Software Engineering Group
Department of Computer Science
Nanjing University
http://seg.nju.edu.cn

Technical Report No. NJU-SEG-2014-CJ-003

混成系统形式化验证

卜磊，解定宝

Postprint Version. Originally Published in: 软件学报，2014,25(2):219-233

[doi: 10.13328/j.cnki.jos.004535]

软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2014,25(2):219−233 [doi: 10.13328/j.cnki.jos.004535]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

混成系统形式化验证*

卜 磊[1,2], 解定宝[1,2]

[1](南京大学 计算机科学与技术系,江苏 南京 210023)

[2](计算机软件新技术国家重点实验室(南京大学),江苏 南京 210023)

通讯作者: 卜磊, E-mail: bulei@nju.edu.cn

摘 要: 混成系统是实时嵌入式系统的一种重要子类,其行为中广泛存在离散控制逻辑跳转与连续实时行为交织混杂的情况,因此行为复杂,难以掌握与控制.由于此类系统广泛出现在工控、国防、交通等与国计民生密切相关的安全攸关的领域,因此,如何对相关系统进行有效的分析与理解,从而保障系统安全运营,是一项具有重要意义的工作.常规的系统安全性分析手段,如测试、仿真等仅能在一定输入的情况下运行系统来观测系统行为,无法穷尽地检测复杂混成系统在所有可能输入下的行为,因此并不足以保证系统的安全性.区别于测试等方法,形式化方法通过求解系统模型状态取值范围等方法来确认系统模型中一定不会出现相关错误.因此,其对于保障安全攸关混成系统的安全性具有十分重要的意义.形式化方法由形式化规约与形式化验证两个方面构成.因此从以上两个角度分别对形式化规约方向上现有混成系统建模语言、关注性质以及形式化验证方向的混成系统模型检验、定理证明的现有主要技术与方法进行了综述性的回顾与总结.在此基础上,针对现阶段实时嵌入式系统复杂化、网络化的特性,对混成系统形式化验证的重要关注问题与研究方向进行了探索与讨论.

关键词: 混成系统;形式化方法;模型检验;定理证明

中图法分类号: TP311 **文献标识码**: A

中文引用格式: 卜磊,解定宝.混成系统形式化验证.软件学报,2014,25(2):219−233. http://www.jos.org.cn/1000-9825/4535.htm

英文引用格式: Bu L, Xie DB. Formal verification of hybrid system. Ruan Jian Xue Bao/Journal of Software, 2014,25(2): 219−233 (in Chinese). http://www.jos.org.cn/1000-9825/4535.htm

Formal Verification of Hybrid System

BU Lei[1,2], XIE Ding-Bao[1,2]

[1](Department of Computer Science and Technology, Nanjing University, Nanjing 210023, China)

[2](State Key Labotary for Novel Software Technology (Nanjing University), Nanjing 210023, China)

Corresponding author: BU Lei, E-mail: bulei@nju.edu.cn

Abstract: Hybrid System is a very important subclass of real time embedded system. The behavior of hybrid system is tangled with discrete control mode transformation and continuous real time behavior, therefore very complex and difficult to control. As hybrid system is widely used in safety-critical areas like industry, defense and transportation system, it is very important to analyze and understand the system effectively to guarantee the safe operation of the system. Ordinary techniques like testing and simulation can only observe the behavior of the system under given input. As they cannot exhaust all the possible inputs and scenarios, they are not enough to guarantee the safety of the system. In contrast to testing based techniques, formal method can answer questions like if the system will never violate certain specification by traversing the complete state space of the system. Therefore, it is very important to pursue the direction of formal verification of safety-critical hybrid system. Formal method consists of formal specification and formal verification. This paper reviews

* 基金项目: 国家自然科学基金(61100036, 91318301, 61321491); 国家高技术研究发展计划(863)(2012AA011205); 江苏省自然科学基金(BK2011558)

收稿时间: 2013-05-07; 定稿时间: 2013-09-29

the modeling language and specification of hybrid system as well as techniques in model checking and theorem proving. In addition, it discusses the potential future directions in the related area.

Key words: hybrid system; formal method; model checking; theorem proving

1 混成系统简介

在实时嵌入式系统,特别是复杂的实时控制系统中,广泛存在这样一类子系统,它们的行为中离散化的逻辑控制与连续性的时间性行为相互依赖,相互影响,彼此互为依存,息息相关.以列车控制系统为例,典型的列车控制系统一般存在多种不同的控制模式来应对当前的车况、路况以及各种突发事件,而系统中的重要参数,如列车速度、当前位置、与前车距离等等,会随着时间连续变化.列车在运行中会为了满足特定的时间约束或者调整当前参数的取值而调整列控模式,而在不同的列控模式下,列车中重要参数的变化规律完全不同,相应地,对各种事件的响应时间也会有所区别.在这种类似的系统中,逻辑控制与时间性行为并不是孤立的两个部分,而是交错地有机结合在一起,构成了一种非常复杂的系统,而这样的复杂实时系统因为其离散控制与实时连续行为混杂叠加的特性,一般被称为混成系统(hybrid system)[1].

混成系统是一种嵌入在物理环境下的实时系统,一般由离散组件和连续组件连接组成,组件之间的行为由计算模型进行控制.对于经典混成系统,其构成体现了计算机科学和控制理论的交叉,一般分为两个层面:离散层与连续层.在连续层,一般通过系统变量对时间的微分方程来描述系统的实际控制操作模型以及系统中参数的演变规律;在离散层,则通过状态机、Petri 网等高抽象层次的模型来描述系统的逻辑控制转换过程.在两层之间,通过一定的接口与规则将连续层的信号与离散层的控制模式进行关联与转换.

大多数复杂实时控制系统行为都包含了连续变化的物理层与离散变化的决策控制层之间的交互过程,因此,混成系统在工业控制和国防等领域大量存在,特别是安全关键系统,如交通运输、航空航天、医疗卫生、工业控制等等系统.相应地,随着它们在人类生活中的应用面越来越广,重要性越来越高,对相应系统质量,特别是可信性的需求也快速提升,系统失效所造成的灾难也越来越沉重,甚至难以接受.在日常生活方面,车载导航系统的小小失误就可能造成交通事故,而飞机导航系统的失误则可能导致机毁人亡.如果扩展到国防军用领域,对软件系统的错误已经几乎进入了零容忍度的阶段.因此,如何对混成系统进行有效的可信性保障,已成为一个亟待解决的问题.

一般而言,测试、仿真[2,3]等技术是研究和保障软件质量的主要方法.然而,这些方法主要以运行系统为发现问题的主要手段,由于人力无法穷尽地遍历系统所有可能的运行输入与场景,也就不足以保证检测的完备性,这也就可能给系统后期运行留下了不安全隐患.因此,在对系统错误零容忍的安全攸关系统领域,采用可证明系统模型正确性的形式化验证理论与技术[4,5]来对系统模型进行安全性验证就显得极为重要,这也成为相关领域近期主要关注的问题.

形式化方法(formal method)是对以数学为基础的,用以对系统进行说明、设计和验证的语言、技术和工具的总称,其主要可以分为形式化规约(specification)与形式化验证(formal verification)两个方面.形式化规约就是用形式化语言在不同抽象层次上描述部分或整个系统的行为与性质.一般而言,我们称表示系统性质的语言为规约语言(specification language),比如各种时序逻辑(temporal logic)等;我们称描述系统行为的数学模型为系统刻画语言(system description language),如 CSP[6]、状态图[7]等.在我们描述了系统的行为与需要满足的性质之后,就需要采用形式化验证来判定最终的软件产品是否满足这些需求和具备这些特征.通过验证,可以判定系统是否满足某个特定的性质,并在系统不满足性质时给出理由.目前,对软硬件系统的验证主要包括模型检验(model checking)[5]和定理验证(theorem proving)[8,9]两个方面.

对应地,在混成系统形式化建模与验证方向的研究也主要在上述方向上展开,例如,如何设计具有足够表达能力的建模语言来描述混成系统中的复杂行为;如何设计有效的模型检验方法、定理证明方法来对大规模复杂系统进行有效的验证,回答系统是否满足特定性质的问题.在近 20 年的密集投入与研究之后,混成系统的形式化建模与验证已经取得了一系列成果[10].本文将对相关主要研究方向与成果进行概述性的总结与回顾.在此基

础上,针对现阶段实时嵌入式系统所呈现的智能化、开放化等特征,本文对混成系统近期的研究热点以及下一阶段重点关注的问题进行了探讨与展望.

2 混成系统建模语言

针对混成系统行为中离散逻辑跳转与连续时间行为交织的特征,相关科研工作者对状态机、CSP 等建模语言进行了实时变量定义、微分方程连续行为等扩展,提出了包括混成自动机(hybrid automata)[1]、混成 CSP[11,12]、混成 Petri 网[13]、混成程序[14]等在内的一系列形式化建模语言.尽管以上语言之间各有侧重与不同,但是在如何表达系统中的离散与连续行为交织等特性方面,以上语言的扩展之间存在着大量的共通之处.在上述语言当中,混成自动机得到了最为广泛的认可与应用.下面我们就以混成自动机为例来说明相关建模语言是如何针对混成系统的相关特性进行建模与描述的.

混成自动机是在状态机基础上进行实时连续变量扩展所构成的一种建模语言.可以定义混成自动机为多元组 $H=(X,\Sigma,V,E,V^0,\alpha,\beta,\gamma)$,其中,

- X 是实数值系统变量的有限集合,X 中变量的个数也被称为自动机的维度.
- Σ是事件名的有限集合,V 是位置节点的有限集合.
- E 是转化关系的集合,E 中的元素 e 具有形式$(v,\sigma,\varphi,\psi,v')$.其中,$v,v'$是 V 中的元素;$\sigma\in\Sigma$是转换上的事件名;转换卫式φ是一个将 E 中的转换 e 标注为一组约束的标注函数,表示当系统行为触发转换 e 时,相应变量的取值满足此约束;ψ是形为 $x:=c$ 的重置动作集合,表示当系统行为触发此转换后,相应变量 x 的取值会被重置为 c,以上 $c\in R,x\in X$.
- $V^0\subseteq V$ 是初始位置的集合.
- α是一个标注函数,它将每个位置映射到一个节点不变式,表示系统行为停留在相关节点时,相应变量取值满足此约束.
- β是一个为 V 中每个位置节点添加流条件(微分方程)的标注函数,表示当系统行为停留在相关节点时,相应变量取值变化随着时间增长满足此条件.对任意 $x\in X$,有且仅有 1 个 x 的流条件属于$\beta(v)$.
- γ是一个标注函数,它将初始位置 V^0 中每个位置映射到一组初始条件,初始条件具有形式 $x:=a(x\in X, a\in R)$.对任意位置 $v\in V^0$,对任意 $x\in X$,有且仅有 1 个 $x:=a\in\gamma(v)$.

图 1 是一个经典的自动温度控制器模型[1],我们用此模型来对混成自动机及其各个组成部件进行一个简要的描述.此模型中,变量 x 描述的是系统中实时变化的温度数值.当系统驻留在控制模式 off 时,加热器被关闭,环境中的温度按照 off 节点上的流条件 $\dot{x}=-0.1x$ 下降(可理解为微分方程 $dx/dt=-0.1x$);而当系统驻留在控制模式 on 时,加热器被打开,环境中的温度按照 on 节点上的流条件 $\dot{x}=5-0.1x$ 上升.系统的初始条件被设定为温度 20°C,控制模式为 off.转换卫式 $x<19$ 与 $x>21$ 表示当系统温度降低到 19°C 以下时,控制模式就可以从 off 切换到 on,从而打开加热器;而当系统温度高于 21°C 时,则正好相反,控制模式可以跳转到 off 模式,从而关闭加热器.最后,在此模型中分别存在两个不变式:$x\geq 18$ 与 $x\leq 22$,这表明了系统停留在控制模式 off 和 on 时,其实时变量 x 的合法取值范围.

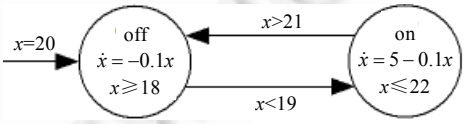


Fig.1 Hybrid automata for temperature comtroller

图 1 温度控制器混成自动机模型

显然,如果忽略变量 x 以及相关的不变式、转换卫式、流条件等元素,这个混成自动机的结构就是一个基本的状态机图.通过将状态机图中的状态节点概念拓展到位置节点(可理解为控制模式),并在每个位置节点上添加相应的连续变量变化规则,可以描述系统在不同控制模式下的实时参数变化过程,从而描述混成系统的具体

连续行为.

由于混成自动机中连续行为与离散行为共存的特质,混成自动机的行为非常复杂,难以控制与把握.因此,现在相关研究领域主要关注于其中一个比较特别的子类——线性混成自动机(linear hybrid automata).给定一个变量集合 X,我们称表达式 $\sum_{i=0}^{l} c_i x_i \sim b$ 为线性表达式(linear term),其中,$c_i \in R, x_i \in X, \sim \in \{>,<,=,\leqslant,\geqslant\}, b \in R$.我们称一组线性表达式的布尔组合(Boolean combination)为一个线性公式(linear formula).给定一个混成自动机 H 满足下列条件,我们称其为线性混成自动机(linear hybrid automata):

- 在任意控制转换 $e \in E$ 上,转换卫式 φ 中任一约束均为线性公式;
- 对任意控制位置 $v \in V$,变量 x 在 $\alpha(v)$ 中的定义均为线性公式;
- 流条件是形如 $\dot{x} \in [a,b]$ 或者 $\dot{x} \sim a$ 的变化率集合,其中,$x \in X, a,b \in R, a \leqslant b, \sim \in \{>,<,=,\leqslant,\geqslant\}$.

如果将图 1 中的温度控制器模型的流条件进行简单的转换,我们就可以获得一个线性混成自动机版本的温度控制器模型,如图 2 所示.

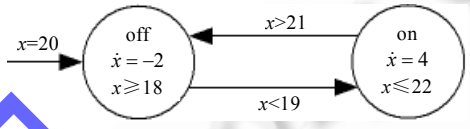


Fig.2 Linear hybrid automata for temperature comtroller

图 2 温度控制器线性混成自动机模型

线性混成自动机是混成自动机中的一种比较重要的子类.众所周知,线性系统的复杂度远低于非线性系统,并且现有数学技术在线性系统领域已经颇为成熟,可以处理相当大规模的问题空间;而在非线性系统上,现有的数学技术可以处理的问题空间非常有限,远远达不到实际应用的需求.因此,通过线性表达式来描述流条件、不变式、跳转条件等部件,可以大幅度降低系统的复杂度,并且使设计者更容易把握系统的行为,保证系统的正确性.尽管实际应用中的主要系统大部分需要使用非线性控制,特别是流条件部分,无法直接应用线性混成自动机来对系统建模或者描述,但是设计者可以通过抽象(abstraction)的方法拆分原系统的行为,使用一个包含更多控制节点的线性混成自动机模型来逼近非线性自动机的行为,并逐步逼近直到该线性混成自动机的精度可以在最大程度上拟合原非线性系统,从而通过对该线性混成自动机进行分析的方法达到分析原系统的目的[15].

事实上,通过在标准线性混成自动机的基础上进一步添加相应的约束与限制,我们可以将其转化成一些非常重要的子类乃至于读者相对更加熟悉的建模语言,如下所示[16]:

- 如果对任意控制位置 $v \in V$,变量 x 在 $\beta(v)$ 的定义中均形如 $\dot{x}=0$,即,变量 x 在所有节点上的变化率均为 0,则称 x 为一个离散变量(discrete variable).如果一个线性混成自动机中所有变量均为离散变量,则称此自动机为离散系统(discrete system).
- 如果对任意控制转换 $e \in E$,离散变量 $x \in X$ 在 ψ 中均形如 $x:=0$ 或者 $x:=1$,即,变量 x 在系统触发每个跳转之后的新取值必为 0 或者 1,则称此变量为一个命题(proposition).如果一个线性混成自动机中所有变量均为命题,则称此自动机为一个有穷状态系统(finite-state system).
- 如果对任意控制位置 $v \in V$,变量 x 在 $\beta(v)$ 的定义中均形如 $\dot{x}=1$,即,变量 x 在所有节点上的变化率均为 1;并且如果对任意控制转换 $e \in E$,离散变量 $x \in X$ 在 ψ 中均形如 $x:=0$ 或者 $x:=x$,即,系统触发每个跳转之后会将变量 x 的值赋值为 0 或者不变,则称变量 x 为一个时钟(clock).如果一个线性混成自动机中所有变量均是命题或者时钟,并且系统内所有线性公式均为形如 $x \sim c$ 或者 $x-y \sim c$ 的线性表达式的布尔组合,其中,$x,y \in X, \sim \in \{>,<,=,\leqslant,\geqslant\}, c \in \mathbf{Z}^{\geqslant 0}$,则称此线性混成自动机为时间自动机(timed automata)[17].
- 如果存在非零整数 $k \in \mathbf{Z}$,并且对任意位置节点 $v \in V$,变量 x 在 $\beta(v)$ 的定义中均形如 $\dot{x}=k$,与上类似,如果对任意控制转换 $e \in E$,变量 $x \in X$ 在 ψ 中均形如 $x:=0$ 或者 $x:=x$,则称变量 x 为一个倾斜时钟(skewed clock),即,此变量在每个节点都按照一个不为 1 的固定变化率进行变化.如果一个线性混成自动机中每个变量均为命题或者倾斜时钟,则称此自动机为多级时间系统(multirate timed system).如果一个多级时间

系统中变量的变化率共有 n 种,则称此系统为 n 级时间系统(n-rate timed system).

- 如果对任意位置 $v\in V$,变量 x 在$\beta(v)$的定义中均形如 $\dot{x}=0$ 或者 $\dot{x}=1$,即,变量 x 在所有节点上的变化率不是 0 就是 1;并且如果对任意控制转换 $e\in E$,变量 $x\in X$ 在ψ中均形如 $x:=0$ 或者 $x:=x$,则称 x 为一个积分器(integrator).如果一个线性混成自动机中所有变量均为命题或者积分器,则称此自动机为积分器系统(integrator system).

3 混成系统模型检验

模型检验是通过穷尽遍历待验证软件系统模型的状态空间来检验系统的行为是否具备预期性质的一种自动验证方法.自 20 世纪 80 年代被提出以来,模型检验技术得到了学术界、工业界的广泛重视,并在芯片设计等领域得到了广泛应用.

如上文所述,混成自动机是混成系统最主流建模语言.目前,学术界对混成自动机模型检验研究主要集中在混成自动机的安全性性质(safety)[18,19]检验上.安全的直观概念解读为系统运行中不会发生不好的行为.安全性检验即检验从给定的初始条件出发,系统运行中是否会出现违背规约或者不安全的行为.由于混成自动机的运行既包含状态的离散变化,又包含状态的连续变化,其相应的模型检验问题十分困难.因此,目前对混成自动机的模型检验主要集中在安全性的一个子集——可达性问题(reachability)上.

对于混成自动机 H,它的可达性规约由 H 中的一个位置节点 v 和一个变量约束集ϕ组成,我们通过符号 $R(v,\phi)$来表示.H 满足 $R(v,\phi)$可定义为:H 存在一个具体的实时行为可进入节点 v,并且当系统停留在节点 v 中一段时间后,系统中实时变量 x 取值可满足ϕ中的所有变量约束.

可达性问题与安全性问题的关系在于:将系统不安全的行为构建成一个独立的离散“坏”状态,然后检验此“坏”状态是否可达:如可达,则系统中可能发生不安全行为;反之,则系统行为安全.因此,检验系统安全性的问题就变成了计算系统可达空间的问题.也就是说,我们要计算出系统的所有可达状态,并判定这些状态中是否有状态可以使规约 $R(v,\phi)$成真.由于混成自动机中连续行为的存在,混成自动机拥有极为庞大的无限状态空间,所以我们不能像一般模型检验方法一样通过直接枚举遍历的方法来计算出整个可达集,而是必须通过符号化的方法来计算.目前,主流方法是通过一定的约束集来描述系统初始状态集合,并称其为系统初始可达空间.然后,基于系统的流条件、不变式、转换卫式等元素定义相应自动机上的 $Post$ 操作,以计算系统在当前状态集下后续可达状态集区间.将计算得到的状态集并入已有可达状态集空间并重复上述过程,直到系统可达状态空间集收敛,即,从当前所有可达状态空间上基于系统规定的连续/离散演变规则不再能抵达新的状态.此时,称所计算得到的状态空间为系统完整可达状态空间集,并进行判断计算所得系统完整可达状态空间集中是否有状态满足规约 $R(v,\varphi)$:如有,则称系统满足相应可达性规约;否则称系统不满足相应规约.

上述方法的可行性、有效性等等,与能否在任意的系统状态集上针对给定的微分方程等元素进行数值计算与推演从而生成后续可达状态集密切相关.众所周知,在任意形态数值状态域上进行任意形式的非线性微分方程计算复杂度极高,目前,数学领域还没有有效的方法来解决相应大规模问题.因此,针对一般混成自动机,目前可达集计算的主流方法为将系统的状态域用一种特定的数学形态来过抽象(over-approximation).目前常用的数学形态包括凸多面体(convex polyhedra)[20,21]、分段仿射系统(piecewise affine system)[22,23]、椭圆体(ellipsoidal)[24,25]等等.在使用以上数学形态对可达域进行标识之后,使用相关领域的成熟数学计算方法来求取从当前形态开始后继形态的演变范围及过程.但是,这类方法仍然存在很多问题:首先,过抽象带来的问题是状态域的表达不够精确;其次,此过程无法保证收敛,很可能一直循环而无法停止;最后,以上过抽象方法中数学计算复杂度很高,对系统资源消耗极大,从而无法对高维度复杂系统进行分析.实际上,即使是针对混成自动机中相对简单的子类,线性混成自动机,其可达性问题也已被证明是不可判定的[1,16,26,27].

由于以上原因,相关方法在一般性非线性混成自动机上表现并不如人意.特别是由于非线性计算的高度复杂度,目前已有相关工具可验证非线性混成自动机模型的规模非常有限,文献[28]在对主流工具进行整体评估后得出结论为,现有工具很难验证超过 5 个变量的系统.由于线性数值域计算方法远成熟于非线性计算,复杂度

也能得到较好的控制,因此基于上述理念,相关研究人员以多面体作为线性混成自动机基本状态域的数值表现形式,开发了一系列线性混成自动机模型检验工具,如 HyTech[29],PHAVer[30]等,并成功地验证了飞机防撞系统等典型混成自动机案例.

值得注意的是,上述线性混成自动机特指前文中所描述的流条件形如 $\dot{x}\in[a,b]$ 的自动机形式,称其为线性混成自动机是因为相应流条件积分展开后可以得到变量 x 的取值是与时间 t 相关的线性函数.在相关研究中,存在另一类类似的混成自动机被称为线性混成系统(linear hybrid system),其流条件形如 $\dot{x}=Ax+b$,相关研究者(特别是控制领域)普遍称其为线性混成系统是因为其流条件表现为 x 的线性方程,但是其积分展开后会成为包含 e^t 的非线性方程,所以其并不是线性混成自动机.针对相应的自动机,也有大量相关工具被开发出来,其中较著名的包括来自美国卡耐基梅隆大学的 Checkmate[31]、来自法国 Verimag 实验室的 d/dt[20]等等.

目前,相关领域科研工作者在非线性混成自动机模型检验上投入了大量的精力并取得了一定的进展,特别是泰勒模型(Taylor model)、Support Function 等数学模型被应用到了混成系统状态域的表达与计算当中,近年出现的工具 flow*[32]就是一个成功应用泰勒模型对非线性混成系统进行验证的示例.flow*要求混成自动机的流条件必须由多项式微分方程描述.用户给定初始区间与一个固定的基本时间步之后,可利用泰勒模型来分析其可达区间.事实上,泰勒模型很适合连续状态的计算,但是当进行离散跳转时,需要和转换卫式做相交操作,而这个操作的复杂度很高.与 flow*不同,SpaceEx[33]可处理的系统是对线性混成系统,即,流条件为 $\dot{x}=Ax+b$,进行一定放宽后的非线性自动机.SpaceEx 允许系统中的不变式、迁移卫式等元素为凸函数.同样地,在给定一个基本时间步之后,可以利用 support function 来计算在此时间步之后的系统状态域.相关工具目前已经在部分非线性系统上进行了验证,但如何扩展可验证系统种类与规模,仍然是一项值得关注的问题.

另一方面,反例制导的抽象精化(counter example guided abstraction refinement,简称 CEGAR)[34]是一种非常有效的复杂系统形式化验证方法.其基本思想在于:在原系统因过于复杂而难以验证的情况下,对原系统进行抽象后,在简化系统上进行验证.由于系统抽象会引入一些原系统所不包含的行为,因此当进行安全性/可达性检查时,若抽象系统不满足相应规约,则原系统必然也不满足相应规约.然而当抽象系统满足相应规约时,满足此规约的可疑行为可能是由于抽象所带入,因此需要将相应的可疑行为在原系统上进行确认:若此行为在原系统上也满足,则验证结束;否则,分析此行为在抽象前后系统上的区别,从而指导下一轮精化后抽象的展开.通过循环迭代此步骤,可以有效地缩减每次待验证系统规模,从而对复杂系统进行分析.

CEGAR 思想同样也在混成系统模型检验上得到了应用.Clarke[35],Alur[36]等人提出了大量混成系统 CEGAR 验证方法来处理大规模复杂系统.相关方法提出,将状态谓词作用与混成系统状态空间上进行抽象.然而,此类方法在谓词精化中经常需要对系统状态空间进行拆分,从而大幅度增加了待验证系统的结构,难以对复杂系统开展验证.同样使用 CEGAR 思想,针对线性混成自动机,Clarke 等人提出了一种称为迭代式松弛抽象(iterative relaxation abstraction,简称 IRA)的 CEGAR 框架[37].此方法的主要思路在于:每次抽象时,选取系统中部分实时变量进行丢弃,从而保持系统图形结构,但是对系统变量空间进行降维处理.在降维后的系统上采用 PHAVer 等现有工具进行验证,若降维后系统上存在路径可满足相应可达性约束,则将此路径上的变量还原,采用南京大学团队提出的面向路径可达性验证方法[38]对原路径进行编码并验证其是否可行.若此路径在原系统中不可行,则抽取导致此路径在原系统中不可行的约束集,并基于此约束集中的变量修正下一轮系统抽象时所抛弃的变量集合的内容.此方法在典型案例上的性能明显优异于现有工具,如 PHAVer 等等,但是由于此方法中仍然依赖于 PHAVer 对抽象后系统进行验证,因此当抽象后系统仍然维度较高时,此方法也难以进行处理.

4 混成系统有界模型检验

近年来,作为基于 BDD(binary decision diagram)的符号化模型检验[39]的一种补充方法,有界模型检验(bounded model checking,简称 BMC)技术[40]被提出并得到了广泛的应用.其基本思想是:将模型行为步数通过正整数 k 来限制,将系统 k 步内行为采用布尔约束编码,然后利用 SAT 方法来寻找此布尔约束集的可行解,从而判定系统在 k 步内行为是否有不满足规约的情况.文献[40]发现,虽然 BMC 方法因为只能验证有限步数内系统

性质而缺失了一定的完整性,但是通过限制步数,其在发现错误上的能力超越了传统模型检验方法.即,当模型规模超过经典模型检验方法可检验范围而无法验证的情况下,通过限定被检验模型状态空间范围的方法来使用有界模型检验技术高效地发现限定范围内系统的问题,而这也正是 BMC 方法被广泛认可的亮点所在.

BMC 思想同样被应用到混成自动机,特别是线性混成自动机领域的可达性检验工作中.作为一种将 SAT 技术从纯布尔值域向其他多种混合值域的扩展,SMT(satisfiability modulo theories)[41]思想被提出来并在近 10 年得到了较大的突破.SMT 相关的研究者对时间自动机与线性混成自动机的有界可达性问题进行了大量的实验与分析,如文献[42−44]等等.目前,相关主要方法是通过 SMT 技术来对混成自动机在有限步数内行为编码,并通过 SMT 约束求解器及其可调用的底层域相关约束求解器来对相应约束集求解,从而寻找给定步长内可达相应目标的路径.针对线性混成自动机的特征,SMT 处理器在面向线性实数域求解时会结合 DPLL 方法以及相关的实数域经典分析方法,如 Simplex 算法、Interior-point 算法等来进行计算.但是由于此方法需要在检验前将系统 k 步内所有离散跳转与实时连续行为统一编码成一个 SMT 约束集,当问题规模,如给定步长大小、系统变量数目、自动机组合内成员数目等等增长后,约束集大小将快速增长,从而导致相应的内存需求急剧上升,进而限制了可解决问题的规模.

为了有效地控制验证的复杂度,南京大学研究团队提出了面向路径可达性验证方法,限制单次仅验证自动机图形结构上单次路径的可达性[38].相关问题可被线性编码成一组线性不等式的可满足性问题,从而用线性规划技术(LP)高效判定.在此基础上,可通过深度优先遍历(DFS)、SAT 求解等多种技术进行阈值内可疑路径枚举,从而对每条可疑路径采用上述面向路径可达性验证进行判定来实现自动机的有界模型检验[45].在此基础上,可结合前后向并行 DFS[46]、基于 IIS 技术的不满足路径分析与精化[47]等技术来加速相应遍历与验证过程.

另一方面,组合系统验证的主流方法是将各成员依据共享事件标签进行笛卡尔乘积,构造出系统组合同步后的乘积自动机再进行验证.然而,依据此方法所生成的组合自动机的规模随着成员数目的增长而急剧增加,严重抑制了可验证系统的规模,而这也正是著名的组合状态空间爆炸问题.为了应对此问题,南京大学团队从面向路径的角度出发,提出了一种线性混成自动机组合验证的浅同步语义[48],提出了线性混成自动机的一个组合路径组,即,每个成员自动机各提供一条路径的可达性问题可对各成员独立编码之后再针对路径上的同步事件添加为数不多的同步时间约束编码,从而避免昂贵的笛卡尔乘积.基于上述思想,南京大学团队开发了线性混成自动机有界可达性验证工具集 BACH[45].该工具在国际公认案例上性能优异,并成功地验证了有 300 多名成员的大规模组合系统.同时,相关团队基于浅同步组合语义也提出了一套新的线性混成自动机组合验证 SMT 编码方法,通过在 MathSAT 等知名 SMT 处理器上对主流案例进行编码并实验,显示在相关案例上性能明显优于经典强同步语义编码[49].

上述工作主要在线性混成自动机上展开.在非线性混成自动机有界模型检验上,相关工作也已经取得了一定的进展.基于区间运算方法(interval analysis)[50],德国 Oldenberg 大学的研究者开发了面向非线性混成自动机的有界模型检验工具 HySAT/iSAT[51].通过定义最小的时间步长$\delta_{\min}$,HySAT 可使用区间运算方法,根据变量取值范围,计算在相应时间步后,相关函数的取值范围.因此,HySAT 可处理包括 sin,cos 在类的多种非线性约束而不需要强制必须为多项式.但是由于区间运算中计算误差的传播性,因此取值区间在复杂情况下可能快速发散,从而影响最后验证结果的精确程度.

类似地,通过在 DPLL(T)框架上结合区间运算技术,美国卡耐基梅隆大学的相关研究人员在近年来提出了一种δ-complete 判定准则[52]来对实数域上的非线性函数进行判定.与传统的 SMT,SAT,LP 等方法针对特定问题给出可满足或者不可满足结论不同,此判定方法为待判定函数给出不满足或者δ可满足两种结果,其中,δ可满足之中的δ为用户给定的一个任意小正实数.δ可满足是指在允许δ误差存在的情况下,相关函数可满足.基于此判定方法,相关研究团队开发了非线性约束集 SMT 求解器 dReal[53],并在此基础上开发了非线性混成自动机有界模型检验工具 dReach.

此外,针对特定类型的非线性混成系统,相关有界可达性验证研究也取得了一定的进展.由于一般非线性数值计算的复杂度过高,难以处理大规模系统,而若对问题进行一定限制,则当所有非线性函数均为凸函数时,相

关约束集的求解与判定可以采用类似线性规划的方法进行处理,因此可判定问题规模大幅度增加.针对此特性,南京大学团队提出了凸性混成自动机,并给出了其上的面向路径可达性判定方法及有界可达性判定方法,以对大规模凸性系统进行判定[54].类似地,Cimatti 等人提出了一种分段单调的非线性混成自动机,并基于 iSAT 实现了相关系统的有界可达性验证[55].

5 混成系统定理证明

与模型检验主要通过遍历状态空间来证明性质不同,定理证明主要解决如何利用逻辑和数学推理证明的手段验证软件的关键性质.经过多年的研究,定理证明已经取得了一系列进展,提出了一系列相关方法,比如顺序程序的公理化理论[56]、CSP[6]、CCS[57]等等.该方法需要有经验的用户提供大量的公理、前提条件及其他的系统相关信息,如不变式(invariant)等等,再使用逻辑推理的方法对其证明,如著名的 Hoare 三元组$\{P\}S\{Q\}$[56].这种三元组描述了一段代码的执行如何改变计算的状态,其中,S为执行的程序命令,而 P 与 Q 为系统状态断言,分别描述 S 执行前与执行后的前置与后置条件.整个三元组可以解读为,只要 P 在执行 S 前的状态下成立,则在执行之后 Q 也成立.以此三元组为中心,Hoare 为顺序程序的正确性构造了公理和推理规则[56],可以基于此公理系统对程序进行推理证明.在 Hoare 逻辑基础上,相关科研工作者为并发、指针等复杂程序上也建立了相关逻辑系统,并开发实现了一系列工具,如 Isabelle[58],PVS[59],HOL[60]等.目前,相关工具已经在大量领域得到了应用.

Hoare 逻辑同样被扩展应用到混成系统的定理证明当中.为了对混成系统进行推理证明,需要可以描述混成系统实时行为及相互间交互的混成系统建模语言,并且需要建立在其基础上的可对混成系统实施微分行为进行推理的逻辑系统.何积丰院士等人在 CSP 基础上提出了混成 CSP 语言[11,12]可描述组合混成系统.在此基础上,周巢尘院士等人对 Hoare 逻辑进行扩展,建立了能够验证混成系统的逻辑[61,62],并使用混成 CSP 对包括国产列控系统 CTCS-3 在内系统的大量关键场景进行描述,并成功地进行了定理证明.

如果系统中所有状态均满足性质ψ,则称ψ为此系统的不变式(invariant).不变式在定理证明中发挥了重要的作用,因此,如何生成不变式,特别是如何针对混成系统的特性生成系统行为中的连续不变式(continuous invariant),是相关方向的一个重要研究方向.Sankaranarayanan 等人在文献[63,64]提出了系列方法来生成混成系统的不变式.该方法通过预先给出带参多项式型模板(template)的方式,利用约束求解(constraint solving),基于相应模板生成多项式等式型的混成系统不变式.与此类似,Gulwani 等人采用 SMT 约束求解方式,基于多项式模板生成混成系统不变式[65].针对相关问题,国内研究者也进行了一定研究,如詹乃军老师等人在文献[61]中给出了一种通过李导数(Lie derivatives)求解混成系统半代数不变式(semi-algebraic invariants)的方法,并在后续工作中将其成功地应用到混成系统控制生成中[66].

Barrier Certificate 生成[67]是混成系统不变式生成方向上的另一个重要成果.文献[67]提出了一种通过利用平方和(sum of square)与半定规划(semidefinite programming)技术来生成多项式不等式形式的混成自动机不变式的方法,并称此类多项式不等式为 barrier certificate.因为这些多项式构成了系统状态集外的一层屏障(barrier),从而将系统状态集与不安全状态集相隔离.

上述方法主要通过给定不变式的多项式模板,再通过约束求解形式来计算不变式.Tiwari 等人在文献[68,69]中给出了一种面向线性系统的不变式生成方式,并进一步展示对于相关类型的线性系统都可以通过此方法计算最强多项式型不变式[69].然而,此方法的主要限制也在于它主要面向线性系统有效.

动态逻辑(DL)[70,71]是一种成功的通过推理证明方式验证有限状态离散系统的方法.它使用一种规约和验证语言来描述系统行为以及这些行为可以使系统到达的状态.动态逻辑提供了参数化的模态算子$\langle\alpha\rangle$和$[\alpha]$来表示可以被系统α达到的状态.这两个算子可以放在任意公式的前面,$[\alpha]\psi$表示系统α能够达到的所有状态都满足公式ψ,$\langle\alpha\rangle\psi$则表示至少有 1 个状态是系统α可达的,同时这个状态满足ψ.这些形式可以很自然地表示α的行为的必然性质与可能性质,并且可以以命题逻辑的方式组合起来.

通过对动态逻辑进行扩展,美国卡耐基梅隆大学的 Platzer 等人近年来基于定理证明的方法提出了一种可以验证含有复杂动态行为的混成系统的可靠符号验证算法[15,72,73],这被视为近期混成系统定理证明领域的重

大突破性进展.其主要思想是:不求解微分方程,而是寻找这些微分方程的微分不变式(differential invariant),从而避免了求解微分方程可能导致的不可控计算量.对动态逻辑进行扩展以后,Platzer 等人提出了一种混成系统的验证逻辑——微分动态逻辑 dL[14],并通过不动点来计算微分不变式[72].在微分动态逻辑中,整个系统可以分解为多个子系统,并计算出各个子系统的子不变式,然后将这些子不变式重新组合为可靠的全局不变式.同时,Platzer 引入了微分饱和过程.该过程可以通过引入辅助微分变量来不断精化混成系统的动态模型,直到安全性声明能够被证明是不断被精化的系统的不变式.Platzer 等人开发了面向混成系统验证的基于微分动态逻辑的定理证明工具 KeYmaera[73]来自动化这一过程.目前,相关工作已在飞机防撞系统、欧洲列控 ETCS 系统等重要案例上进行了成功的应用.

6 混成系统形式化验证当前的研究方向

在前面几节中,我们围绕着建模语言、模型检验、定理证明等方向对现有混成系统形式化建模与验证方向上的工作与进展进行了概要的回顾与总结.然而随着科技的快速发展,当前嵌入式系统呈现出智能化、交互化、开放化等一系列特征.在前期工作的基础上,如何进行扩展与进一步研究,以应对现阶段的复杂实时嵌入式系统?现阶段的主要研究问题与重点关注又集中在什么方向?我们将在本节进行相应的探讨与展望.

1) 大规模组合非线性混成系统验证

长期以来,组合验证一直是形式化验证的难点.多成员组合后,系统行为状态空间会急剧增长,从而引发著名的状态空间爆炸问题.尽管混成系统形式化验证已经取得了较大的进展,但是目前能够验证的系统维度仍然面临着较大的限制.而实际应用系统中,更是经常会牵涉到大量成员的协作.因此,如何扩展现有混成系统的验证方法对大规模组合系统进行验证,是现阶段亟需突破的问题之一.

另一方面,虽然在上文中提到,在非线性混成系统形式化验证领域已经取得了一系列进展,dReach,HySAT,SpaceEx,flow*等工具纷纷出现,但是必须要承认的是,现有方法或者工具在可处理系统的种类以及可处理问题的规模上仍然存在着较大的限制.如何提出新方法,或者基于新的数学技术开发出有效的大规模非线性系统的验证工具,仍然是相关研究领域重点关注的问题.

2) 开放动态系统行为预测与验证

与一般静态可预测的简单嵌入式系统相比,在 CPS、物理网等概念兴起之后,现在的嵌入式系统更加强调开放、协作,通过实时捕获、采集环境中或者其他协作成员的实时参数,从而进行自身策略的更加智能的调整.另一方面,在现代的复杂嵌入式系统中,参与协作的成员数目也存在动态变化的可能,随着时间的变化,老成员退出、新成员加入等情况会频繁出现.因此,相关系统模型中组合自动机数目可能不确定.此外,单个成员模型中可能存在大量自由参数,等待运行时填充.这就导致了模型状态空间无法在设计阶段进行预测、描述.而形式化验证,特别是模型检验的基本核心原理在于自动化遍历、求解系统的完整状态空间,当状态空间无法描述时,如何基于已有混成系统形式化验证技术成果,设计新型验证方法对不可控的系统行为进行可控的验证,是相关领域目前重点关注的问题.目前,在模型检验和定理证明领域已经开始有学者在相关方向上进行了研究,包括混成自动机在线建模与验证[74]、基于量化微分动态逻辑的动态结构定理证明[75]等等,并且已经成功地在包括列控系统、汽车网络等典型开放动态复杂嵌入式系统上进行了验证.

3) 概率、随机行为建模与验证

如上文所述,在现代复杂嵌入式网络行为中,不确定性行为越来越常见.如何在建模阶段对相关随机、不确定行为进行描述,是对复杂不确定嵌入式网络建模研究的一个重要方向.此外,前文中所介绍的工作所验证问题主要关注于系统运行中会否出现不安全状态、系统运行中变量取值能否稳定在一定范围之内等确定性问题,事实上,正如现阶段工业界所采用的质量保障准则一样,验证系统行为中出现不安全状态的概率有多大、相关概率数值是否可以容忍,对系统可信性理解与保障也具有重要意义.

基于马尔可夫链的概率行为建模与验证,是一种对概率系统进行描述与验证的重要技术.大量工具,如 PRISM 等,基于此思想开发并对实际系统进行了验证[76].此外,基于统计的模型检验技术[77]可以通过对系统随

机运行、仿真并观察、统计现有运行情况来对系统满足性质的概率给出其数值范围.在对大规模难以验证复杂系统上采用此方法,可以有效地规避系统复杂性,并对系统质量给出佐证.上述工作目前已经开始被拓展到混成系统领域.通过为离散跳转、连续行为等不同元素添加随机因素,已经提出了概率混成自动机(probabilistic hybrid automata)[78]、随机混成自动机(stochastic hybrid automata)[79]等一系列概率、随机混成模型.在这些模型基础上,相关研究人员开展了一系列以可达性验证为主的研究,如文献[80−83],开发了以随机时间自动机验证工具 UPPAAL[84]及基于 PHAVer 开发的概率混成自动机验证工具 ProHVer[78]等为代表的相关验证工具.同时,也提出了面向随机混成系统的随机 SMT 决策方法,为相关系统的有界模型检验的开展提供了可能[83].如何在以上研究的基础上进一步拓展,在大规模复杂不确定嵌入式系统上进行概率、随机混成自动机建模并进行相应概率性质验证,是一个非常值得关注的方向.

4) 混成系统控制生成

以上所有内容主要关注于混成系统的形式化验证,即,给定系统模型与规约,证明该模型满足相关规约.与此不同,控制生成(control synthesis)则关注于如何基于给定的系统规约来设计并生成符合相应规约的控制模型.在离散、连续行为交织的复杂混成系统领域,如何进行自动化控制生成显得更为重要.目前,相关研究者已经在此领域取得了一定的成果,主要工作集中在面向安全需求的控制生成及最优控制生成(optimal control synthesis)等方面.在面向安全需求的控制生成方面,文献[85]利用迭代式计算可达状态集不动点的方式来生成符合安全需求的控制器,文献[86,87]提出了另一种类似的基于零和博弈的可达集计算方法,以解决相关控制生成问题.此外,Tiwari 等人在文献[88,89]中采用基于不变式约束求解的方式来生成符合安全需求的控制模型.

在安全控制生成的基础上增加相应的开销方程(cost function),生成使得相应开销方程取值最小的控制器,是近年来混成系统控制生成的一个关注点.文献[90−93]分别采用了诸如博弈、机器学习、量词消除、高斯平滑等多种方法来进行混成系统最优控制生成的探索性研究,并取得了良好的效果.如何拓展相关研究,处理规模更大、行为更加复杂的系统,仍然有待进一步研究.另一方面,以上工作都是在形式化验证的角度上开展的最优控制生成相关工作,而在控制领域,目前已有大量的混成系统最优控制相关工作,如文献[94,95].如何将两个领域的相关技术相融合,也是一个值得探索的问题.

7 总 结

混成系统是实时嵌入式系统中的一种重要子类,在工控、国防等安全攸关系统广泛存在并发挥着至关重要的作用.因此,如何对其进行有效的质量保障,成为一个具有重要意义的问题.由于混成系统行为中离散逻辑跳转与实时连续行为交织混杂非常复杂,难以掌握与控制,测试与仿真等常规手段无法穷举系统的所有可能运行情况,不足以发现系统中的所有隐藏问题,因此,对混成系统进行形式化建模与验证非常重要.

混成系统的形式化验证是一项被广泛研究了近 20 年的重要问题,目前已经取得了大量重要成果,并且在很多实际系统中得到了应用.本文从混成系统形式化规约语言与混成系统形式化验证技术,特别是模型检验技术、定理证明技术等层面对相关研究进展以及在当代系统发展背景下的新研究热点与方向进行了总结与讨论,以期帮助读者了解现有研究工作的概要情况,并对今后的研究开展有所启发.

致谢 诚挚感谢审稿专家所给出的宝贵意见与建议.同时,我们也非常感谢在本文写作过程中,国防科学技术大学计算机学院并行与分布处理国防科技重点实验室助理研究员陈立前博士在混成系统定理证明方向的讨论与帮助,感谢南京大学计算机科学与技术系黄超博士生在混成系统控制生成方向的探讨.

References:

[1] Henzinger TA. The theory of hybrid automata. In: Proc. of the 11th Annual IEEE Symp. on Logic in Computer Science (LICS'96). New Brunswick: IEEE, 1996. 278−292. [doi: 10.1109/LICS.1996.561342]

[2] Myers GJ. Art of Software Testing. 2nd ed., New York: John Wiley & Sons, Inc., 1979.

[3] Bertolino A. Software testing research: Achievements, challenges, dreams. In: Briand LC, ed. Proc. of the 2007 Future of Software Engineering (FOSE 2007). Washington: IEEE Computer Society, 2007. 85−103. [doi: 10.1109/FOSE.2007.25]

[4] Peled DA. Software Reliability Methods. New York: Springer-Verlag, 2001.

[5] Jr. Clarke EM, Grumberg O, Peled DA. Model Checking. Cambridge: The MIT Press, 1999.

[6] Hoare CAR. Communicating Sequential Processes. Hertfordshire: Prentice-Hall Int'l, 1985.

[7] Harel D. Statecharts: A visual formalism for complex systems. Science of Computer Programming, 1987,8(3):231−274. [doi: 10.1016/0167-6423(87)90035-9]

[8] Rushby JM. Automated deduction and formal methods. In: Alur R, Henzinger TA, eds. Proc. of the 8th Int'l Conf. on Computer Aided Verification (CAV'96). London: Springer-Verlag, 1996. 169−183. [doi: 10.1007/3-540-61474-5_67]

[9] Hoare CAR, He J. Unifying Theories of Programming. Hertfordshire: Prentice-Hall Int'l, 1998.

[10] Alur R. Formal verification of hybrid systems. In: Chakraborty S, Jerraya A, Baruah SK, Fischmeister S, eds. Proc. of the 11th Int'l Conf. on Embedded Software (EMSOFT 2011). New York: ACM Press, 2011. 273−278. [doi: 10.1145/2038642.2038685]

[11] He J. From CSP to hybrid systems. In: Roscoe AW, ed. A Classical Mind: Essays in Honour of C.A.R. Hoare. Hertfordshire: Prentice-Hall Int'l, 1994. 171−189.

[12] Zhou C, Wang J, Ravn AP. A formal description of hybrid systems. In: Alur R, Henzinger TA, Sontag ED, eds. Proc. of the Hybrid Systems'95. LNCS 1066, Heidelberg: Springer-Verlag, 1995. 511−530. [doi: 10.1007/BFb0020972]

[13] David R. Modeling of hybrid systems using continuous and hybrid Petri nets. In: Proc. of the 6th Int'l Workshop on Petri Nets and Performance Models (PNPM'97). Washington: IEEE Computer Society, 1997. [doi: 10.1109/PNPM.1997.595536]

[14] Platzer A. Differential dynamic logic for hybrid systems. Journal of Automated Reasoning, 2008,41(2):143−189. [doi: 10.1007/s10817-008-9103-8]

[15] Henzinger TA, Ho PH, Wong-Toi H. Algorithmic analysis of nonlinear hybrid systems. IEEE Trans. on Automatic Control, 1998, 43(4):540−554. [doi: 10.1109/9.664156]

[16] Alur R, Courcoubetis C, Halbwachs N, Henzinger TA, Ho PH, Nicollin X, Olivero A, Sifakis J, Yovine S. The algorithmic analysis of hybrid systems. Theoretical Computer Science, 1995,138(1):3−34. [doi: 10.1016/0304-3975(94)00202-T]

[17] Alur R, Dill DL. A theory of timed automata. Theoretical Computer Science, 1994,126(2):183−235. [doi: 10. 1016/0304-3975(94)90010-8]

[18] Lamport L. Proving the correetness of multiprocess programs. IEEE Trans. on Software Engineering, 1977,SE-3(2):125−143. [doi: 10.1109/TSE.1977.229904]

[19] Manna Z, Pnueli A. The Temporal Logic of Reactive and Concurrent Systems. New York: Springer-Verlag, 1992.

[20] Asarin E, Dang T, Maler O. d/d*t*: A tool for reachability analysis of continuous and hybrid systems. In: Proc. of the Int'l Federation of Automatic Control (IFAC), Nonlinear Control Systems. Oxford: Pergamon, 2001. 20−31.

[21] Asarin E, Dang T, Maler O, Bournez O. Approximate reachability analysis of piecewise-linear dynamical systems. In: Lynch N, ed. Proc. of the 3rd Int'l Workshop on Hybrid Systems: Computation and Control (HSCC 2000). London: Springer-Verlag, 2000. 20−31. [doi: 10.1007/3-540-46430-1_6]

[22] Bemporad A, Morari M. Verification of hybrid systems via mathematical programming. In: Vaandrager FW, ed. Proc. of the 2nd Int'l Workshop on Hybrid Systems (HSCC'99). London: Springer-Verlag, 1999. 31−45. [doi: 10.1007/3-540-48983-5_7]

[23] Bemporad A, Torrisi FD, Morari M. Optimization-Based verification and stability characterization of piecewise affine and hybrid systems. In: Lynch N, ed. Proc. of the 3rd Int'l Workshop on Hybrid Systems: Computation and Control (HSCC 2000). London: Springer-Verlag, 2000. 45−58. [doi: 10.1007/3-540-46430-1_8]

[24] Botchkarev O, Tripakis S. Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations. In: Lynch N, ed. Proc. of the 3rd Int'l Workshop on Hybrid Systems: Computation and Control (HSCC 2000). London: Springer-Verlag, 2000. 73−88.

[25] Kurzhanski AB, Varaiya P. Ellipsoidal techniques for reachability analysis. In: Lynch N, ed. Proc. of the Hybrid Systems: Computation and Control (HSCC 2000). London: Springer-Verlag, 2000. 202−214. [doi: 10.1007/3-540-46430-1_19]

[26] Kesten Y, Pnueli A, Sifakis J, Yovine S. Integration graphs: A class of decidable hybrid systems. In: Grossman RL, ed. Proc. of the Hybrid Systems. LNCS 736, Berlin, Heidelberg: Springer-Verlag, 1993. 179−208. [doi: 10.1007/3-540-57318-6_29]

[27] Henzinger TA, Kopke PW, Puri A, Varaiya P. What's decidable about hybrid automata? In: Leighton F, Borodin A, eds. Proc. of the 27th Annual ACM Symp. on Theory of Computing. New York: ACM, 1995. 373−382. [doi: 10.1145/225058.225162]

[28] Silva BI, Stursberg O, Krogh BH, Engell S. An assessment of the current status of algorithmic approaches to the verification of hybrid systems. In: Proc. of the 40th Conf. on Decision and Control. Orlando: IEEE, 2001. 2867−2874. [doi: 10.1109/.2001.980711]

[29] Henzinger TA, Ho PH, Wong-Toi H. HYTECH: A model checker for hybrid systems. In: Grumberg O, ed. Proc. of the 9th Int'l Conf. on Computer Aided Verification (CAV'97). London: Springer-Verlag, 1997. 460−463. [doi: 10.1007/3-540-63166-6_48]

[30] Frehse G. PHAVer: Algorithmic verification of hybrid systems past HyTech. In: Morari M, ed. Proc. of the Int'l Conf. on Hybrid Systems: Computation and Control. Heidelberg: Springer-Verlag, 2005. 258−273. [doi: 10.1007/978-3-540-31954-2_17]

[31] Silva B, Richeson K, Krogh B, Chutinan A. Modeling and verifying hybrid dynamic systems using checkmate. In: Proc. of the 4th Int'l Conf. on Automataion of Mixed Processes. Shaker Publisher, 2000. 323−328.

[32] Chen X, Abraham E, Sankaranarayanan S. Flow*: An analyzer for non-linear hybrid systems. In: Sharygina N, Veith H, eds. Proc. of the Computer Aided Verification 2013. LNCS 8044, Beilin, Heidelberg: Springer-Verlag, 2013. 258−263. [doi: 10.1007/978-3-642-39799-8_18]

[33] Frehse G, Le Guernic C, Donzé A, Cotton S, Ray R, Lebeltel O, Ripado R, Girard A, Dang T, Maler O. SpaceEx: Scalable verification of hybrid systems. In: Gopalakrishnan G, Qadeer S, eds. Proc. of the Computer Aided Verification. LNCS 6806, Berlin, Heidelberg: Springer-Verlag, 2011. 379−395. [doi: 10.1007/978-3-642-22110-1_30]

[34] Clarke E, Grumberg O, Jha S, Lu Y, Veith H. Counterexample-Guided abstraction refinement. In: Emerson EA, Sistla AP, eds. Proc. of the Computer aided verification. LNCS 1855, Berlin, Heidelberg: Springer-Verlag, 2000. 154−169. [doi: 10.1007/10722167_15]

[35] Clarke E, Fehnker A, Han Z, Krogh B, Stursberg O, Theobald M. Verification of hybrid systems based on counterexample-guided abstraction refinement. In: Garavel H, Hatcliff J, eds. Proc. of the Tools and Algorithms for the Construction and Analysis of Systems. LNCS 2619, Berlin, Heidelberg: Springer-Verlag, 2003. 192−207. [doi: 10.1007/3-540-36577-X_14]

[36] Alur R, Dang T, Ivančić F. Counterexample-Guided predicate abstraction of hybrid systems. Theoretical Computer Science, 2006, 354(2):250−271. [doi:10.1016/j.tcs.2005.11.026]

[37] Jha S, Krogh BH, Weimer JE, Clarke EM. Reachability for linear hybrid automata using iterative relaxation abstraction. In: Bemporad A, Bicchi A, Buttazzo G, eds. Proc. of the Hybrid Systems: Computation and Control. Berlin, Heidelberg: Springer-Verlag, 2007. 287−300. [doi: 10.1007/978-3-540-71493-4_24]

[38] Li XD, Jha SK, Bu L. Towards an efficient path-oriented tool for bounded reachability analysis of linear hybrid systems using linear programming. Electronic Notes on Theoretical Computer Science, 2007,174(3):57−70. [doi: 10.1016/j.entcs.2006.12.023]

[39] Bryant RE. Graph-Based algorithms for boolean function manipulation. IEEE Trans. on Computers, 1986,C-35(8):677−691. [doi: 10.1109/TC.1986.1676819]

[40] Biere A, Cimatti A, Clarke EM, Strichman O, Zhu Y. Bounded model checking. Advance in Computers, 2008,58:117−148.

[41] Zhang L, Malik S. The quest for efficient boolean satifiability solvers. In: Brinksma E, Larsen K, eds. Proc. of the Computer Aided Verification. Heidelberg: Springer-Verlag, 2002. 17−36.

[42] Audemard G, Bozzano M, Cimatti A, Sebastiani R. Verifying industrial hybrid systems with MathSAT. Electronic Notes on Theoretical Computer Science, 2005,119(2):17−32. [doi: 10.1016/j.entcs.2004.12.022]

[43] Audemard G, Cimatti A, Kornilowicz A, Sebastiani R. Bounded model checking for timed systems. In: Peled DA, Vardi M, eds. Proc. of the FORTE 2002. LNCS 2529, Berlin, Heidelberg: Springer-Verlag, 2002. 243−259. [doi: 10.1007/3-540-36135-9_16]

[44] Franzle M, Herde C. Efficient proof engines for bounded model checking of hybrid systems. Electronic Notes on Theoretical Computer Science, 2005,133:119−137. [doi: 10.1016/j.entcs.2004.08.061]

[45] Bu L, Li Y, Wang LZ, Li XD. BACH: Bounded reachability checker for linear hybrid automata. In: Cimatti A, Jones RB, eds. Proc. of the 8th Int'l Conf. on Formal Methods in Computer Aided Design (FMCAD 2008). Portland: IEEE Computer Society Press, 2008. 1−4. [doi: 10.1109/FMCAD.2008.ECP.13]

[46] Yang Y, Bu L, Li XD. Forward and backward: Bounded model checking of linear hybrid automata from two directions. In: Cabodi G, Singh S, eds. Proc. of the Formal Methods in Computer-Aided Design. Cambridge: IEEE Computer Society Press, 2012. 204−208.

[47] Bu L, Yang Y, Li XD. IIS-Guided DFS for efficient bounded reachability analysis of linear hybrid automata. In: Eder K, Lourenço J, Shehory O, eds. Proc. of the Hardware and Software: Verification and Testing 2011. LNCS 7261, Berlin, Heidelberg: Springer-Verlag, 2012. 35−49. [doi: 10.1007/978-3-642-34188-5_7]

[48] Bu L, Li XD. Path-Oriented bounded reachability analysis of composed linear hybrid systems. Int'l Journal on Software Tools for Technology Transfer, 2011,13(4):307−317. [doi: 10.1007/s10009-010-0163-9]

[49] Bu L, Cimatti A, Li XD, Mover S, Tonetta S. Model checking of hybrid systems using shallow synchronization. In: Hatcliff J, Zucca E, eds. Proc. of the Formal Techniques for Distributed Systems. LNCS 6117, Berlin, Heidelberg: Springer-Verlag, 2010. 155−169. [doi: 10.1007/978-3-642-13464-7_13]

[50] Moore RE, Bierbaum F. Methods and Applications of Interval Analysis. Philadelphia: Society of Industrial and Applied Mathematics, 1979.

[51] Franzle M, Herde C, Teige T. Efficient solving of large non-linear arithmetic constraint systems with complex boolean structure. Journal on Satisfiability, Boolean Modeling and Computation, 2007,1:209−236.

[52] Gao SC, Avigad J, Clarke EM. Delta-Complete decision procedures for satisfiability over the reals. In: Gramlich B, Miller D, Sattler U, eds. Proc. of the Int'l Joint Conf. on Automated Reasoning (IJCAR). Heidelberg: Springer-Verlag, 2012. 286−300.

[53] Gao SC, Kong S, Clarke EM. dReal: An SMT solver for nonlinear theories of the reals. In: Bonacina MP, ed. Proc. of the Conf. on Automated Deduction 2013. LNAI 7898, Berlin, Heidelberg: Springer-Verlag, 2013. 208−214. [doi: 10.1007/978-3-642-38574-2_14]

[54] Bu L, Zhao JH, Li XD. Path-Oriented reachability verification of a class of nonlinear hybrid automata using convex programming. In: Barthe G, Hermenegildo M, eds. Proc. of the Verification, Model Checking, and Abstract Interpretation 2010. LNCS 5944, Berlin, Heidelberg: Springer-Verlag, 2010. 79−94. [doi: 10.1007/978-3-642-11319-2_9]

[55] Cimatti A, Mover S, Tonetta S. A quantifier-free SMT encoding of non-linear hybrid automata. In: Cabodi G, Singh S, eds. Proc. of the Formal Methods in Computer-Aided Design. Cambridge: IEEE, 2012. 187−195.

[56] Hoare CAR. An axiomatic basis for computer programming. Communications of the ACM, 1969,12(10):576−585. [doi: 10.1145/363235.363259]

[57] Milner R. A Calculus of Communicating Systems. Heidelberg: Springer-Verlag, 1980.

[58] Paulson LC. Isabelle: A generic theorem prover. LNCS 828, Heidelberg: Springer-Verlag, 1994.

[59] Owre S, Rushby M, Shankar N. PVS: A prototype verification system. In: Kapur D, ed. Proc. of the 11th CADE. LNCS 607, Heidelberg: Springer-Verlag, 1992. 748−752. [doi: 10.1007/3- 540-55602-8_217]

[60] Gordon M. Introduction to the HOL system. In: Archer M, Joyce JJ, Levitt KN, Windley PJ, eds. Proc. of the Int'l Workshop on the HOL Theorem Proving System and Its Applications. Davis: IEEE Computer Society, 1991. 2−3.

[61] Liu J, Zhan N, Zhao H. Computing semi-algebraic invariants for polynomial dynamical systems. In: Chakraborty S, Jerraya A, Baruah SK, Fischmeister S, eds. Proc. of the Embedded Software (EMSOFT 2011). New York: ACM Press, 2011. 97−106. [doi: 10.1145/2038642.2038659]

[62] Liu J, Lv J, Quan Z, Zhan N, Zhao H, Zhou C, Zou L. A calculus for hybrid CSP. In: Ueda K, ed. Proc. of the APLAS 2010. LNCS 6461, Heidelberg: Springer-Verlag, 2010. 1−15. [doi: 10.1007/978-3-642-17164-2_1]

[63] Sankaranarayanan S, Sipma H, Manna Z. Constructing invariants for hybrid systems. In: Alur R, Pappas GJ, eds. Proc. of the Hybrid Systems: Computation and Control. Heidelberg: Springer-Verlag, 2004. 539−554. [doi: 10.1007/978-3-540-24743-2_36]

[64] Sankaranarayanan S. Automatic invariant generation for hybrid systems using ideal fixed points. In: Johansson KH, Wang Y, eds. Proc. of the Hybrid Systems: Computation and Control. New York, ACM, 2010. 221−230. [doi: 10.1145/1755952.1755984]

[65] Gulwani S, Tiwari A. Constraint-Based approach for analysis of hybrid systems. In: Proc. of the Computer Aided Verification. Heidelberg: Springer-Verlag, 2008. 190−203. [doi: 10.1007/978-3-540-70545-1_18]

[66] Zhao H, Zhan N, Kapur D. Synthesizing switching controllers for hybrid systems by generating invariants. In: Liu Z, Woodcock J, Zhu H, eds. Proc. of the Theories of Programming and Formal Methods. Berlin, Heidelberg: Springer-Verlag, 2013. 354−373. [doi: 10.1007/978-3-642-39698-4_22]

[67] Prajna S, Jadbabaie A. Safety verification of hybrid systems using barrier certificates. In: Alur R, Pappas GJ, eds. Proc. of the Hybrid Systems: Computation and Control. Heidelberg: Springer-Verlag, 2004. 477−492. [doi: 10.1007/978-3-540-24743-2_32]

[68] Tiwari A. Approximate reachability for linear systems. In: Maler O, Pnueli A, eds. Proc. of the Hybrid Systems: Computation and Control 2003. LNCS 2623, Heidelberg: Springer-Verlag, 2003. 514−525. [doi: 10.1007/3-540-36580-X_37]

[69] Carbonell E, Tiwari A. Generating polynomial invariants for hybrid systems. In: Morari M, Thiele L, eds. Proc. of the Hybrid Systems: Computation and Control 2005. LNCS 3414, Heidelberg: Springer-Verlag, 2005. 590−605. [doi: 10.1007/978-3-540-31954-2_38]

[70] Harel D, Kozen D, Tiuryn J. Dynamic Logic. Cambridge: The MIT Press, 2000.

[71] Harel D. First-Order Dynamic Logic. In: Hansen PB, Gries D, Moler C, eds. Lecture Notes in Computer Science, Vol.68. Berlin, Heidelberg, New York: Springer-Verlag, 1979.

[72] Platzer A, Clarke E. Computing differential invariants of hybrid systems as fixedpoints. Formal Methods in Systems Design, 2009, 35(1):98−120. [doi: 10.1007/s10703-009-0079-8]

[73] Platzer A, Quesel JD. KeYmaera: A hybrid theorem prover for hybrid systems. In: Armando A, Baumgartner P, Dowek G, eds. Proc. of the Int'l Joint Conf. on Automated Reasoning. LNCS 5195, Berlin, Heidelberg: Springer-Verlag, 2008. 171−178. [doi: 10.1007/978-3-540-71070-7_15]

[74] Bu L, Wang QX, Chen X, Wang LZ, Zhang T, Zhao JH, Li XD. Toward online hybrid systems model checking of cyber-physical systems time-bounded short-run behavior. ACM SIGBED Review, 2011,8(2):7−10. [doi: 10.1145/2000367.2000368]

[75] Platzer A. Quantified differential invariants. In: Caccamo M, Frazzoli E, Grosu R, eds. Proc. of the 14th ACM Int'l Conf. on Hybrid Systems: Computation and Control 2011. New York: ACM Press, 2011. 63−72. [doi: 10.1145/1967701.1967713]

[76] Kwiatkowska M, Norman G, Parker D. Advances and challenges of probabilistic model checking. In: Proc. of the 48th Annual Allerton Conf. on Communication, Control and Computing. Allerton: IEEE Press, 2010. 1691−1698. [doi: 10.1109/ALLERTON.2010.5707120]

[77] Younes HLS, Simmons RG. Probabilistic verification of discrete event systems using acceptance sampling. In: Brinksma E, Larsen KG, eds. Proc. of the Computer Aided Verification. LNCS 2404, Berlin, Heidelberg: Springer-Verlag, 2002. 223−235. [doi: 10.1007/3-540-45657-0_ 17]

[78] Zhang L, She Z, Ratschan S, Hermanns H, Hahn EM. Safety verification for probabilistic hybrid systems. In: Touili T, Cook B, Jackson P, eds. Proc. of the Computer Aided Verification. Berlin, Heidelberg: Springer-Verlag, 2010. 196−211. [doi: 10.1007/978-3-642-14295-6_21]

[79] Hu J, Lygeros J, Sastry S. Towards a theory of stochastic hybrid systems. In: Lynch N, Krogh B, eds. Proc. of the Hybrid Systems: Computation and Control. LNCS 1790, Berlin, Heidelberg: Springer-Verlag, 2000. 160−173. [doi: 10.1007/3-540-46430-1_16]

[80] Fränzle M, Hahn E, Hermanns H, Wolovick N, Zhang L. Measurability and safety verification for stochastic hybrid systems. In: Caccamo M, Frazzoli E, Grosu R, eds. Proc. of the 14th Int'l Conf. on Hybrid Systems: Computation and Control. Chicago: ACM Press, 2011. 43−52. [doi: 10.1145/1967701.1967710]

[81] Abate A, Katoen J, Lygeros J, Prandini M. Approximate model checking of stochastic hybrid systems. European Journal of Control, 2010,16(6):624−641. [doi: 10.3166/ejc.16.624-641]

[82] Abate A, Prandini M, Lygeros J, Sastry S. Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. Automatica, 2008,44(11):2724−2734. [doi: 10.1016/j.automatica.2008.03.027]

[83] Fränzle M, Hermanns H, Teige T. Stochastic satisfiability modulo theory: A novel technique for the analysis of probabilistic hybrid systems. In: Egerstedt M, Mishra B, eds. Proc. of the Hybrid Systems: Computation and Control. LNCS 4981, Berlin, Heidelberg: Springer-Verlag, 2008. 172−186. [doi: 10.1007/978-3-540-78929-1_13]

[84] David A, Larsen KG, Legay A, Mikucionis M, Wang Z. Time for statistical model checking of real-time systems. In: Gopalakrishnan G, Qadeer S, eds. Proc. of the Computer Aided Verification. LNCS 6806, Berlin, Heidelberg: Springer-Verlag, 2011. 349−355. [doi: 10.1007/978-3-642-22110-1_27]

[85] Asarin E, Bournez O, Dang T, Maler O, Pnueli A. Effective synthesis of switching controllers for linear systems. Proc. of the IEEE, 2000,88(7):1011−25. [doi: 10.1109/5.871306]

[86] Lygeros J, Tomlin C, Sastry S. Controllers for reachability specifications for hybrid systems. Automatica, 1999,35(3):349−370. [doi: 10.1016/S0005-1098(98)00193-9]

[87] Tomlin C, Lygeros L, Sastry S. A game-theoretic approach to controller design for hybrid systems. Proc. of the IEEE, 2000,88(7): 949−970. [doi: 10.1109/5.871303]

[88] Taly A, Gulwani S, Tiwari A. Synthesizing switching logic using constraint solving. In: Proc. of the Verification, Model Checking, and Abstract Interpretation. LNCS 5403, Berlin, Heidelberg: Springer-Verlag, 2009. 305−319. [doi: 10.1007/978-3-540-93900-9_25]

[89] Taly A, Tiwari A. Switching logic synthesis for reachability. In: Carloni LP, Tripakis S, eds. Proc. of the 10th ACM Int'l Conf. on Embedded Software. New York: ACM Press, 2010. 19−28. [doi: 10.1145/1879021.1879025]

[90] Cassez F, Jessen JJ, Larsen KG, Raskin JF, Reynier PA. Automatic synthesis of robust and optimal controllers—An industrial case study. In: Majumdar R, Tabuada P, eds. Proc. of the Hybrid Systems: Computation and Control. LNCS 5469, Berlin, Heidelberg: Springer-Verlag, 2009. 90−104. [doi: 10.1007/978-3-642-00602-9_7]

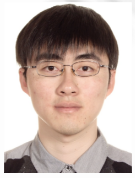
[91] Jha S, Seshia S, Tiwari A. Synthesis of optimal switching logic for hybrid systems. In: Chakraborty S, Jerraya A, Baruah SK, Fischmeister S, eds. Proc. of the Embedded Software (EMSOFT 2011). New York: ACM Press: IEEE, 2011. 107−116. [doi: 10.1145/2038642.2038 660]

[92] Zhao H, Zhan N, Kapur D, Larsen K. A "Hybrid" approach for synthesizing optimal controllers of hybrid systems: A case study of the oil pump industrial example. In: Giannakopoulou D, M´ery D. eds. Proc. of the Formal Methods (FM 2012). LNCS 7436, Berlin, Heidelberg: Springer-Verlag, 2012. 471−485. [doi: 10.1007/978-3-642-32759-9_38]

[93] Chaudhuri S, Solar-Lezama A. Smooth interpretation. ACM Sigplan Notices, 2010,45(6):279−291. [doi: 10.1145/1809028.1806629]

[94] Hedlund S, Rantzer A. Optimal control of hybrid systems. In: Proc. of the 38th IEEE Conf. on Decision and Control, Vol.4. Phoenix: IEEE, 1999. 3972−3977.

[95] Xu J, Chen Q. Optimal control of switched hybrid systems. In: Proc. of the 2011 8th Asian Control Conf. (ASCC). Kaohsiung: IEEE, 2011. 1216−1220.



卜磊(1983－),男,江苏东台人,博士,讲师,CCF 会员,主要研究领域为形式化方法,模型检验,实时混成系统,信息物理融合系统.
E-mail: bulei@nju.edu.cn

解定宝(1988－),男,博士生,主要研究领域为混成系统形式化验证.
E-mail: xdb@seg.nju.edu.cn