软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2015,26(2):380−389 [doi: 10.13328/j.cnki.jos.004788]

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

信息物理融合系统控制软件的统计模型检验*

单黎君[1,2], 周兴社[1], 王宇英[1], 赵 雷[3], 万丽景[3], 乔 磊[3], 陈建新[3]

[1](西北工业大学 计算机学院,陕西 西安 710072)

[2](国家数字交换系统工程技术研究中心,河南 郑州 450000)

[3](北京控制工程研究所,北京 100190)

通讯作者: 单黎君, E-mail: iceslj@163.com

摘 要: 信息物理融合系统常采用嵌入式实时多任务系统作为其控制软件,这类软件的并发和非确定性给验证带来了困难.提出了一种利用统计模型检验技术分析多任务系统的功能正确性的方法.该方法构造的时间自动机模型以模块化的方式描述了实时多任务系统中的主要成分,包括实时操作系统、周期性任务、偶发任务、共享资源以及物理环境,能够展现多任务系统的细粒度的运行过程及其对物理环境的实时响应.应用该方法分析了玉兔号月球车控制软件的一个早期版本,发现了系统运行中出现的一个特殊错误,识别了实际系统出现错误的条件,再现了出现错误的场景.

关键词: 形式化验证;统计模型检验;信息物理融合系统;多任务系统

中图法分类号: TP311

中文引用格式: 单黎君,周兴社,王宇英,赵雷,万丽景,乔磊,陈建新.信息物理融合系统控制软件的统计模型检验.软件学报, 2015,26(2):380−389. http://www.jos.org.cn/1000-9825/4788.htm

英文引用格式: Shan LJ, Zhou XS, Wang YY, Zhao L, Wan LJ, Qiao L, Chen JX. Statistical model checking of cyber-physical systems control software. Ruan Jian Xue Bao/Journal of Software, 2015,26(2):380−389 (in Chinese). http://www.jos.org.cn/1000-9825/4788.htm

Statistical Model Checking of Cyber-Physical Systems Control Software

SHAN Li-Jun[1,2], ZHOU Xing-She[1], WANG Yu-Ying[1], ZHAO Lei[3], WAN Li-Jing[3], QIAO Lei[3], CEHN Jian-Xin[3]

[1](School of Computer Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China)

[2](National Digital Switching System Engineering and Technological Research Center, Zhengzhou 450000, China)

[3](Beijing Institute of Control Engineering, Beijing 100190, China)

Abstract: Cyber physical systems (CPS) typically employ real-time multitasking systems as their control software. This paper proposes an approach to formally analyzing such control software using statistical model checking of UPPAAL. The main contribution of this study is a model in timed automata which modularly describes the major components of a multitasking system. The model supports the analysis of timing-related functional properties as well as schedulability analysis, and can easily be adapted and extended for verifying different properties of various multitasking systems. A case study on an early version of the Chinese Lunar Rover control software shows that the proposed method is able to track down undesired behavior in real-world industrial CPS.

Key words: formal verification; statistical model checking; cyber-physical system; multitasking system

今天,越来越多的物理设备集成了计算与通信功能,以计算部件为控制中心,由通信网络传输数据,通过计算部件与物理设备之间的交互,控制设备的物理过程.这类系统被称为信息物理融合系统(cyber-physical

* 基金项目: 国家自然科学基金(61472327)
收稿时间: 2014-07-15; 修改时间: 2014-10-31; 定稿时间: 2014-11-26

system,简称 CPS),已广泛应用于航空、交通和医疗等安全攸关领域[1].

由于 CPS 需要满足较高的实时性需求,常采用运行于实时操作系统上的多任务系统作为其控制软件.测试此类控制软件面临诸多困难,测试人员难以控制多个任务交织的详细过程,在发现错误时也难以重现导致此错误出现的事件序列,而利用模型检验能够获得系统的可控性和可见性[2].

本文提出一种针对实时多任务系统的形式化分析方法,采用时间自动机构造系统模型,利用统计模型检验(statistical model checking)技术验证与时间有关的功能属性.我们采用此方法分析了玉兔号月球车控制软件的一个早期版本,针对一个已在测试中发现但原因未知的错误,发现了实际系统中导致此错误出现的条件和原因,并验证了对此错误所做修改的有效性.我们构造的多任务系统模型具有一般性和可扩展性,可用于描述和分析不同类型的多任务系统.

1 方法概述

Clarke 等人在文献[3]中指出:由于多数 CPS 规模大且具有随机特征,不可能进行完全的形式化验证,而统计模型检验适用于验证这类系统.统计模型检验将模型检验和统计方法相结合,利用可执行模型的多次相互独立的模拟,估计系统满足某性质的概率,所需资源与系统描述呈大致的线性关系,因而避免了状态爆炸问题,且能够解决不可判定问题[4].与传统模型检验相比,统计模型检验并未穷尽系统状态,因而不是严格意义上的形式化验证.由于统计模型检验结合了形式化验证的严谨性和测试的高效性,非常适合分析航天 CPS 等因复杂度过高而无法实现精确验证的实际系统.

实时多任务系统是一类典型的 CPS 控制软件,验证此类系统面临的主要挑战来源于并发性和随机性:

- 并发性指共享同一个 CPU 的多个任务相互交织的执行过程;
- 随机性既来自 CPS 内部,即操作系统调度任务的不确定性,也来自 CPS 外部,即物理环境中偶发事件带来的不确定性.

为了观察多任务系统的行为,需要通过构造形式化模型对系统获得完全的控制.在文献[5]中,UPPAAL 研究组为 Herschel-Planck 卫星的位置与轨道控制系统构造了时间状态机模型(本文称为 Herschel 模型),并利用 UPPAAL 的统计模型检验功能分析了该系统的可调度性.Herschel 模型描述了基于优先级的抢占调度策略和共享资源管理机制,能够展示多任务系统的并发执行过程.但是 Herschel 模型不足以支持分析 CPS 控制软件的功能正确性,其原因在于:

(1) 缺少对偶发任务的描述,难以准确刻画包含偶发任务的系统;
(2) 在阻塞任务的调度方面,Herschel 模型与常见的实时操作系统的行为不符;
(3) 只能体现任务中各操作所需时间,不能反映操作产生的效果,因而难以支持对多任务系统功能属性的描述和分析;
(4) 缺少对物理环境的刻画,难以体现外部环境中的非确定性事件对 CPS 的影响.

针对上述问题,我们对 Herschel 模型做了扩展和改进,使得新模型不仅支持对 CPS 控制软件的实时性分析,也支持功能正确性分析.我们提出的方法包含以下步骤:

- 首先,将被验证系统分解成实时操作系统、应用任务和物理环境等部分,分别用时间自动机建模各个部分,用同步通道刻画各部分之间的交互关系;
- 然后,用时态逻辑描述被验证性质,利用模型检验工具的统计模型检验功能分析这些性质,并利用模拟器观察特定的执行路径.

利用此方法,能够观察模型的并发执行过程,有助于理解实际系统的运行情况,发现系统中由复杂原因造成的错误.

2 多任务系统的时间自动机模型

本文考虑的多任务系统包含实时操作系统和多个应用任务,这些任务共享同一个 CPU.为了展示多个任务

并发执行的过程,模型需要分别描述调度器和各种类型的任务.

2.1 调度器

操作系统在任务调度和共享资源管理等方面采取的策略会影响多任务系统的行为.本文以基于优先级的抢占调度为例,介绍对调度器的建模.调度器维护一个就绪任务队列,此队列按任务的优先级从高到低排序.图 1 中,*main*(·),*poll*(·),*add*(·)都是 *C* 函数,用于实现对任务队列的操作:*main*(·)根据任务的 ID 给所有任务赋初始优先级,*poll*(·)从任务队列中取队首,*add*(·)向任务队列加入一个任务.

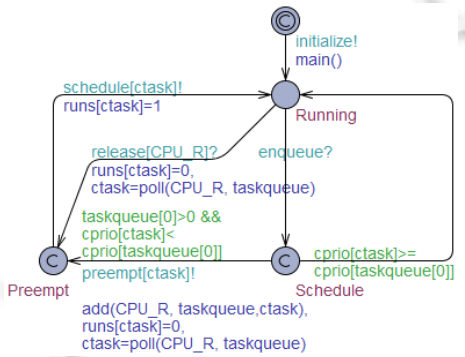


Fig.1 Priority-Based preemptive scheduler

图 1 基于优先级的抢占式调度器

调度器有两种调度时机:

- 其一是有任务加入就绪队列时,比较正在执行的当前任务与队列头的优先级,若当前任务的优先级较低则被抢占并加入就绪队列,队列的头被调度;
- 其二是当前任务释放 CPU 时,调度就绪队列的头.

2.2 周期性任务

应用任务运行于实时操作系统上,实现多任务系统的各种功能.根据触发条件,任务可以划分为周期性任务和偶发任务.周期性任务指时间触发的任务,通常以固定时间间隔到达.CPS 控制软件中的周期性任务可用于实现以固定频率读敏感器数据、计算 CPS 的物理位置、导航等功能.我们将任务建模为参数化的时间自动机,即模板,每个具体的任务都是模板的一个实例.模板的参数包括任务的 *id*(序号)、*offset*(在周期内的开始时间)、*period*(周期)和 *flow*(操作流).操作表示任务程序中的一条或一组语句,每个任务的语句序列描述为一个操作流.定义各种操作的目的是展示被验证系统的行为.在分析实际系统时,如何定义操作的类型取决于需要验证何种系统的何种性质.例如,要分析与数据访问和传输时间有关的属性,可以定义如表 1 所示的 9 种指令.

Table 1 Types, meanings and parameters of the operations

表 1 指令的类型、含义和参数

指令类型	含义	参数
END	无操作(代表程序结束)	–
COMPUTE	计算	*delay*(计算消耗的时间)
LOCK	试图给资源加锁(代表程序中的获取信号量操作)	*res*(被请求的资源)
UNLOCK	释放资源(代表程序中的释放信号量操作)	–
SUSPEND	释放 CPU,等待一段时间	*delay*(等待时间)
COND	条件转移	*trueStep*(条件为真时跳转的步数),*falseStep*(条件为假时跳转的步数),*truePercent*(条件为真的概率)
GOTO	无条件转移	*trueStep*(跳转的步数)
READ	读数据	*res*(被读的消息队列),*dataLength*(数据帧数)
WRITE	写数据	*res*(被写的消息队列),*dataLength*(数据帧数)

图 2 给出了周期性任务的模板.该自动机使用以下数据变量:

(1) *id* 表示任务的标识符;

(2) *job*[*id*]表示任务的执行时间;

(3) 局部时钟 *sub* 表示任务中各个操作的执行时间;

(4) 局部变量 *ic* 表示操作计数器,每个操作执行之后 *ic* 加 1.

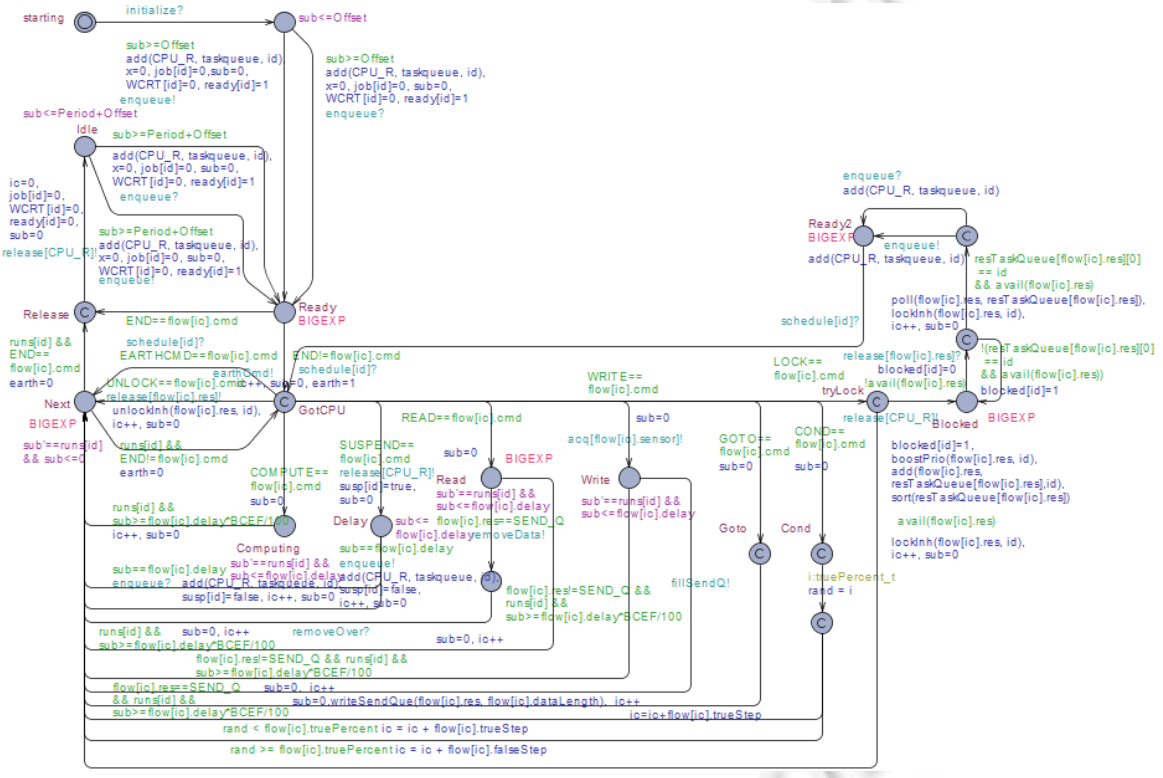


Fig.2 Timed automaton template for the periodic tasks

图 2 周期性任务的时间自动机模型

此自动机的状态迁移体现了任务的执行过程:

- *Starting* 为初始位置;
- 系统初始化之后,任务在 *offset* 时间后加入就绪队列,自动机迁移到 *Ready* 位置;
- 当任务接收到调度器发出的调度通知时,自动机迁移到 *GotCPU* 位置;
- 任务逐一执行操作流中的各个操作,自动机根据操作的类型发生状态迁移.

当任务需要访问某个共享资源而该资源不可用时,任务阻塞并进入等待资源的任务队列,自动机从 *tryLock* 迁移到 *Blocked*.资源可用时,立即被等待队列中优先级最高的任务获取,此任务变为就绪状态,等待被调度,自动机迁移到 *Ready*2 位置.模型为每个共享资源维护一个按任务优先级排序的等待队列,使得模型中操作系统调度任务的行为与实际实时操作系统的行为一致.任务在所有操作执行完之后释放 CPU,等待下一个周期,自动机迁移到 *Idle* 位置.当新的周期到来时,任务开始新一轮执行.

2.3 偶发任务

偶发任务是指由不定期发生的事件触发的任务,这些事件包括来自物理环境的外部事件或来自软件系统的内部事件.偶发任务反映了 CPS 控制软件的一个重要特性:必须实时响应来自环境的随机事件,如操作人员发

来的指令,或物理环境中不定期发生的事件.在实际的多任务系统中,偶发任务由操作系统的中断触发,但是描述中断机制会使模型过于复杂.我们采用 UPPAAL 提供的通道同步机制来体现事件对任务的触发.如图 3 所示,偶发任务的自动机模板类似于周期性任务,主要的区别在于触发条件:当收到某个同步消息时,自动机从初始位置 *Starting* 经过 *Idle* 迁移到 *Ready*,体现了任务被事件触发后进入就绪队列的过程.任务被调度后,自动机迁移到 *GotCPU*,开始逐一执行操作流中的各个操作.任务执行完之后释放 CPU,自动机从 *Release* 迁移到 *Starting*.当再次收到同步消息时,任务被再次触发并进入就绪队列,开始新一轮执行.

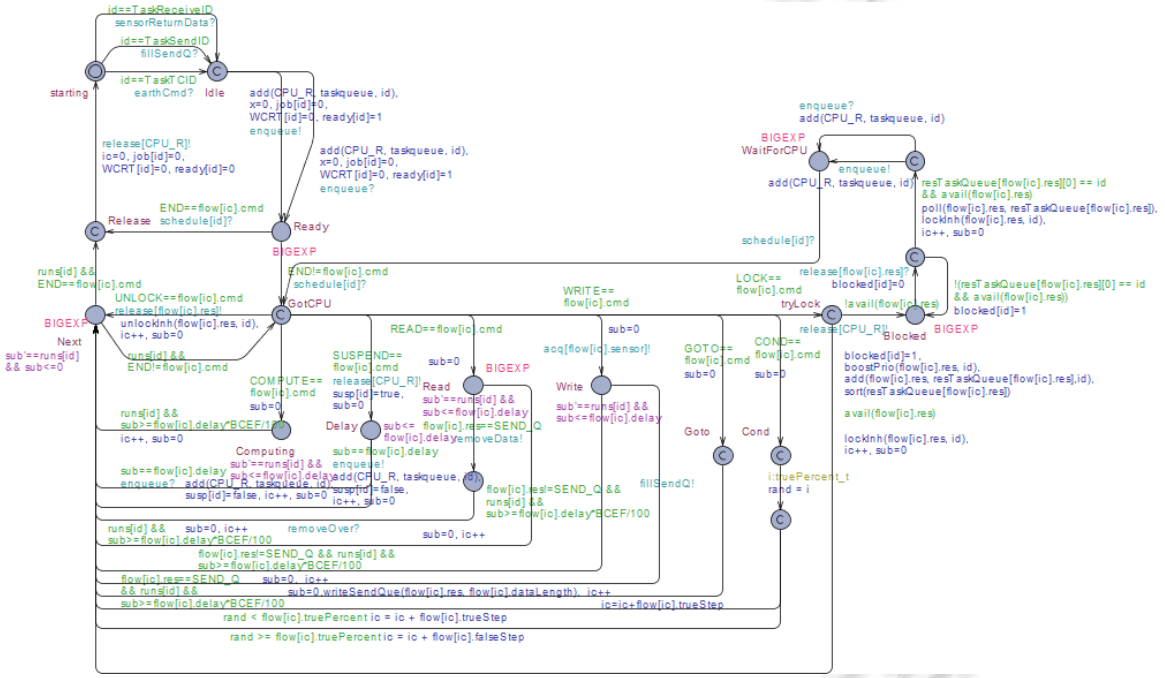


Fig.3 Timed automaton template for the sporadic tasks

图 3 偶发任务的时间自动机模型

3 案例分析

采用上述方法,我们分析了玉兔号月球车控制软件的一个早期版本,目的是定位在测试中发现的一个不明原因的错误.

3.1 问题描述

月球车是一个典型的 CPS 系统,软件子系统作为其控制核心,通过敏感器感知外部环境并做出相应决策,控制月球车在月面上的自主行走.控制软件的结构如图 4 所示:计算系统与敏感器之间通过 CAN 总线连接,敏感器包括 IMU(惯性单元)、APS(太阳敏感器)、Motor(电机控制单元),PayLoad(电控箱)等;任务需要处理来自地面的遥控指令(telecommand,简称 TC)以及向地面发送遥测数据(telemetry,简称 TM),图 4 中,箭头表示数据流.月球车控制软件包含一个自研的实时操作系统和 30 多个应用任务.为了验证与数据传输有关的性质,开发人员对实际系统做了简化,只提取了需要通过 CAN 总线读写数据的 6 个任务.系统中的共享资源包括消息队列 *SendQueue* 和 *ReceiveQueue*,各个任务通过这两个消息队列实现与 CAN 端口的通信:任务向 CAN 端口发送的消息先写入 *SendQueue* 消息队列,再由 *Task*3*Send* 通过 CAN 端口发出;从 CAN 端口返回的数据由 *Task*4*Receive* 接收,先写入 *ReceiveQueue* 消息队列,各任务再从 *ReceiveQueue* 获得数据.各个任务通过信号量实现对消息队列的互斥访问,互斥信号量具有优先级继承功能.

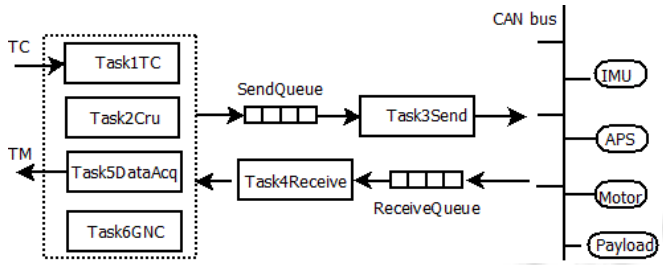


Fig.4 Structure of the rover control software

图 4 月球车控制软件的结构

表 2 总结了 6 个任务的功能和触发条件,表中任务按优先级从高到低排列.

Table 2 Tasks in the rover control software

表 2 月球车控制软件中的任务

任务名	功能	就绪条件/周期(ms)
Task1TC(遥控任务)	接收从地面不定期发送的 4 帧遥控指令,写入 *SendQueue*	有遥控指令
Task2Cru(重要数据管理任务)	将 8 帧重要数据写入 *SendQueue*	5 000
Task3Send(消息发送任务)	将数据队列 *SendQ* 中的全部数据发送到 CAN 总线的端口	数据到达 *SendQueue*
Task4Receive(消息接收任务)	接收 CAN 总线返回的数据并存储到 *ReceiveQueue* 中	CAN 端口有返回数据
Task5DataAcq(请求数据任务)	将“向敏感器请求数据”消息(1 帧)写入 *SendQueue*	200
Task6GNC(导航任务)	制导、巡航与控制	200

在 3 年的开发和测试中,月球车控制软件的开发人员观察到几次“遥测超时错误”:负责以固定频率向敏感器请求遥测数据的任务 *Task5DataAcq* 没有在预期时间内接收到完整的遥测数据.开发人员猜测,遥控指令在特殊时刻到达可能引发此错误.但是此猜测难以通过测试获得确认,原因在于:

- 首先,操作系统的任务调度具有随机性,多任务系统的某次执行难以重现.
- 其次,为了分析多任务系统的执行过程,需要观察实时操作系统对任务的调度,而观察如此细粒度的操作非常困难.
- 其三,此错误出现的次数很少,很难总结错误出现的条件.因此,我们需要用形式化方法来定位此错误.

3.2 时间自动机模型

我们采用第 2 节描述的方法构建了调度器、周期性任务和偶发任务的时间自动机模型.为了验证目标性质,模型还需要描述实际系统中的其他相关部分.由于待验证性质与将 *SendQueue* 中消息发送到 CAN 总线上所需的时间密切相关,需要刻画消息队列 *SendQueue*.图 5 给出了 *SendQueue* 的时间自动机模型.

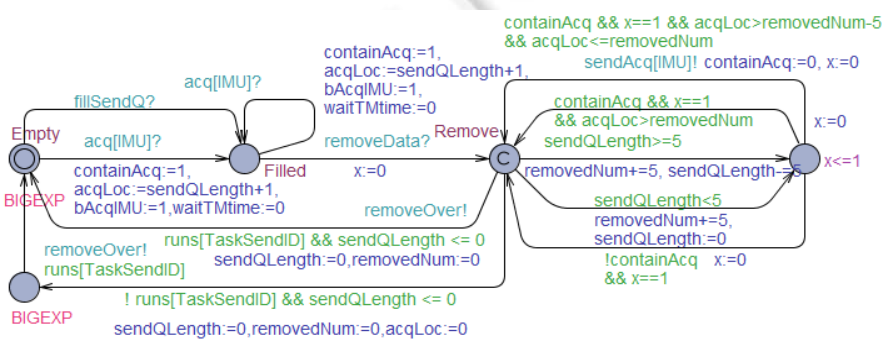


Fig.5 Timed automaton model for *SendQueue*

图 5 *SendQueue* 的时间自动机模型

*Task*3*Send* 读 *SendQueue* 时,以先进先出方式发送队列中的全部数据,发送数据所需时间取决于队列的长度.根据 CAN 总线波特率,传输 1 帧数据需要 0.192ms,因此每个时间单位(1ms)可以发送 5 帧数据.*SendQueue* 自动机还负责记录是否有“请求敏感器的遥测数据”类型的数据,发送此类数据时,向所请求的敏感器发送同步消息,触发敏感器在一定时延后传回遥测数据.

月球车控制软件的物理环境包括敏感器和地面遥控指令发送者.图 6 所示的自动机模拟敏感器与任务的交互过程:在接到“请求敏感器遥测数据”消息后,敏感器在一定延时后返回遥测数据.在月球车行走过程中,地面监控人员随时可能发出遥控指令,实施对月球车的控制.地面指令的到达使得任务 *TaskTC* 进入就绪状态.在上述 6 个任务中,*Task*1*TC* 的优先级最高,可能打断其他任务的执行,导致出现多个任务嵌套的复杂情况.

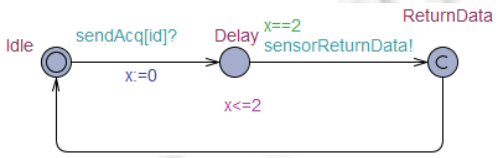


Fig.6 Timed automaton model for sensors

图 6 敏感器的时间自动机模型

图 7 所示的自动机模拟地面随机发送指令的场景,假设其频率为平均每秒发送 1 次指令.

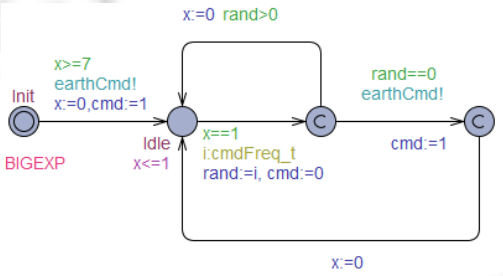


Fig.7 Timed automaton model for ground commander

图 7 地面指令发送者的时间自动机模型

3.3 正确性验证

月球车控制软件中的周期性任务 *Task*5*DataAcq* 通过一个同步过程向敏感器(以 IMU 为例)请求遥测数据,如图 8 中的序列图所示:*Task*5*DataAcq* 将 1 帧“向 IMU 请求数据”指令写入消息队列 *SendQueue*(事件 1),之后延时 4ms,期望在延时结束时收到 IMU 返回的全部 6 帧遥测数据.在这 4ms 中,控制软件和敏感器 IMU 应协同完成以下操作:数据到达 *SendQueue*,触发任务 *Task*3*Send*(事件 2),*Task*3*Send* 将 *SendQueue* 中数据发送到 CAN 总线(事件 3),IMU 从 CAN 总线上接收到数据请求指令(事件 4),0.5ms 后返回 6 帧遥测数据(事件 5),最后,*Task*5*DataAcq* 接收到遥测数据(事件 6).在正常情况下,这一过程耗时 0.192(事件 3)+0.5(IMU 响应时间)+0.192×6(事件 5)=1.844ms.因此,*Task*5*DataAcq* 在 4ms 的等待时间内应能接收到 IMU 返回的全部数据.发生遥测超时错误可能是由于其他任务干扰上述“请求-接收”同步过程,导致此过程中某个事件消耗太多时间.例如,*SendQueue* 中可能不仅有 *Task*5*DataAcq* 写入的数据,还有其他没有被及时发送的数据.由于 *SendQueue* 是一个先进先出的消息队列,在 *Task*3*Send* 发送“请求 IMU 数据”指令之前,需要先把 *SendQueue* 中已经累积的数据全部发送出去,使得 *Task*3*Send* 发送数据这一事件消耗太多时间.但是,通过测试和调试难以跟踪此场景中涉及的操作系统和任务的细粒度事件(如调度、抢占、优先级继承等),无法确定上述猜测是否错误发生的真正原因.

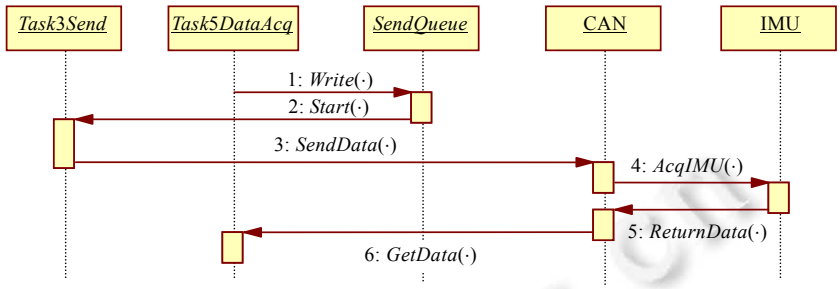


Fig.8 Request-Receive scenario

图 8 “请求-接收”过程

通过模拟执行模型我们发现,如果在 T 时刻系统满足以下条件,则出现遥测超时错误:

(1) *Task*6*GNC* 在 T 时刻就绪,*Task*2*Cru* 和 *Task*5*DataAcq* 在 8ms 后就绪;

(2) T+7ms 后,有地面命令.

由于地面遥控指令可能在任意时刻发送,上述条件(2)在系统的实际运行过程中可能出现.月球车控制软件的开发人员认为:由于地面指令可能触发科学数据下载等其他任务,加上时钟漂移等因素导致周期性任务的就绪时间与预期不同,因此实际系统可能满足条件(1).在此情况下,由于 *Task*1*TC* 和 *Task*2*Cru* 的优先级高于 *Task*5*DataAcq*,*Task*1*TC* 和 *Task*2*Cru* 分别向 *SendQueue* 写入 4 帧和 8 帧数据,之后,由于优先级继承,*Task*5*DataAcq* 的优先级被提升,*Task*5*DataAcq* 被调度并向 *SendQueue* 写 1 帧数据,*Task*5*DataAcq* 执行延时操作时 *Task*3*Send* 才被调度.此时,*SendQueue* 中已经累积了 13 帧数据,需要花费(13×0.192)ms 才能把这些数据都发送出去.因此,*Task*5*DataAcq* 需要等待 13×0.192+0.5+6×0.192=4.148ms 才能获得所有由 IMU 传回的遥测数据.*Task*5*DataAcq* 在延时的 4ms 内无法获得全部遥测数据,发生了超时错误.由此可见,遥测超时错误的产生与操作系统的调度机制、任务的优先级设置、共享资源的互斥访问、随机发生的外部事件等因素都有关,系统在特定条件下可能出现此错误.

通过进一步分析,开发人员认为:产生此错误的根源是任务优先级设置不合理,*Task*3*Send* 的优先级不够高,导致 *Task*3*Send* 没有及时将 *SendQueue* 中的数据发送出去.为此,开发人员调整了任务的优先级,将 *Task*3*Send* 的优先级由 4 升到 5,*Task*4*Receive* 的优先级由 3 升到 4,*Task*2*Cur* 的优先级由 5 降到 3.我们利用 UPPAAL 的统计模型检验分别验证了修改优先级之前和之后的模型,得到的结果见表 3.对于原始模型,*Task*5*DataAcq* 等待 IMU 返回遥测数据的最大时间为 5ms,说明遥测超时错误可能发生;*SendQueue* 的最大长度接近 13,说明导致超时错误的可能原因是 *SendQueue* 中累积了较多数据.对于调整优先级后的模型,*Task*5*DataAcq* 等待 IMU 返回遥测数据的最大时间为 3ms,*SendQueue* 的最大长度接近 8,这说明调整优先级后不会发生超时错误.这是由于 *Task*3*Send* 的优先级仅次于 *Task*1*TC*,每当 *SendQueue* 中有数据时,都会立刻调度 *Task*3*Send*,将 *SendQueue* 中的数据发送出去.此案例分析说明:我们的验证方法能够有效地分析 CPS 控制软件中由复杂原因引起的错误,有助于保障系统的正确性.

Table 3 Verification results on the original and revised models

表 3 对原始和修改后模型的验证结果

Query	含义	原始模型	修改后模型
E[≤100;2000] (max: *waitIMUtime*)	*Task*5*DataAcq* 等待 IMU 遥测数据的最大时间	4.907	3
E[≤100;2000] (max: *sendQLength*)	*SendQueue* 的最大长度	12.790 5	7.977 5

4 结束语

近年来,研究人员提出了多种对 CPS 系统进行形式化建模和验证的方法,这些方法关注不同领域 CPS 的不同特性.例如,Platzer 等人提出的微分动态逻辑(differential dynamic logic)着重刻画 CPS 的连续动态[6,7];Gupta 等人提出的时空混成自动机(spatio-temporal hybrid automata)适用于描述医疗领域 CPS 多节点并发操作产生的聚集效应对人体的影响[8];Johnson 等人针对由多个实体组成的分布式 CPS,提出了一种基于混成自动机的参数化建模方法,适用于分析分布式空中交通控制系统[9].与这些方法相比,我们的方法利用时间自动机刻画 CPS 的控制软件,着重体现 CPS 行为的并发性以及对环境的实时响应,有助于实现非确定物理环境下对 CPS 系统的精确控制.

CPS 系统多为实时系统,且常用于安全攸关领域,其控制软件需要严格遵守两个方面的时间约束:一是可调度性,即,任务集合的所有可能的执行序列都能在 deadline 之前完成[10];二是与时间相关的功能正确性,即,有时间要求的操作都能在预期时间内产生预期的效果.利用形式化方法分析实时系统是一个新兴的研究领域.在实时多任务系统的形式化建模与分析方面,一些研究者致力于设计适用于描述实时多任务系统的形式化语言,如扩展了时间自动机的任务自动机[11]和中断时间自动机[12]等.也有研究者设计了多任务系统的形式化模型库,如 SPIN 提供了由周期性任务组成的实时系统模型库[13],Herschel 模型也可被看作是一个描述多任务系统的建模框架[5].这些模型库刻画了具有相似结构特性的系统,提升了对系统建模的抽象层次.但是,现有的研究都是对实时多任务系统的可调度性分析,尚未见到验证此类系统功能正确性的研究.

本文提出的方法可以形式地分析 CPS 控制软件的正确性.利用此方法构造的模型是模块化的,具有通用性和可扩展性,可作为模型库,用于描述和分析不同类型的多任务系统.统计模型检验避免了状态爆炸问题,适用于分析规模大、复杂度高的实际 CPS 系统.与 Herschel 模型相比,我们构造的模型更真实、细致地模拟了多任务系统的行为,能够刻画与时间有关的功能属性.案例分析展示了此方法的有效性,特别是能够定位由复杂原因引起的错误.

References:

[1] Lee EA. Cyber physical systems: Design challenges. In: Proc. of the 11th IEEE Int'l Symp. on Object Oriented Real-Time Distributed Computing (ISORC 2008). Orlando: IEEE, 2008. 363−369. [doi: 10.1109/ISORC.2008.25]

[2] Glück PR, Holzmann GJ. Using SPIN model checking for flight software verification. In: Proc. of the Aerospace Conf. IEEE, 2002. 105−113. [doi: 10.1109/AERO.2002.1036832]

[3] Clarke EM, Zuliani P. Statistical model checking for cyber-physical systems. In: Proc. of the Automated Technology for Verification and Analysis (ATVA 2011). LNCS 6996, Springer-Verlag, 2011. 1−12. [doi: 10.1007/978-3-642-24372-1_1]

[4] Younes HLS. Ymer: A statistical model checker. In: Proc. of the 17th Int'l Conf. on Computer Aided Verification. LNCS 3576, Springer-Verlag, 2005. [doi: 10.1007/11513988_43]

[5] David A, Larsen KG, Legay A, Mikučionis M. Schedulability of Herschel-Planck revisited using statistical model checking. In: Proc. of the 5th Int'l Symp. on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA 2012). LNCS 7610, Springer-Verlag, 2012. 293−307. [doi: 10.1007/978-3-642-34032-1_28]

[6] Jeannin JB, Platzer A. dTL2: Differential temporal dynamic logic with nested temporalities for hybrid systems. In: Proc. of the 7th Int'l Joint Conf. on Automated Reasoning (IJCAR 2014). LNCS 8562, Springer-Verlag, 2014. 292−306. [doi: 10.1007/978-3-319-08587-6_22]

[7] Platzer A. Logics of dynamical systems. In: Proc. of the 27th Annual ACM/IEEE Symp. on Logic in Computer Science (LICS 2012). IEEE, 2012. 541−550. [doi: 10.1109/LICS.2012.13]

[8] Banerjee A, Gupta SKS. Spatio-Temporal hybrid automata for safe cyber-physical systems: A medical case study. In: Proc. of ACM/IEEE the 4th Int'l Conf. on Cyber-Physical Systems (ICCPS 2013). ACM Press, 2013. 71−80. [doi: 10.1109/ICCPS.2013.6604001]

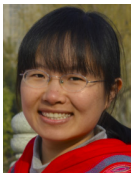
[9] Johnson TT, Mitra S. Parametrized verification of distributed cyber-physical systems: An aircraft landing protocol case study. In: Proc. of the IEEE/ACM 3rd Int'l Conf. on Cyber-Physical Systems (ICCPS 2012). IEEE Computer Society, 2012. 161−170. [doi: 10.1109/ICCPS.2012.24]

[10] Davis RI. A review of fixed priority and EDF scheduling for hard real-time uniprocessor systems. ACM SIGBED Review, 2014,11(1):8−19. [doi: 10.1145/2597457.2597458]

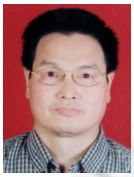
[11] Fersman E, Krcál P, Pettersson P, Yi W. Task automata: Schedulability, decidability and undecidability. Information and Computation, 2007,205(8):1149−1172. [doi: 10.1016/j.ic.2007.01.009]

[12] Bérard B, Haddad S, Sassolas M. Interrupt timed automata: Verification and expressiveness. Formal Methods in System Design, 2014,40(1):41−87. [doi: 10.1007/s10703-011-0140-2]

[13] Florian M, Gamble E, Holzmann G. Logic model checking of time-periodic real-time systems. In: Proc. of the Infotech@Aerospace 2012. 2012. 2455−2462.



单黎君(1979－),女,河南开封人,博士,讲师,主要研究领域为模型驱动开发方法,形式化方法.



周兴社(1955－),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为嵌入式计算,分布计算.

王宇英(1978－),女,博士生,讲师,CCF 会员,主要研究领域为嵌入式系统仿真验证技术.

赵雷(1981－),男,博士,工程师,主要研究领域为软件测试及验证技术.

万丽景(1984－),男,工程师,主要研究领域为嵌入式软件,导航制导与控制.

乔磊(1982－),男,博士,高级工程师,主要研究领域为操作系统设计及验证技术.

陈建新(1969－),男,博士,研究员,主要研究领域为空间机器人,航天器控制.