空间生命支持系统中VCCR子系统的安全性验证

李 倩 郁文生
(华东师范大学上海高可信计算重点实验室 上海 200062)

摘 要 基于动态微分逻辑的混成系统形式化验证理论,分析空间生命支持系统的一个子系统VCCR(Variable Configuration Carbon Dioxide Removal)的安全性。将VCCR系统基于混成程序建模,并给定需验证的安全性性质,使用KeYmaera混成系统形式化验证工具进行验证,证明了空间生命支持系统中VCCR子系统的安全性。

关键词 混成系统,生命支持系统,VCCR系统,形式化验证,KeYmaera工具
中图法分类号 TP302.2 **文献标识码** A

Safety Verification for VCCR Subsystem of Space Life Support System

LI Qian YU Wen-sheng
(Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai 200062, China)

Abstract This paper applied the differential-dynamic logic theory to analyze and verify the safety properties of a hybrid system named Variable Configuration Carbon Dioxide Removal (VCCR), which is a subsystem of space life support system. Based on the hybrid program, a model of the hybrid system VCCR with its safety properties was built. Using the hybrid system verification tool KeYmaera, it was formally proved that the system satisfies the safety properties under all possible scenarios.

Keywords Hybrid system, Life support system, VCCR system, Formal verification, KeYmaera tool

1 引言

自1957年苏联成功发射了世界上第一颗人造地球卫星以来,人们对宇宙航天的探索从未停止过。各国的科学工作者都致力于设计各种航天器和宇宙空间站,使航天员能够在外太空进行科学探索。在这些设计工作中首先要考虑的是保证航天员的生命健康和安全。VCCR(Variable Configuration Carbon Dioxide Removal)系统[6,10,14]就是保证宇航员在船员舱内能够获得安全气体环境(合理的氧气和二氧化碳比例)的一种生命支持系统的子系统,它根据船员舱当前各种气体的密度含量,通过吸附(Absorption)和解吸附(Desorption)二氧化碳的过程,控制船员舱内氧气和二氧化碳的含量。

混成系统[1,5]是含离散事件决策的一类动态系统,其演化过程由离散事件跳变和连续时间动态过程共同完成。温度控制系统是一个典型的混成系统实例,该系统由加热器和温度调节装置组成,系统中的变量就是室内温度和加热器的运行模式(开或关)。为了控制温度调节系统,需要统筹考虑整个系统中离散事件决策变量(加热器的运行模式)和连续变量(室内温度)之间的相互作用。现实环境中的大多数动态系统都具有混成的特性,本文要研究的VCCR系统即可通过混成系统理论很好地描述。本文使用混成系统领域新近提出的微分动态逻辑[3,4]形式化验证的理论和方法来验证VCCR系统的安全性。

本文第2节介绍混成系统的基本概念和混成系统形式化验证方法;第3节详细介绍VCCR系统;第4节将VCCR系统分为4个模式进行混成系统建模;第5节使用一种混成系统的形式化验证工具KeYmaera[7,8]对模型进行验证;最后对全文进行总结。

2 混成系统及其安全性验证

2.1 混成系统

混成系统是由离散事件跳变和连续时间动态变化组成的复杂系统。一个混成系统可以由以下的六元组来定义:$H=(\chi,L,X_0,I,F,T)$[1,3,5],其中:

$\chi\in\mathbf{R}^n$表示连续状态空间。

L表示一个有限的位置集。整个系统的状态空间表示为$X=L\times\chi$,系统中具体的一个状态可以表示为$(l,x)\in L\times\chi$。

$X_0\subseteq X$表示初始集。

$I:L\to 2^{\chi}$表示混成系统中的不变式。每一个位置l都对应一个集合$I(l)\subseteq\chi$,该集合包含在位置l所有可能的连续状态。

$F:X\to 2^{\mathbf{R}^n}$表示向量场的集合。F给每一个$(l,x)\in X$赋

到稿日期:2013-07-09 返修日期:2013-10-21 本文受国家自然科学基金(61370176,61070048),国家自然科学基金委员会创新研究群体科学基金(61021004),国家"863"计划(2011AA010101),国家"973"计划(2011CB302802),上海市重点学科建设项目(B412),上海市教育委员会科研创新项目(11ZZ37)资助。

李 倩(1989—),女,硕士生,主要研究领域为形式化建模,E-mail:liqian19891011@hotmail.com;郁文生(1967—),男,博士,教授,博士生导师,主要研究领域为物理信息融合系统、控制理论与控制工程。

值 $F(l,x)\in\mathbb{R}^n$ 时要满足微分包含约束 $\dot{x}\in F(l(t),x(t))$。

$T\in X\times X$ 表示在两个位置的离散跳变关系。也即一个离散跳变 $((l,x),(l',x'))\in T$ 表示系统可以从状态 (l,x) 跳变到状态 (l',x')。

嵌入式系统和基于软件的控制系统是典型的混成系统的例子，它们既有基于逻辑和事件驱动的离散状态变迁也有连续时间动态变化过程。将嵌入式系统和基于软件的控制系统作为混成系统研究的例子很多，比如自动公路系统[13]，也有很多使用微分动态逻辑对混成系统安全性验证的例子，如列车控制系统[2]、飞机飞行控制系统[15]、医疗系统[18]等。对于空间生命支持系统的子系统 VCCR，我们也将使用微分动态逻辑对混成系统建模，验证其安全性以保证宇航员的生命健康。

2.2 混成系统的形式化验证

混成系统在日常生活以及工业界有非常重要的应用，因此验证混成系统的设计方案是否安全可靠十分重要。混成系统的安全性可以理解为：对于一个混成系统，给定初始状态集、安全区域和非安全区域，从初始状态集开始的任意状态，经过符合系统定义的离散跳变和连续动态变化之后，不能抵达非安全区域，即状态轨线一直停留在安全区域内[1]。对于离散系统和连续时间动态系统的安全性验证，已有很多方法。对于离散系统来说，主要有两类方法：模型检测[11]和演绎式推理[12,19]。模型检测适用于有限状态空间的系统，对于有连续状态空间的混成系统而言，模型检测不能够覆盖全部的状态空间。演绎式推理通过推理规则集合验证系统的性质，但是这个过程中往往需要人工干预。对于连续系统，从控制理论[2]的角度看，有判断连续系统的稳定性和鲁棒性的方法。对于混成系统来说，结合和扩展上面离散系统和连续系统的研究方法，已提出了很多有成效的混成系统验证方式，文献[17]利用仿真通过模拟系统的运动轨迹，观察这些轨迹是否始终在安全区域内；文献[16]通过平方和理论构造障碍函数来判定系统是否会越过障碍函数进入不安全区域；文献[3,4]提出基于微分动态逻辑的形式化验证方法等等。

本文使用 A. Platzer 提出的微分动态逻辑的形式化验证方法[3,4]来验证混成系统的安全性。在微分动态逻辑中，我们使用混成程序 HP(Hybrid Program)来表示一个混成系统。混成程序 HP 的常用语法见表 1。

表 1 混成程序的常见语法

语法	语义
$x_1:=\theta_1,\cdots,x_n:=\theta_n$	离散的跳变集。同时将值 $\theta_1,\cdots,\theta_n$ 分别赋值给 $x_1,\cdots,x_n$
$x_1'=\theta_1,\cdots,x_n'=\theta_n\&\chi$	连续的演化过程。由项 $\theta_1,\cdots,\theta_n$ 组成的关于 $x_1,\cdots,x_n$ 的微分方程，并且带有一阶逻辑限制 χ
$?\chi$	状态检测。检测一阶逻辑范式 χ 是否为真
$\alpha;\beta$	两个混成程序 α,β 顺序发生
$\alpha\cup\beta$	两个混成程序 α,β 不确定性选择
α^*	重复执行混成程序 α n 次。n 为任意自然数

微分动态逻辑的描述可分为 3 部分：(1)一阶逻辑运算的逻辑连接，如 $\varphi\wedge\phi,\varphi\vee\phi,\varphi\rightarrow\phi,\neg\varphi$；(2)量词，比如 $\exists x\varphi$，$\forall x\varphi$；(3)模态操作符，比如$[\alpha]\varphi$表示在混成程序 α 的全部过程中满足条件 φ，$\langle\alpha\rangle\varphi$ 表示在混成程序 α 的至少一个过程中满足条件 φ。动态微分逻辑的证明策略是利用一系列推理规则把证明目标转换成实数运算域内的重言式[3,4]（具体的证明策略详见文献[3,4]）。

微分动态逻辑的验证过程与一般的形式化方法[9]类似：使用混成程序将混成系统及其安全性质表达出来，再根据微分动态逻辑的理论，实现自动化的推理证明。

下面的程序是一个使用微分动态逻辑建立混成系统模型的简单例子。混成程序如下：

$((a:=b)\cup(?v<20;a:=A));z'=v;v'=a$

该混成程序表示火车动态运动的系统。其中 a 表示加速度，z 表示位移。公式的前半部分$((a:=b)\cup(?v<20;a:=A))$表示火车既可以做减速运动($a:=b$)又可以做加速运动 $a:=A$，用符号$\cup$来连接表示不确定选择。但是如果是加速运动 $a:=A$，需要通过一个检测条件$?v<20$，如果条件满足则执行顺序操作符“;”的下一步。公式的后半部分 $z'=v;v'=a$ 则表示火车运动系统的物理过程。

3 空间生命支持系统

3.1 生命支持系统

生命支持系统[6,10,14]是一种用于支持在密闭空间中维持生命所必需环境的系统，常用于长时间的空间探测任务。生命控制系统的一个主要的部分是空气再生系统(Air Revitalization System，ARS)。ARS 通过提供氧气、清除二氧化碳、中间气体的处理和存储等功能为船舱提供充足的新鲜空气。本文主要研究 ARS 中与二氧化碳控制相关的子系统 VCCR 系统的安全性。

3.2 VCCR 系统

VCCR 系统[6,10,14]是空间生命支持系统的一个子系统，主要功能是从船员舱通过吸附(adsorption)过程吸收 CO_2，然后解吸附(desorption)CO_2 并且把解吸附之后的 CO_2 存储用于 O_2 的生成。整个 VCCR 由 3 个部分构成，分别是一个船员舱(Crew Cabin)，两个功能舱(Bed1，Bed2)和一个气体缓存区(Buffer)。两个功能舱的主要作用是通过如下 3 个阶段保证船员舱内的氧气和二氧化碳的含量维持在正常水平。

吸附阶段(adsorption phase)。吸附舱吸附船员舱的 CO_2，同时将含 CO_2 量很低的空气输入船员舱。整个吸附过程的原理是利用吸附物将气液混合物中的某种物质吸收。

解吸附阶段(desorption phase)。将吸附物中被吸附的物质还原(在我们的例子中是 CO_2)并且将其释放到气体缓存区当中。

空气补给阶段(air-save phase)。解吸附舱将除去 CO_2 的气体返回释放到船员舱当中。在这个过程中我们假设已经被吸附的 CO_2 被固化，也即不会再以气体的形式泄漏到船员舱当中。

由于吸附舱有可以吸收固体 CO_2 的上限值，因此该系统设计为当一个功能舱处于吸附阶段时，另一个功能舱处于解吸附和空气补给阶段。当在吸附阶段的功能舱达到它的吸收上限值时，两个功能舱阶段进行互换，继续工作(详见图 1)。我们假设吸附和解吸附的速率是常量，两个功能舱的吸附和解吸附速率相同。

我们要考虑的安全问题是根据以上的设计，在两个功能舱都满足吸收上限值的限制下，整个系统是否能一直满足船员舱内的氧气和二氧化碳的含量保持在合理安全的水平。Glavaski 等人从控制理论的角度，使用了基于平方和最优化构造障碍函数的方法[14]对 VCCR 系统的安全性进行了验证，

但是这种方法存在半定规划的局限性，并且最终得到的是一个包含十多个变量、几十项的多项式的障碍函数。本文将使用新近提出的微分动态逻辑对 VCCR 系统的安全性进行建模和验证，最终得到的结果与文献[14]中的结果一致。

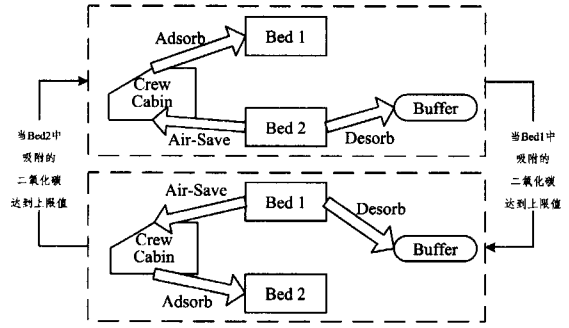

图 1　VCCR 工作模式示意图

4　VCCR 混成系统模型

在建立模型之前，首先要明确所要验证的系统的安全性质：在整个生命支持系统的运行过程当中，O_2 和 CO_2 的含量都要维持在一个合理的范围内。我们可以用如下的形式来表示，其中 *safe* 表示我们要验证的安全性质，*co2lower* 和 *co2upper* 分别表示 CO_2 的合理范围的上限和下限值，*o2lower* 和 *o2upper* 分别表示 O_2 的合理范围的上限和下限值。*LifeSupport* 表示生命支持系统，该系统有一个初始状态，然后有 n 个模式互相转换，每个模式 i 都有一组微分方程(*mode i diff enq*)和一组约束(*mode i constraints*)。

safe→[LifeSupport]safe

safe≡[co2lower≤co2≤co2upper&o2upper≤o2≤o2lower]

LifeSupport≡(Initial;

((mode 1 diff enq & mode 1 constraints)∪

(mode 2 diff enq & mode 2 constraints)∪

…

(mode n diff enq & mode nconstraints)))*

4.1　简单的 VCCR 模型

在形式化建模验证的过程中，一般先建立一个比较简单的模型来研究系统最基本的性质，然后再逐步对初步模型进行精化，使之更加完整、更能准确地反映实际系统。在本文的建模过程中，我们首先只验证 CO_2 含量的安全性质(下文中称这个模型为 VCCR 简单模型)。对于 VCCR 简单模型，要考虑的有如下 3 个问题：

1)模型共有几种模式

根据 Bed1 以及 Bed2 所处的功能阶段将整个 VCCR 系统分为以下 4 种模式(详见图 2)：

Mode1：Bed1 吸附阶段，Bed2 空气补给阶段；

Mode2：Bed1 吸附阶段，Bed2 解吸附阶段；

Mode3：Bed2 吸附阶段，Bed1 空气补给阶段；

Mode4：Bed2 吸附阶段，Bed1 解吸附阶段；

2)每个模式的微分方程和约束如何描述。

在给出具体的微分方程和约束之前，先给出模型中使用的变量的符号、单位以及含义。这里的变量定义声明和微分方程参考了文献[14]的部分相关内容。

V_C：船员舱的体积(m^3)；

V_1，V_2：Bed1 以及 Bed2 的体积(m^3)；

ρ_C，ρ_1，ρ_2：分别表示船员舱、Bed1 以及 Bed2 内的 CO_2 密度(g/m^3)；

m_{CO_2}：船员舱内补充 CO_2(亦即从 Buffer 流入船员舱)的速率(g/hr)；

Q_1，Q_2：CO_2 在 Bed1 以及 Bed2 中固体质量(g)；

Q_{max}：Bed1 以及 Bed2 内容纳固态 CO_2 的最大值(g)；

r_{CO_2}：船员舱内 CO_2 产生的速率(g/hr)；

r_{ads}，r_{des}：分别表示功能舱的吸附速率(g/hr)和解吸附速率(g/hr)；

r_1：在吸附阶段中从船员舱到功能舱的容积流率(m^3/hr)；

r_2：在解吸附和空气补给阶段从船员舱到功能舱的容积流率(m^3/hr)。

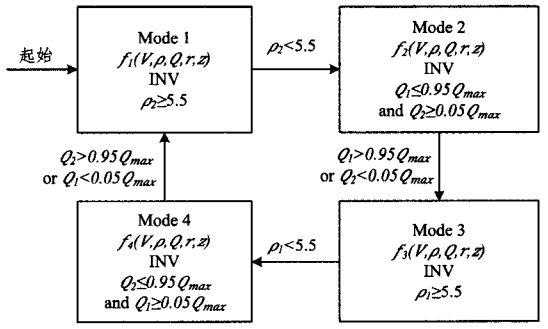

图 2　VCCR 工作模式转换图

根据上述变量，现在给出具体的微分方程和约束。Mode1 和 Mode3 以及 Mode2 和 Mode4 是类似的，不同的只是 Bed1 和 Bed2 的角色互换。所以下面只给出 Mode1 和 Mode2 的方程和约束，Mode3 和 Mode4 的可以类似给出。

Mode1：Bed1 吸附阶段，Bed2 空气补给阶段

在这个模式下，船员舱内产生 CO_2，Bed1 从船员舱吸收 CO_2，Bed2 以及 Buffer 会向船员舱内进行空气补给。以下是微分方程：

$$V_C \cdot \rho_C' = r_1 \cdot \rho_1 - r_1 \cdot \rho_C + r_2 \cdot \rho_2 + m_{CO_2} + r_{CO_2}$$

$$V_1 \cdot \rho_1' = r_1 \cdot \rho_C - r_1 \cdot \rho_1 - r_{ads}$$

$$V_2 \cdot \rho_2' = -r_2 \cdot \rho_2$$

$$Q_1' = r_{ads}$$

$$Q_2' = 0$$

因为 Bed2 在此过程中进行空气补给，而只有当 Bed2 中的 CO_2 大于一定值(5.5g/m^3)时空气补给才能正常进行，所以整个过程中要满足约束 $\rho_2 \geq 5.5$。

Mode2：Bed1 吸附阶段，Bed2 解吸附阶段

在这个模式下，Bed1 的行为和 Mode1 中相同，Bed2 进行解吸附过程，因此不会有气体从 Bed2 到船员舱当中。以下是微分方程：

$$V_C \cdot \rho_C' = r_1 \cdot \rho_1 - r_1 \cdot \rho_C + m_{CO_2} + r_{CO_2}$$

$$V_1 \cdot \rho_1' = r_1 \cdot \rho_C - r_1 \cdot \rho_1 - r_{ads}$$

$$V_2 \cdot \rho_2' = -r_2 \cdot \rho_2 - r_{ads}$$

$$Q_1' = r_{ads}$$

$$Q_2' = -r_{des}$$

过程中要保证 $Q_1 \leq 0.95Q_{max}$ & $Q_2 \geq 0.05Q_{max}$。因为 Bed1 中的固体 CO_2 超过它能容纳的最大固体 CO_2 含量之后，就不能再进行吸附；同理 Bed2 中固体 CO_2 的量若过少，

也不能进行 CO_2 的解吸附过程。

3)模式之间的转换条件

分析 Mode1 和 Mode2 的约束条件时可以看出,当不能满足模式中的约束时就需要转换到下一个模式。亦即,若 Bed2 中的 CO_2 小于一定值(5.5g/m³)时,就不能再进行空气补给而开始进行 CO_2 解吸附过程;当 Bed1 中的固体 CO_2 含量已经很高或 Bed2 中的固体 CO_2 含量很低时,两者互换角色,Bed1 开始进行 CO_2 补给而 Bed2 进行吸附。具体的模式转换详见图 2。

根据图 2 可以得到完整的 VCCR 简单模型,只需把 $f_i(V,\rho,Q,r,z)$ 替换成具体每个模式的一组微分方程即可,然后依照前面给出的 *LifeSupport* 混成程序结构写成混成程序。该模型即要验证在 Bed1 和 Bed2 都在负荷内工作的情况下,船员舱的氧气和二氧化碳含量能维持在安全水平。

4.2 复杂的 VCCR 模型

在前面的简单的 VCCR 模型中,我们只考虑了 CO_2 的含量。但是在实际情况中,还需要考虑 O_2 的含量,同时也需要考虑 CO_2 和 O_2 的反馈控制情况。首先我们引入一些新的变量如下:

z_C:船员舱内 O_2 的密度(g/m³);

z_1,z_2:Bed1 以及 Bed2 内 O_2 的密度(g/m³);

m_{O_2}:船员舱内补充 O_2(也即从 Buffer 流入船员舱)的速率(g/hr);

r_{O_2}:船员舱内 O_2 产生的速率(g/hr);

ref_{CO_2},ref_{O_2}:船员舱内预期的 CO_2 和 O_2 的密度(g/m³)。

我们假设简单 VCCR 模型中 CO_2 的补充速率为一个常量,但是这是不符合实际应用的。现在,我们使用 PID 控制器(比例-积分-微分控制器)[1] 原理根据历史数据和差别的出现率来调整 CO_2 和 O_2 的补充速率(也即 m_{CO_2} 与 m_{O_2})。PID 控制器由比例单元 P、积分单元 I 和微分单元 D 组成,分别通过 k_p,k_i 和 k_d 3 个参数来设定。PID 控制器是一个在工业控制应用中常见的反馈回路部件,主要适用于基本上线性且动态特性不随时间变化的系统。这个控制器把收集到的数据和一个参考值进行比较,然后把这个差别用于计算新的输入值,这个新的输入值可以让系统的数据达到或者保持在参考值。根据上面的介绍以及 PID 控制器的转移函数,得到 m_{CO_2} 与 m_{O_2} 的新的定义:

$$m_{CO_2}{}' = k_p \cdot \rho_C{}' + k_d \cdot \rho_C{}'' + k_i \cdot (\rho_C - ref_{CO_2})$$

$$m_{O_2}{}' = k_p \cdot z_C{}' + k_d \cdot z_C{}'' + k_i \cdot (z_C - ref_{O_2})$$

同时,对于各个模式也需要加入与 O_2 相关的微分方程,各个模式的约束与简单模型中的约束相同。下面是复杂 VCCR 模型 Mode1 和 Mode2 的微分方程,Mode3 和 Mode4 与前面相同,也只需要相应改变 Bed1 和 Bed2 的角色即可。

Mode1:Bed1 吸附阶段,Bed2 空气补给阶段

$$V_C \cdot \rho_C{}' = r_1 \cdot \rho_1 - r_1 \cdot \rho_C + r_2 \cdot \rho_2 + m_{CO_2} + r_{CO_2}$$

$$V_1 \cdot \rho_1{}' = r_1 \cdot \rho_C - r_1 \cdot \rho_1 - r_{ads}$$

$$V_2 \cdot \rho_2{}' = -r_2 \cdot \rho_2$$

$$Q_1{}' = r_{ads}$$

$$Q_2{}' = 0$$

$$V_C \cdot z_C{}' = r_1 \cdot z_1 - r_1 \cdot z_C + r_2 \cdot z_2 + m_{CO_2} - r_{O_2}$$

$$V_1 \cdot z_1{}' = r_1 \cdot z_C - r_1 \cdot z_1$$

$$V_2 \cdot z_2{}' = -r_2 \cdot z_2$$

$$m_{CO_2}{}' = k_p \cdot \rho_C{}' + k_d \cdot \rho_C{}'' + k_i \cdot (\rho_C - ref_{CO_2})$$

$$m_{O_2}{}' = k_p \cdot z_C{}' + k_d \cdot z_C{}'' + k_i \cdot (z_C - ref_{O_2})$$

Mode2:Bed1 吸附阶段,Bed2 解吸附阶段

$$V_C \cdot \rho_C{}' = r_1 \cdot \rho_1 - r_1 \cdot \rho_C + m_{CO_2} + r_{CO_2}$$

$$V_1 \cdot \rho_1{}' = r_1 \cdot \rho_C - r_1 \cdot \rho_1 - r_{ads}$$

$$V_2 \cdot \rho_2{}' = -r_2 \cdot \rho_2 - r_{ads}$$

$$Q_1{}' = r_{ads}$$

$$Q_2{}' = -r_{des}$$

$$V_C \cdot z_C{}' = r_1 \cdot z_1 - r_1 \cdot z_C + m_{CO_2} - r_{O_2}$$

$$V_1 \cdot z_1{}' = r_1 \cdot z_C - r_1 \cdot z_1$$

$$V_2 \cdot z_2{}' = -r_2 \cdot z_2$$

$$m_{CO_2}{}' = k_p \cdot \rho_C{}' + k_d \cdot \rho_C{}'' + k_i \cdot (\rho_C - ref_{CO_2})$$

$$m_{O_2}{}' = k_p \cdot z_C{}' + k_d \cdot z_C{}'' + k_i \cdot (z_C - ref_{O_2})$$

各个模式之间的转换条件也与 VCCR 简单模型中相同。因此同样对照图 2,将通过 PID 控制器得到的两个公式以及与 O_2 相关的微分方程添加到 VCCR 简单模型的混成程序中,就可以得到包含 CO_2 和 O_2 含量以及反馈控制的精化的 VCCR 模型。在复杂模型中,本文只考虑了氧气和二氧化碳的气体含量情况,还可以使用同样的方式对惰性气体等的含量进行建模和验证,对模型进行进一步精化。

5 VCCR 设计的安全性验证

5.1 KeYmaera 简介

KeYmaera[7,8] 是一种综合演绎推理、实代数理论以及计算机代数证明器并且提供自动以及交互式验证混成系统的工具。它支持微分动态逻辑,并且实现了微分动态逻辑、微分代数动态逻辑以及微分时序动态逻辑的证明演算。KeYmaera 基于证明器 KeY 开发,提供自带的微分方程和实数运算函数。对于量词消去、微分方程以及符号计算,KeYmaera 提供了使用 Mathematica 的接口,也可以使用 Redlog、QEPCAD 作为实数量词消去的工具。KeYmaera 的具体结构详见图 3。它同时提供 Eclipse 插件方便代码编写。本文使用 KeYmaera 来验证所建立的混成系统模型是否满足我们定义的安全性质。

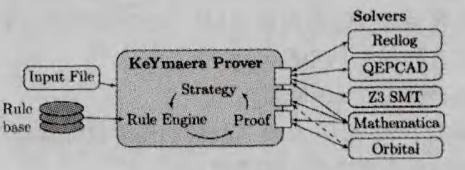

图 3 KeYmaera 的结构

KeYmaera 的基本工作原理是利用微分动态逻辑、微分代数动态逻辑以及微分时序动态逻辑的证明演算公式(详见文献[3])对混成程序进行化简和分析,并且利用半代数几何、柱形代数分解、Groebner 基以及半定规划等原理进行量词消去及性质证明。

KeYmaera 中混成系统的模型一般遵循图 4 所示的模板来建造,更加详细的 KeYmaera 语法可以参见 KeYmaera 官方教程[7,8]。在其中我们要定义变量、初始状态、离散跳变、连续演化和整个混成过程满足的条件等。

```
\problem{
\[
  R t;/*变量的定义*/
\]
(
  t>=0 /*初始状态的描述*/
  ->\[
    t:=0;/*离散跳变*/
    {t'=1} /*连续演化,微分方程描述*/
  \](t>=0) /*整个过程满足的条件*/
  )
}
```

图 4　KeYmaera 建模的一般模板

5.2　验证过程与结果

实验环境是 KeYmaera3.2 及 Mathematica 8。我们将第 4 节提出的简单 VCCR 模型和复杂的 VCCR 模型转化为符合 KeYmaera 语法的混成程序,在 KeYmaera 中进行了验证。在此需要说明的是,我们根据实际情况以及 VCCR 系统的设计情况[6]对模型中的一些参数给定符合实际的值。

初始状态,各个舱内 CO_2 和 O_2 的密度含量:

$\rho_C=9.13;\rho_1=9.13;\rho_2=5.56;z_C=21;z_1=0;z_2=0;Q_1=20;Q_2=200;Q_{max}=500$

CO_2 和 O_2 的密度的上限和下限值:

$co2lower=5.3867;co2upper=9.5865;o2lower=20.5425;o2upper=23.1902$

PID 控制器中各个参数的值:

$k_p=8;k_i=9.0001;k_d=1$

图 5 是简单 VCCR 系统的验证结果,左边的 Proof Tree 窗口中显示了证明的过程,Proof Closed 窗口中"Property proved"说明验证成功,Proof Statistics 窗口中列举了证明过程的各种数据,包括整个证明过程所用的时间、证明节点的个数、证明分支的个数、使用的内存等等。类似地,图 6 是复杂 VCCR 系统的验证结果,同样 Proof Closed 窗口中"Property proved"说明证明成功。对比图 5 和图 6 中的 Proof Statistics 发现,复杂 VCCR 模型比简单 VCCR 模型自动验证花费的时间更长,证明节点和证明分支更多,这也与形式化验证的一般常识相符。这两个图最终可以说明在如上的初始条件和参数设定下,整个 VCCR 系统的运行过程始终能满足我们所要验证的安全性质:CO_2 和 O_2 的含量始终保持在定义的上界和下界范围内。

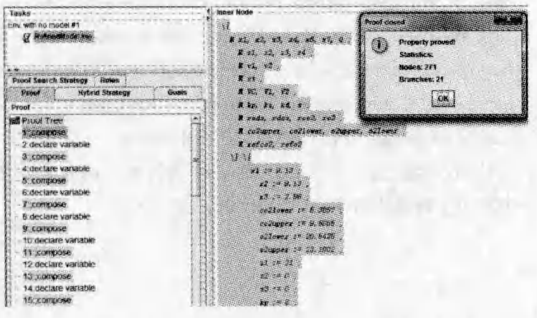

图 5　简单 VCCR 模型的 KeYmaera 验证结果

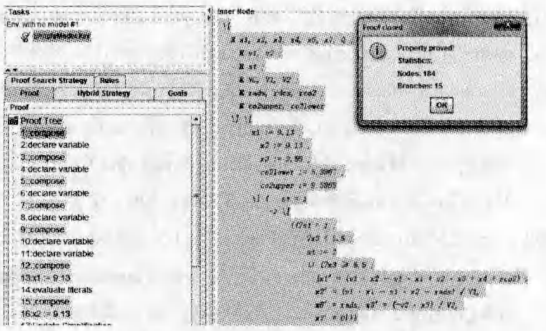

图 6　复杂 VCCR 模型的 KeYmaera 验证结果

结束语　本文利用混成系统的理论以及微分动态逻辑对空间生命支持系统的一个子系统 VCCR 系统进行建模并且验证其安全性。建立了两个模型:首先只考虑 CO_2 的含量情况建立了一个简单的 VCCR 模型;然后考虑 CO_2 以及 O_2 的含量情况,并且利用经典的 PID 控制原理建立了更加贴近实际情况的复杂的 VCCR 模型。最后用 KeYmaera 工具验证了这两个模型均符合预先定义的安全性质。在后面的工作中可以考虑对整个空间生命支持系统 ARS 进行安全性验证,这会涉及到多个子模块的通信等更多问题,建模的过程会更加复杂,也会更加贴合实际的情况。

参 考 文 献

[1] Astrom A J, Wittenmark B. Computer Controller Systems-Theory and Design(Third Edition)[M]. Prentice Hall/Pearson, 2001

[2] Platzer A, Quesel J D. European Train Control System: A Case Study in Formal Verification [R]. SFB/TR 14 AVACS 54, 2009

[3] Platzer A. Logical Analysis of Hybrid Systems: Proving Theorems for Complex Dynamics [M]. Springer, 2010

[4] Platzer A. The complete proof of theory of hybrid systems [C]// ACM/IEEE Symposium on Logic in Computer Science, LICS 2012. 2012: 541-550

[5] Schaft A, Schumacher H. An introduction to Hybrid Dynamical Systems [M]. Springer-Verlag, 2000

[6] Subramanian D, Ariyur K, Lamba N, et al. Control design for a hybrid dynamical system: A NASA life support system [C]// LNCS 2993. Springer-Verlag, 2004: 570-584

[7] KeYmaera: A Hybrid Theorem Prover for Hybrid Systems [EB/OL]. http://symbolaris.com/info/KeYmaera.html, 2013-05-13

[8] Guide for KeYmaera Hybrid Systems Verification Tool [EB/OL]. http://symbolaris.com/info/KeYmaera-guide.html., 2012-10-27

[9] Monin J F, Hinchey M G. Understanding Formal Methods[M]. Springer, 2003

[10] Malin J, Nieten J, Schreckenghost D, et al. Multi-agent diagnosis and control of an air revitalization system for life support in space [C] // Proceedings of the IEEE Aerospace Conference. 2000: 309-326

[11] Edmund M C, Orna G, Doron P. Model Checking [M]. Cambridge, MA: MIT Press, 2000

[12] Alur R, Dang T, Ivancic F. Progress on reachability analysis of hybrid systems using predicate abstraction [C]// LNCS 2623. Springer-Verlag, 2003: 4-19

[13] Horowitz R, Varaiya P. Control design of an automated highway system [J]. Proceedings of the IEEE, 2000, 88(7): 913-925

[14] Glavaski S, Papachristodoulou A, Ariyur K. Safety verification of controlled advanced life support system using barrier certificates [C]// Hybrid Systems: Computation and Control, Lecture Notes in Computer Science. 2005: 3414: 306-321

[15] Loos S M, Renshaw D W, Platzer A. Formal verification of distributed aircraft controllers [C]// Hybrid Systems: Computation and Control. Philadelphia, PA, USA, 2013: 125-130

[16] Prajna S. Optimization-based Methods for Nonlinear and Hybrid Systems Verification [D]. California Institute of Technology, 2005

[17] Glover W, Lygeros J. A stochastic hybrid model for air traffic control simulation. LNCS 2993[C]// Hybrid Systems: Computation and Control. Heidelberg: Springer-Verlag, 2004: 372-386

[18] Kouskoulas Y, Renshaw D W, Platzer A. Certifying the safe design of a virtual fixture control algorithm for a surgical robot [C]// Hybrid Systems: Computation and Control. Philadelphia, PA, USA, 2013

[19] Manna Z, Pnueli A. Temporal Verification of Reactive Systems [M]. Springer-Verlag, 1995

(上接第 160 页)

图挖掘算法并行化，提出了基于 MapReduce 模型的大规模导出子图提取算法，其共分为 4 个阶段，每个阶段对应于一个算法：Find_VE、G_F1、FindPartFG、FindAllFG，并将算法应用于自适应云平台中，构成整个挖掘体系。最后通过大量实验可以表明，算法具有可行性、良好的加速性能与运行效率。

未来的工作主要包括：

1)继续优化自适应云端与导出子图提取算法，以进一步提高挖掘性能。

2)从真实环境实验中可以发现，子图同构问题已经严重制约了算法的运行效率，虽然可以通过云计算平台增加计算资源，以达到提高算法运行效率的目的，但并不能从根本上解决子图同构时间复杂度高的问题，特别是在复杂图数据环境下表现尤为明显，因此子图同构问题也将是我们未来持之以恒研究的工作之一。

3)现在是大数据时代，改进现有算法，考虑将自适应云端应用于其它数据类型的挖掘中。

参 考 文 献

[1] 覃雄派，王会举，杜小勇，等. 大数据分析—RDBMS 与 MapReduce 的竞争与共生[J]. 软件学报，2012，23(1)：32-45

[2] TDWI Checklist Report: Big Data Analytics[OL]. http://tdwi.org/research/ 2010/08 Big-Data-Analytics. aspx

[3] 邹兆年，李建中，高宏，等. 从不确定图中挖掘频繁子图模式[J]. 软件学报，2009，20(11)：2965-2976

[4] Zou Xiao-hong, Chen Xiao, Guo Jing-feng, et al. An improved algorithm for mining Close Graph[J]. ICIC Express Letters Journal of Research and Surveys, 2010, 4(4): 1135-1140

[5] 薛冰，张俊峰，郑超. 基于分割图集的频繁闭图挖掘算法[J]. 计算机应用研究，2011，28(1)：61-64

[6] Guo Jing-feng, Chai Ran, Li Jia. Top-down algorithm for mining maximal frequent subgraph[J]. Advanced Research on Industry, Information System and Materials Engineering, 2011, 204-210: 1472-1476

[7] 刘勇，李建中，高宏. 从图数据库中挖掘频繁跳跃模式[J]. 软件学报，2010，21(10)：2477-2493

[8] 刘文艳. 基于深度优先策略的频繁导出子图挖掘算法[D]. 西安：西安电子科技大学，2009

[9] Gupta S, Raman V, Saurabh S. Maximum r-Regular Induced Subgraph Problem: Fast Exponential Algorithms and Combinatorial Bounds[J]. SIAM Journal on Discrete Mathematics, 2012, 26(4): 1758-1780

[10] Lenk A, Klems M, Nimis J, et al. What's inside the cloud? An Architectural Map of the Cloud Landscape[C]// Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing. 2009: 23-31

[11] Son J, Choi H, Chung Y D. Skew-tolerant key distribution for load balancing in MapReduce[J]. IEICE Transaction on Information and Systems, 2012, 95(2): 677-680

[12] Valiant Leslie G. A bridging model for parallel computation[J]. Communication of the ACM, 1990, 33(3): 103-111

[13] Grzegorz M, Matthew A H, Bik Art J C, et al. Pregel: A system for large-scale graph processing[C]// Proceedings of the SIGMOD. Indianapolis, Indiana, USA, 2010: 135-145

[14] Avery C. Giraph: Large-scale graph processing infrastruction on Hadoop[C]// Proceedings of the Hadoop Summit. Santa Clara, 2011

[15] Tyson C, Neil C, Peter A, et al. MapReduce Online[C]// Proceedings of the NSDI. San, Jose, California, USA, 2010: 33-48

[16] Islam S, Gregoire J C. Giving user an edge: A flexible cloud model and its application for multimedia[J]. Future Generation Computer Systems, 2012, 28(6): 823-832

[17] Samba A. Logical data models for cloud computing architectures [J]. IT Professional, 2012, 14(1): 19-26

[18] Huang H, Wang L Q. P&P: A combined Push-Pull model for resource monitoring in cloud computing environment[C]// Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing. Miami, Florida, USA, 2010: 260-267

[19] Tsai Wei-Tek, Sun X, Balasooriya J. Service-oriented cloud computing architectued[C]// Proceedings of the 2010 Seventh International Conference on Information Technology: New Generations. Las Vegas, NV, USA, 2010: 684-689

[20] 王桂娟，印鉴，詹卫许. GC-BES：一种新的基于嵌入集的图分类方法[J]. 计算机科学，2012，39(6)：155-158