与技术应用的桥梁，促进与各学科、各行业的交叉融合. 月刊，每月20日出版.

栏目：评述、论文、快报、学术介绍.

检索：被《中文核心期刊要目总览》、《中国科学引文数据库》、《中国期刊全文数据库》、《中国科技论文与引文数据库》、《中国数字化期刊群》等收录.

[曾用名] 中国科学 F辑：信息科学 SCIENCE CHINA Series F: Information Sciences

ISSN 1674-733X
CN 11-5847/TP

SCIENCE CHINA Information Sciences (Monthly)

SCIENCE CHINA Information Sciences is a peer-reviewed monthly academic journal supervised by the Chinese Academy of Sciences, and co-sponsored by the Chinese Academy of Sciences and the National Natural Science Foundation of China. Its primary mission is to encourage communication of basic and innovative research results of high quality in the fields of information sciences. The subject areas featured include computer science and technology, control science and technology, communication and information system, electronic science and technology, and bioinformation, etc. All papers should be intelligible for a broad scientific audience. Contributions are invited from researchers all over the world.

Papers published in ***SCIENCE CHINA Information Sciences*** include: Review, Research Paper, MOO Paper, Short Paper, Letter.

It is indexed by Academic OneFile, Astrophysics Data System (ADS), CSA, Cabells, Current Contents/Engineering, Computing and Technology, DBLP, Digital Mathematics Registry, Earthquake Engineering Abstracts, Engineering Index, Engineered Materials Abstracts, Gale, Google, INSPEC, Journal Citation Reports/Science Edition, Mathematical Reviews, OCLC, ProQuest, SCOPUS, Science Citation Index Expanded, Summon by Serial Solutions, VINITI, Zentralblatt MATH.

《中国科学: 信息科学》编辑部
地址: 北京东黄城根北街16号 (100717)
电话: 010-64015683 传真: 010-64016350
E-mail: informatics@scichina.org

ISSN 1674-7267
03
9 771674 726152