基于微分动态逻辑的铁路道口控制分析

钱 磊 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 利用微分动态逻辑对铁路道口控制进行形式化分析与建模。在火车从发送接近信号到进入道口的运动过程中,根据火车到达道口时间上的要求,将火车速度控制问题抽象成一个混成系统的安全性性质,用微分动态逻辑来描述,并使用混成系统证明工具 KeYmaera 对系统的安全性进行验证,以实现对火车进入道口前速度的正确控制。

关键词 微分动态逻辑,铁路道口控制,混成系统,形式化验证

中图法分类号 TP302.2 **文献标识码** A

Modeling and Analysis of Railway Crossing Based on Differential Dynamic Logic

QIAN Lei YU Wen-sheng

(Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai 200062, China)

Abstract We presented the analysis of railway crossing based on differential dynamic logic. When a train sends approaching signal and goes into the crossing, the limit of time spending in this period helps us identify safe ranges for the train speed control. We also illustrated modeling of this hybrid system using hybrid program and differential dynamic logic. Using the theorem prover KeYmaera, we formally verified hybrid safety properties of Railway Crossing about the train, and obtained the right speed control condition.

Keywords Differential dynamic logic, Railway crossing control, Hybrid system, Formal verification

1 引言

混成系统是连续的实时动态系统和离散事件系统的混合体,这些连续与离散的动力学行为不仅共存,而且相互作用。混成系统的形式化验证是一个复杂的过程[1]。确保复杂物理过程的正确性是计算机科学、数学和工程上的一个具有挑战性的重要问题。对于交通控制,安全性是一个确定各个特殊条件参数的过程,如高速公路上汽车的控制[2,3],轨道交通的控制[4],飞机调度的避撞控制[5]等。对于混成系统复杂性质的正确性验证,混成自动机的模型检验技术常常会带来状态爆炸的问题[6]。而微分动态逻辑,结合了合成验证的技术,着重于将问题模块化,这是混成自动机所欠缺的,其实质是一种分而治之的方法,可以将问题形式化地分解成多个子问题,独立进行验证,加快了自动化验证的速度[6]。

本文利用微分动态逻辑对火车进道口的过程给出了形式化的分析与建模,得出火车在进道口场景下的正确控制策略,并用混成系统证明工具 KeYmaera[7] 对火车进道口模型的安全性进行了验证。火车在接近道口时,会发送一个信号给控制台,通知控制台关闭道口,不让其他车辆通行。火车发送信号到通过道口的过程在时间上有一定的要求,不能在道口关闭前就到达道口,即到达的时间不能太短,否则容易发生交通事故,其次,此过程的时间不能过于冗长,否则两边的车辆等待时间会太长而有碍交通。在这一时间上的规范约束下,火车在运动过程中速度的控制是我们研究的关键。通过建模分析,可根据火车不同阶段的速度实施加速或减速策略,使得火车在经过道口过程中满足时间上的要求,确保此混成系统的安全性,并用混成系统证明工具 KeYmaera 对模型的安全性进行了验证,实现对火车进入道口前速度的正确控制。

2 微分动态逻辑

微分动态逻辑(differential dynamic logic, dL)是对混成系统的规范和约束进行验证的一种逻辑语言。在微分动态逻辑中,混成系统必须要用混成程序(hybrid program)来描述。对混成系统中有关性质描述的规范条件及其正确性的验证可以用微分动态逻辑的演算算子来公式化。dL 和它的验证演算是工具 KeYmaera[7] 对混成系统推导、验证的基础。混成系统以及在逻辑中公式化的性质都可以包含符号变量,这些变量可以是自由变量也可以是能够从中发现安全性条件的量词变量。

微分动态逻辑是建立在混成系统上的一阶逻辑,具有描述混成系统性质的能力,可以简洁地表达出混成系统的安全性的性质。对于混成系统 α,dL 可以提供对于其性质正确性

到稿日期:2012-12-08 返修日期:2013-03-11 本文受国家自然科学基金项目(61070048),国家自然科学基金委员会创新研究群体科学基金项目(61021004),国家"863"计划项目(2011AA010101),国家"973"计划项目(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。

钱 磊(1988—),男,硕士生,主要研究领域为形式化建模、物理信息融合系统,E-mail:nicky_qianlei@163.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。

的描述，形如$[\alpha]\phi$，表示混成系统α中的每一步的状态都满足条件ϕ。如果dL对系统性质的陈述为：$\phi\rightarrow[\alpha]\psi$，则表示如果在初始状态满足条件$\phi$，则可以得出在系统的每个状态都满足条件$\psi$。dL的陈述还可以是$\langle\alpha\rangle\phi$的形式，$\langle\rangle$算子代表在整个混成系统演化过程中存在一个状态满足$\phi$。

微分动态逻辑具有详细的推理证明演算法则，实质上是一种便于证明的标准逻辑公式。一段公式形如$\Gamma\vdash\Delta$，前提条件为Γ，而结论为Δ。此公式的语义等同于$\bigwedge_{\phi\in\Gamma}\phi\rightarrow\bigvee_{\psi\in\Delta}\psi$。证明演算就是一种方法法则，可以推理出公式正确或错误的性质。dL法则是基于一阶逻辑法则的扩展，加入了针对量词消去、微分方程的替换规则[6]，由于混合了$[]$，$\langle\rangle$的运算，需要消去这些运算从而转化为基本的一阶逻辑公式，如下面的式(1)，自底向上说明了在离散的过程中，$\langle x_1:=\theta_1,\cdots,x_n:=\theta_n\rangle\phi$可以表示成在公式$\phi$中将$x_n$替换成$\theta_n$，从而消去了$\langle\rangle$算子。具体更多的法则可以参考文献[6,8]。

$$\frac{\phi_{x_1}^{\theta_1},\cdots,\phi_{x_n}^{\theta_n}}{\langle x_1:=\theta_1,\cdots,x_n:=\theta_n\rangle\phi}\tag{1}$$

上述的混成系统可以用混成程序(Hybrid Program)[8]来进行描述。混成程序是一种规范的应用于混成系统形式化的模型。混成程序可以将混成系统的离散跳变和连续变化相结合，实现既简洁又准确的形式化、结构化的控制。混成程序提供了3种结构：离散跳变集合、微分方程系统和控制结构。离散跳变集包含了系统的离散变化，对状态变量(实质是微分方程中的变量)进行瞬时的赋值操作。微分方程系统表示了动态系统的连续性变化及其变化域。控制结构表示了离散和连续之间转移的关系，利用操作符($\cup$，*，;)可以连接不同的混成程序，实现了系统间的联系和交互。

下面总结了混成程序的表述形式以及对应的作用效果。

(1)$x_1:=\theta_1,\cdots,x_i:=\theta_i$：离散的跳变集，同时将值$\theta_1,\cdots,\theta_i$分别赋值给$x_1,\cdots,x_i$。

(2)$x_1'=\theta_1,\cdots,x_i'=\theta_i\ \&\ \chi$：连续演化过程。由项$\theta_1,\cdots,\theta_i$组成的关于$x_1,\cdots,x_i$的微分方程，并且带有一阶逻辑的限制条件$\chi$(演化域)。

(3)$?\chi$：状态检测。检测当前状态的一阶逻辑范式χ的值是否为真。

(4)$\alpha;\beta$：顺序组合。当HPα完成后再顺序执行HPβ。

(5)$\alpha\cup\beta$：不确定性选择。在HPα和HPβ两者间任选其一执行。

(6)α^*：不确定性重复。重复执行HP$\alpha$$n$次，$n$可以是任意自然数。

详细的微分动态逻辑的介绍可以参见文献[6]。

3 铁路道口控制问题

铁路道口控制系统可描述为“火车”、“道闸”、“控制器”这样3个系统的组合。火车至少在到达道口之前2分钟向控制器发送一个信号，接着控制器向道口发送信号，闭合道闸。在火车通知接近道口后最多5分钟，火车离开[1]。

只考虑火车和控制器，将火车、道口各自简化为一个点(不考虑火车自身的长度)。可以得出这样一个场景：火车到达发送信号的点时，发送接近的信号，之后最少2分钟且最多5分钟经过道口，即不能快于2分钟，以免道闸没来得及关闭就到达道口点，也不能使得道闸持续关闭的时间超过5分钟，以免影响正常的交通。

针对上面的场景，我们对系统分别进行建模。

(1)火车模型

我们假设火车的位移为s，火车的速度为v，加速度为a。得出火车运动的一个连续演化过程：$s'=v,v'=a$。火车可以分为加速行驶和刹车减速两个状态。假设加速行驶的加速度为A，刹车减速的加速度为$-B$。

(2)控制模型

在控制器方面，对火车速度的控制，可以根据到达道口剩余路程与剩余的时间确定火车允许行驶的平均速度来衡量。假设发送信号点距离道口的距离为固定不变的值SG(start sending signal)，火车要能保证在2至5分钟经过道口，速度必须时刻满足$(\forall t)$：

$$(SG-s)/(300-t)\leqslant v\leqslant(SG-s)/(120-t)$$

式中，$SG-s$表示剩余的路程，$300-t$表示若5分钟到达所剩余的时间值。此处t为发送接近信号后火车所行驶的时间。t的初始值为0，单位为秒。

根据上述分析，用混成程序可表示出火车发送信号后的运动演化过程：

```
Train≡
  t:=0;
  ((if(t≤120) then
  (?(v>(SG−s)/(120−t));
    a:=−B;{s'=v,v'=a,t'=1,v≥(SG−s)/(300−t)&s≤SG&t
          >0&t<120&v>0})∪(?(v<(SG−s)/(300−t));
    a:=A;{s'=v,v'=a,t'=1,v≤(SG−s)/(120−t)&s≤SG&t>
         0&t<120&v>0})∪(?(v<(SG−s)/(120−t)&v≥
         (SG−s)/(300−t));
    a:=A;{s'=v,v'=a,t'=1,v≤(SG−s)/(120−t))&v≥(SG−
         s)/(300−t)&s<=SG&v>0&t>0&t<=120})
  else
    a:=A;{s'=v,v'=a,t'=1,v≥0&s≤SG}fi)*
```

上述混成程序中，{}中的公式表示了火车运动的连续演化过程，符号:=表示了对状态变量赋值的离散事件。令火车发送信号后经过的时间为t，整个过程只考虑从火车发送信号时到经过道口的阶段。在火车发出信号后的2分钟内，即$t\leqslant120$，此时在t时刻，火车必须对速度进行适当的控制，使得速度不能太快以确保2分钟后到达道口；同时，速度也有下限，火车必须要满足要求，即在5分钟之内经过道口。所以当速度$v>(SG-s)/(120-t)$时，就要进行刹车减速操作。而当速度$v<(SG-s)/(300-t)$时，就要进行加速操作。如果速度满足$(SG-s)/(300-t)\leqslant v\leqslant(SG-s)/(120-t)$的情况，也让其加速以做到在满足条件的情况下充分快地通过道口。在$t\leqslant120$时，3种情况具有不确定选择的关系，所以用控制操作符$\cup$来连接，对于速度的判断用了控制操作符?。同时在上述演化过程中，火车的位移总是不大于总距离SG，即$s\leqslant SG$。最

后，当 $t>120$ 时，即超过了 2 分钟，火车可以一直加速行驶下去。整个系统加上了 $*$ 算子，表示可以 $n(n\geqslant 0)$ 次重复执行。

这样火车从发送信号到进入道口的一段演化过程所组成的混成系统，就可以用上述的混成程序来描述。

(3)安全性

在上述整个系统的变化中，规范要求使得火车能够在 2 至 5 分钟内到 s 达道口。用 dL 描述为：

$$[Train](s=SG\rightarrow 120\leqslant t\leqslant 300) \tag{2}$$

式(2)表示：在火车发送信号后的整个演变过程中，当火车到达道口，$s=SG$ 条件满足时，此时所经历的时间满足条件 $2*60\leqslant t\leqslant 5*60$。

(4)初始条件以及参数的可控性

在火车运动的混成程序中，有一些参数具有实际的意义，需要规定初始参数的范围，用 *initial* 来表示：$initial\equiv SG>0\&v>0\&B>0\&A>0\&s\leqslant SG$。火车发信号的点距离道口的距离要大于 0；火车的速度要大于 0；刹车的加速度为 $-B$，B 要大于 0；同时加速度 A 大于 0；初始状态，火车行驶的距离 s 要小于 SG。

在初始状态时，为确保火车的可控性，对火车的速度下界还有特殊的限制。假设火车一开始进入混成程序运行时就需要减速，而由于其速度过快，初始速度值远大于安全值，即使整个过程都处于减速状态也不足以减速到符合的速度条件，因此必须对初始速度增加一个上限值：$v*120\leqslant SG+\frac{1}{2}*B*120^2$。同理，能够求出初始速度的一个下限值：$v*300\geqslant SG+\frac{1}{2}*A*300^2$。为了火车能够安全地行驶，火车到达发送信号点时，规定火车初始速度要满足时间上的要求，在可控的范围内，即在 $t=0$ 时，$SG\leqslant v*300$，$v*120\leqslant SG$。

整合所有的条件限制，可以得出完整的初始条件：

$$initial\equiv SG>0\&v>0\&B>0\&A>0\&s\leqslant SG\&SG\leqslant v*300\&v*120\leqslant SG$$

综上，可以得出整个火车道口控制完整的安全性条件：

$$initial\rightarrow[Train](s=SG\rightarrow 120\leqslant t\&t\leqslant 300) \tag{3}$$

安全性条件式(3)表示了在规定的初始条件 *initial* 下，火车只要根据混成系统 *Train* 的控制流程进行行驶，那么在整个运动过程中，各个状态始终就会满足条件($s=SG\rightarrow 120\leqslant t\leqslant 300$)，即到达道口时所花的时间在 2～5 分钟。下面，就需要着重对此安全性进行形式化的证明，以确保得出的 Train 的控制模型是安全可靠的。

4 安全性证明

为了证明用微分动态逻辑描述的混成系统安全性质的正确性，我们使用了工具 KeYmaera[6]。

KeYmaera 是一个支持微分动态逻辑的混成系统的自动证明器。它结合了推理演绎、实数代数以及计算机代数证明的技术，是对混成系统特殊规范以及性质验证的自动化交互的理论证明器。KeYmaera 支持微分动态逻辑，将微分动态逻辑公式写入后缀名为 key 的文件，就可以导入到 KeYmaera 软件中进行自动的证明。KeYmaera 结合了推理理论证明器 KeY[9] 和计算机代数的技术，包括一些量词消去的工具，如 QEPCAD，Mathematica 等。KeYmaera 的整体结构如图 1 所示[6]。

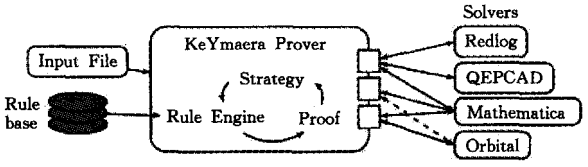

图 1 KeYmaera 的结构

通过第 3 节的分析，结合微分动态逻辑以及 KeYmaera 文件的特殊书写格式，我们编写了下面的验证安全性的 Key 程序。见程序 1。

程序 1 铁路道口控制的安全性验证的 Key 程序

```
\problem {
\[
  R SG;R v;R B;R A;R t;R s;R a;
\]
(
  SG>0&v>0&B>0&A>0&s≤SG
      &s=0&t=0
      &SG≤v*300
      &v*120≤SG
  →\[
  t:=0;
  (
  if(t≤120) then
      (?(t>SG-s/(120-t));
      a:=-B;
    {s'=v,v'=a,t'=1,v≥(SG-s)/(300-t) &s≤SG&t>0&t<120&v>0}
    )
    ++
    (?(v<(SG-s)/(300-t));
        a:=A;
    {s'=v,v'=a,t'=1,v≤(SG-s)/(120-t)&s≤SG&v>0&t>0&t<120}
    )
    ++
    (?(v≤(SG-s)/(120-t))&v≥(sG-s)/(300-t));
        a:=A;
    {s'=v,v'=a,t'=1,v≤(SG-s)/(120-t) &v≥(SG-s)/(300-t)&s≤SG&v>0&
     t>0&t<120}
)
else
        a:=A;
    {s'=v,v'=a,t'=1,v≥0&s≤SG}fi)*
    @invariant(s≤SG&v>0&(v*(120-t)≤SG-s)&(v*(300
      -t)≥SG-s))                                    (*)
    \](s=SG→120≤t&t≤300)
  )
  }
```

\problem 表示整个问题的声明，然后再对涉及到的变量

进行声明，R 表示变量为实数。在\[\]内，是用 dL 公式所表示的火车进入道口的混成系统，主要用混成程序来描述。其中(*)处的@invariant 的标签，其括号内的逻辑公式为整个证明所需要的不变式条件，表示整个演化过程，火车的位移 s 小于等于整个信号点到道口的距离 SG 并且每时每刻的速度始终保持在安全行驶的平均速度之内。

这里证明主要用到了微分动态逻辑中的 ind[8] 法则来对程序中的循环进行简化。具体法则如下：

$$\frac{\Gamma \vdash \phi, \Delta \quad \Gamma \vdash \forall^{\alpha}(\phi \rightarrow [\alpha]\phi), \Delta \quad \Gamma \rightarrow \forall^{\alpha}(\phi \rightarrow \psi), \Delta}{\Gamma \vdash [\alpha^{*}]\psi, \Delta} \quad (4)$$

使用此法则，必须找出其中的不变量 ϕ，使其满足 3 个条件：

(1)初始条件可以推出 ϕ，保证了在循环次数为 0 的时候能满足条件。

(2)在整个系统循环执行过程中，系统能够保持不变量 ϕ 成立。即在系统 α 所有执行状态中，每执行完任何一步都还能满足 ϕ。

(3)条件 ϕ 能推导出 ψ。

以上 3 个条件都成立的 ϕ 是我们要寻求的不变式。这样就可以使用式(4)来对循环操作进行简化，加速 KeYmaera 工具的验证速度。

通过第 3 节的分析，我们将 ϕ 定义为从火车发送信号后到经过道口时速度的约束限制条件。用时间变量 t、位移 s，求出满足条件 2 至 5 分钟内到达的平均速度：

$$(SG-s)/(300-t) \leqslant v \leqslant (SG-s)/(120-t)$$

在 key 程序中，我们用@invariant 标识来表示，为了避免临界条件对证明复杂度的提高以及分母临界点的问题，我们将条件 ϕ 重写，去除分母($120<t<300$)，同时加上 $s \leqslant SG \& v>0$，定义为：

$$\phi \equiv s \leqslant SG \& v>0 \& (v*(120-t) \leqslant SG-s) \& (v*(300-t) \geqslant SG-s))$$

将程序 1 描述的 Key 文件导入到证明器 KeYmaera 中，导入结果如图 2 所示。点击 start 按钮，KeYmaera 会自动对输入的问题进行证明。同时，可能有些步骤需要人为地添加一些微分动态逻辑的法则[8]，这样能够加快 KeYmaera 证明器的证明速度。证明结束会弹出“property proved”对话框，证明了系统性质的正确性。最终证明结果如图 3 所示。

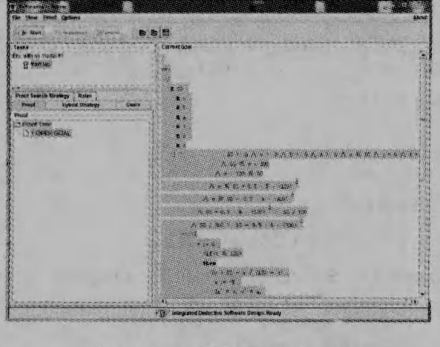

图 2　Key 文件导入 KeYmaera

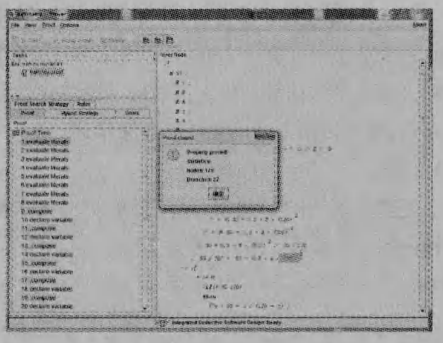

图 3　KeYmaera 证明结果

结束语　综上，对于铁路道口控制问题，按照火车进入道口时间上的要求，引申为火车在运动过程中自身速度控制的问题。结合了微分动态逻辑和混成程序，将火车从发送接近信号到到达道口的过程进行形式化建模，并分析得出安全性性质。最后使用混成系统的证明工具 KeYmaera，通过形式化的验证，得出正确的结论，证明了在火车运动过程中对其速度的正确控制。只要符合此速度控制，就能确保火车进入道口的时间符合规定的要求，从而达到铁路道口的安全和通畅。

参 考 文 献

[1] Schaft A V D, Schumacher H. An Introduction to Hybrid Dynamical System [M]. London: Spring-Verlag, 2000

[2] Loos S M, Platzer A. Safe intersections: At the crossing of hybrid systems and verification[C]// 14th International IEEE Conference on Intelligent Transportation Systems, 2011. Washington, DC, USA: Kyongsu Yi, 2011: 1181-1186

[3] Loos S M, Platzer A, et al. Adaptive cruise control: Hybrid, distributed, and now formally verified [C] // 17th International Symposium on Formal Methods, 2011. Limerick, Ireland: Springer, 2011, volume 6664 of LNCS, 42-56

[4] Platzer A, Quesel J. Logical verification and systematic parametric analysis in train control: Hybrid Systems: Computation and Control[C] // 11th International Conference, 2008. St. Louis, USA: Springer, 2008, volume 4981 of LNCS, 646-649

[5] Platzer A. Differential-algebraic dynamic logic for differential-algebraic programs [J]. Journal of Logic and Computation, 2010, 20(1): 309-352

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

[7] Platzer A, Quesel J. KeYmaera: A hybrid theorem prover for hybrid systems: Automated Reasoning[C] // Fourth International Joint Conference, 2008. Sydney, Australia: Springer, 2008, volume 5195 of LNCS, 171-178

[8] Platzer A. Differential dynamic logic for hybrid systems [J]. Journal of Automated Reasoning, 2008, 1(2): 143-189

[9] Beckert B, Hahnle R, et al. Verification of Object-Oriented Software: The KeY Approach [M]. Berlin, Heidelberg: Springer, 2007