形式化方法和信号解释 Petri 网在 PLC 编程中的应用①

王芳芳[1], 雷建和[1], 张 丹[2], 聂余满[3], 高 志[1]

[1](青岛理工大学 自动化工程学院, 青岛 266033)
[2](安大略理工大学 工程及应用科学学院, 奥沙瓦 L1H7K4)
[3](中科院合肥智能机械研究所, 合肥 230031)

摘 要: 针对传统的 PLC 编程方式在解决复杂控制问题时存在的缺陷, 采用一种将形式化和信号解释 Petri 网(SIPN)应用于PLC程序设计的方法. 通过一个机器人焊接单元的例子来说明这一设计过程, 首先建立系统控制算法的信号解释 Petri 网模型, 验证其是否满足基本 Petri 网的安全性、活性和可逆性的特征, 然后利用模型检测工具 Cadence SMV 对系统模型进行验证和确认(V&V), 检验其是否满足 SIPN 的确定性、终止性和输出正确性, 从而避免了控制算法的设计过程中可能出现的并发、冲突和死锁等事件, 由此设计出具有更高的正确性和可靠度的 PLC 程序.

关键词: 形式化方法; 信号解释 Petri 网; 模型检测; 机器人焊接单元

Application of Formal Metrods and Signal Interpreted Petri Net to PLC Programming

WANG Fang-Fang[1], LEI Jian-He[1], ZHANG Dan[2], NIE Yu-Man[3], GAO Zhi[1]

[1](School of Automation, Qingdao Technological University, Qingdao 266033, China)
[2](School of Engireering and Applied Sciences, University of Ontario Institute Technology, Oshawa L1H7K4, Canada)
[3](Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei 230031, China)

Abstract: To overcome the defects in traditional PLC programming, an approach combined formal methods with Signal Interpreted Petri Net (SIPN) is presented. An example of robot welding unit is used to illustrate this process. This paper builds a model of the control algorithm with Signal Interpreted Petri Net first, and verifies whether it satisfies the safety, liveness and reversibility characteristics of basic Petri net. Then it uses the model checking tool Cadence SMV for model verification and validation, to test whether it meets the properties of certainty, termination and output correctness. Thus the possible events like concurrency, conflict and deadlock in control algorithm designing process can be avoided and correct and dependable PLC programs are designed.

Key words: formal methods; Signal Interpreted Petri Net; model checking; robot welding unit

可编程控制器(PLC)功能强大、可靠性高、使用方便, 因此被广泛应用于工业制造和生产过程中. 基于传统的IEC61131-3标准中定义的逻辑控制设计语言如梯形图(LD)、顺序功能图(SFC)、功能块图(FB)等图形化语言, 虽然程序指令形象直观, 但指令间的逻辑关系抽象, 程序调试的过程较为复杂和繁琐. 近年来, 随着控制系统复杂程度的不断提高和用户对安全性和功能需求的不断增加, 传统的 PLC 程序设计方法已不能满足要求, 主要有如下几个弊端:

(1) 一般说来, 传统的 PLC 程序设计是一个开环的过程, 从程序的编程设计到调试运行工作量大, 整个开发工作周期长、成本高.

(2) 当程序的运行出现错误时, 设计者只能按照程序设计的顺序由上至下逐条检查, 对程序检测的过程不能进行反馈, 不能很快指出问题具体出现在哪一部分.

(3) PLC 编程软件只能检测语法、语义上的错误, 并不能检测到逻辑上的问题, 不能很好地解决系统中

① 基金项目:山东泰山学者建设工程基金(C2010-T005);国家自然科学基金(61201400)
收稿时间:2014-01-04;收到修改稿时间:2014-03-17

存在的并发、冲突和死锁等事件, 因此, 对其性能和可靠性难以评价.

为了解决以上问题, 本文介绍一种将形式化方法(Formal Methods)和信号解释 Petri 网(SIPN)相结合应用在 PLC 程序设计当中的方法. 形式化方法是一种用于规范、设计和验证计算机系统的基本数学方法. 引入形式化方法的目的是使系统程序在真正实施之前保证得到的是一个正确的控制算法, 使设计者能更早地发现问题、解决问题, 使系统具有较高的正确性和可靠度, 并具有良好的结构, 便于系统的调试和维护, 能更好地满足客户需求. Petri 网在对控制算法的建模上展现了很好的特性, 能够清晰地表达控制算法的因果关系和并发性, 但是因为不能很好地处理与外界联系较多的系统建模, 因此引入信号解释 Petri 网(Signal Interpreted Petri Net, SIPN). SIPN 是对基本 Petri 网结构的扩展, 它除了包含基本 Petri 网的图形解释和数学处理等功能, 还能够对输入和输出信号进行详细的处理, 突出显示了外界信号对系统的影响[1,2].

1 信号解释Petri网(SIPN)简介[3-5]

SIPN 在普通 Petri 网的结构基础上加入了输入/输出元素, 一个 SIPN 由一个九元组来描述:

$SIPN=(P,T,F,M_0,I,O,\varphi,\omega,\Omega)$, 其中$(P,T,F,M_0)$为一个普通的 Petri 网, P 表示位置, T 表示变迁, F 表示弧, M_0 表示一个二进制的初始标识. I 为一组输入信号, O 则表示一组输出信号. φ 为每个变迁与发生条件间关联的映射, ω 为每个位置与输出相关的映射, Ω 为网系统的输出函数, 是所有位置输出的组合.

Petri 网由两种基本类型的节点构成, 即位置和变迁, 它们之间通过有向弧连接. 位置通常用来描述系统的资源或状态, 变迁描述系统状态改变的条件或事件, 系统的动态变化则通过托肯(Token)在网中的流动(标识的变化)表现出来, 这一过程通过变迁的使能来实现, 变迁的使能过程有以下五条规则:

(1)一个变迁如果它的前面位置都被标识, 后面位置都没被标识时, 变迁被使能.

(2)如果一个使能变迁能立即发生, 那么它的使能条件必须满足.

(3)所有能够发生的变迁都不能与其它变迁同时发生(因为冲突在 SIPN 中被认为是设计错误, 没有能够解决冲突的办法).

(4)使能的过程会一直重复, 直到达到一个固定的标识状态(即在当前的输入信号中, 没有变迁能够发生).

(5)当到达一个稳定的标识后, 可以通过被标记位置的输出函数来分析和计算这些输出信号.

基于以上理论, 变迁使能的发生即可通过给定的状态进行判断, 同时也可得到变迁发生后的标识状态.

SIPN 具有与普通 Petri 网相似的性质, 如安全性、活性和可逆性. 安全性是指 SIPN 中位置的 Token 值不能超过 1. 根据 Petri 网的变迁规则, 只要初始状态中各位置 Token 值不超过 1, 当所有位置是安全的, 则整个 SIPN 安全; 活性用于检测网中是否存在死锁, 意味着变迁永远可以再次使能, 如果不管标识如何变化, 网中都不存在不可激发的变迁, 则该 Petri 网是活的; 可逆性表示初始标识能再次实现, 确保其 SIPN 描述的控制过程能回到初始状态.

此外, SIPN 自身还具有一些特殊的性质. 确定性, 即防止控制过程出现冲突, 控制算法的设计应能根据实际情况决定控制器的动作, 在两者都有发生权的情况下, 同一时刻只能有一个发生; 终止性, 在控制算法中应至少存在一个稳定状态, 避免出现 “死循环”; 输入依赖性和输出正确性, 即每一个输入信号应该对系统产生作用, 输出信号必须为 “0”或 “1”的正确的形式.

2 用形式化方法进行系统设计的过程

控制系统的设计过程如图 1 所示.

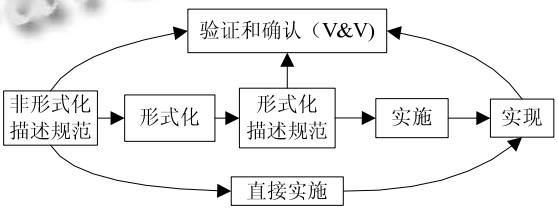

图 1 控制系统的设计过程

在没有使用形式化方法进行设计时, 控制系统的设计过程仅由最外环下部分组成, 即控制过程是从一种非形式化的描述规范直接得来.

多数情况下, 系统的设计是从一个给定的关于控制系统的非形式化描述规范开始的. 这个非形式化的描述规范是对控制系统的过程和要求的描述, 采用一些较易于理解的描述形式包括时序图、方程、草图和

仪表图等. 然而, 这种描述规范的不同部分之间并不能很好地区别开来, 所以不利于系统的完整性、明确性和一致性的测试[6].

所以, 采用形式化方法进行控制系统的设计是较为可靠的一种方法. 首先, 是将这些系统的非形式化描述规范进行形式化, 即将非形式化的描述规范转化成形式化描述规范的过程. 它包括以下几项工作:

对特定属性的形式化, 这样就产生了一系列可满足 PLC 系统和控制过程的属性; 对不可控过程的形式化建模, 从而可以得到那些基于模型的方法中所需要的的过程模型; 如果通过非形式化描述文本给定的控制问题非常清楚, 那么可以对控制算法直接进行形式化建模; 实施的过程, 就是依赖形式化描述文本的实现, 得到目标系统的过程; 设计过程的实现, 通常包括硬件和软件的实现, 若已存在一个具有明确定义了功能的标准的硬件, 那么实现实际就是控制算法程序的实现.

验证和确认(Verification & Validation, V&V), 是在 PLC 编程部分使用的形式化方法. 验证(Verification)意味着形式化方法的使用是为了证明控制算法满足给定的规范, 同时产生关于控制算法正确性的重要信息, 保证在将系统的描述规范或流程图转换成可执行的计算机程序时具有足够的精确度, 也就是将模型做正确. 在确认(Validation)部分, 控制算法的特定功能属性必须都被形式化, 它是从形式化描述文本和实现部分得到的输入信息, 它显示了控制过程是否是按照它应该的行为进行动作, 即确保模型的可靠性, 保证得到的是正确的模型[7,8].

本文中, 我们使用符号模型检验方法来表现验证和确认的过程. 在这种方法中, 系统作为一个有限状态转移系统被建模, 并且用时序逻辑来描述待验证的规范. 然后, 用一个验证程序对这个系统的状态空间进行穷尽搜索, 来判断这些规范是否都为 “真”. 当验证结果出现否定时, 给出一个反例使设计者能够发现并修改存在的问题, 当被检测的规范都能得到 “真”的结果时, 表明期望的特性都能满足, 从而保证设计出正确、可靠的 PLC 程序. 这一过程通过使用 Cadence SMV 软件来完成[9,10].

3 设计过程举例

本文中我们用上面说明的方法来讲述设计系统 PLC 程序的过程. 系统为一个机器人焊接单元, 布局如图 2 所示.

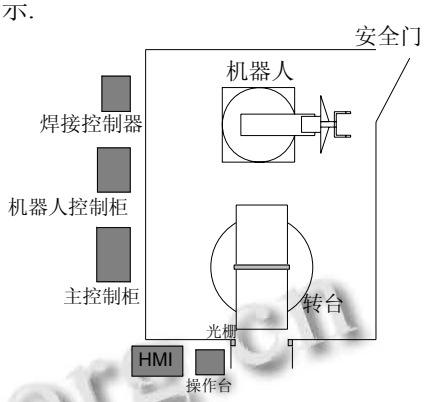

图 2 机器人焊接单元系统工位图

系统的运行过程如下:

机器人位于原点做好焊接准备后, 由工人放入要焊接的板件. 当系统收到板件放置到位信号后, 按下系统的启动按钮, 开始焊接, 此时夹具关闭, 当系统收到夹具夹紧到位的信号后, 转台的定位销下降, 转台开始正转(顺时针). 转台正转到位以后, 定位销上升固定转台. 然后, 系统发出启动焊接指令, 机器人开始焊接. 一段时间后, 焊接完毕, 机器人复位. 当机器人回到原点且静止后, 发出焊接完成信号, 定位销下降, 转台开始反转(逆时针), 反转到位以后, 定位销上升, 夹具松开. 最后, 当夹具松开到位后, 由工人卸下焊接完毕后的焊件, 再放入新的焊件, 开始新一轮的焊接工作.

在该系统中, 某些动作的执行必须存在几个前提条件. 例如, 夹具关闭的前提是板件必须放置到位, 在转台的焊接一侧安置有两个电感式接近开关, 若板件放置无误, 则接近开关点亮, 方可执行夹具的夹紧动作, 以防止将板件夹坏. 而夹具打开则需要转台必须反转到位, 定位销上升以后, 保证转台固定才可松开夹具. 转台转动必须在定位销下降以后, 而定位销下降需要机器人位于原点的位置, 不可在焊接过程中出现转台转动的情况.

为了确保系统的安全性, 系统还设有隔离栅栏和光栅等安全设施. 隔离栅栏的作用是将机器人的工作区域和外界隔绝开来, 机器人在焊接过程中, 若有人试图打开安全门进入机器人焊接区域, 必须及时停止焊接工作. 同样, 在转台转动过程中, 不允许人员进入机器人工作区域, 所以, 在装件区域两侧设置了光

栅, 一侧是发射端, 一侧是接收端, 正常情况下, 接收端能一直接收到发射端发出的光, 但当有人试图从装件区进入机器人工作区域时, 会挡住发射端发出来的光, 这样接收端就接收不到信号了, 此时转台必须及时停止转动[11-13].

PLC 的输入/输出信号如表格 1 所示.

表 1 PLC 的输入/输出信号

类型	名称	含义
输入信号	I1	系统的启动按钮
	I2	机器人在原点
	I3	板件放置到位
	I4	夹具夹紧到位
	I5	夹具松开到位
	I6	转台正转到位
	I7	转台反转到位
	I8	定位销上升到位
	I9	定位销下降到位
	I10	机器人焊接完毕
	I11	光栅信号
	I12	安全门信号
	I13	系统故障排除
输出信号	O1	定位销上升
	O2	定位销下降
	O3	夹具松开
	O4	夹具夹紧
	O5	转台正转
	O6	转台反转
	O7	启动机器人焊接
	O8	转台停止转动
	O9	机器人停止焊接

根据系统的工作过程, 可以画出如图 3 所示的机器人焊接系统的信号解释 Petri 网模型.

在系统的初始状态, 只有 P1 被标识, 系统处于待命状态, 等待焊接工作. 由工人在转台的 A 面将需要焊接的板件放置到位后, 接近开关的指示灯点亮, PLC 收到板件放置到位信号(I3=1), 按下启动按钮(I1=1), 且此时机器人在原点做好焊接准备(I2=1)后, T1 发生, Token 从 P1 移到 P2, 夹具执行夹紧动作(O4=1), 当夹具加紧到位(I4=1), Token 移到 P3, 转台上的定位销下降(O2=1), 下降到底(I9=1)后, P4 发生, 转台正转(O5=1), 即顺时针转动. 此时, 系统可能发生两种状态. 在正常情况下, 转台会正转到位(I6=1), T4 发生, Token 从 P4 移到 P5, 定位销上升(O1=1), 旋转过程顺利完成. 若在转台旋转过程中, 有人试图从装件区进入机器人焊接区域, 则会遮挡住光栅信号, 使得光栅信号输出为 0(I11=0), T11 发生, Token 从 P4 移到 P11, 然后, 转台应立即停止转动(O8=1). 当故障排除以后(I11=1 且 I13=1), 转台继续转动(O5=1), 直到正转到位(I6=1), 定位销上升(O1=1). 定位销上升到位(I8=1)后, 启动机器人焊接(O7=1).

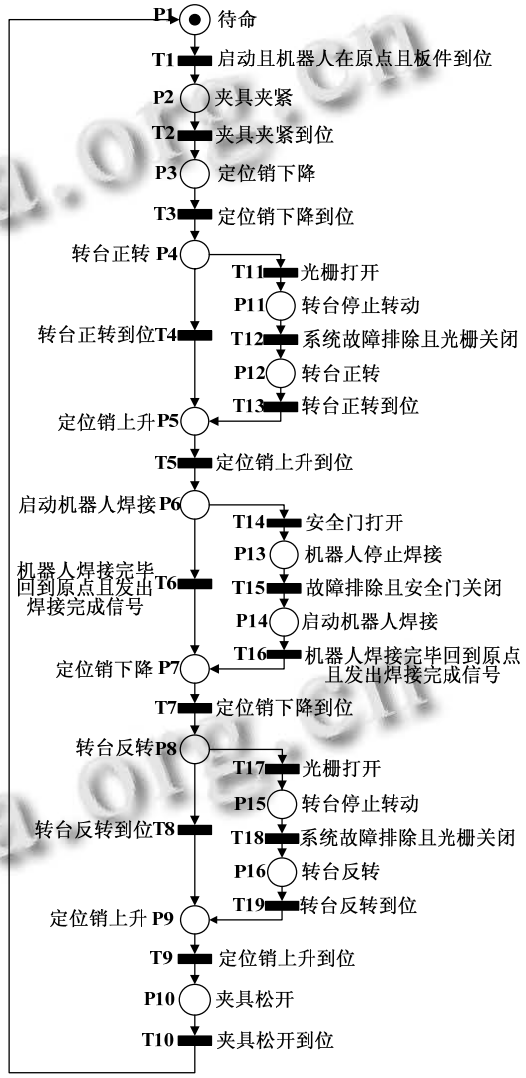

图 3 机器人焊接系统的信号解释 Petri 网模型

在机器人焊接过程中, 系统可能出现两种情况. 正常情况下, 机器人顺利焊接完成并退回到原点, 向发出发出焊接完成信号, 即 T6 发生(I2=1 且 I10=1), Token 从 P6 移到 P7. 但是, 在机器人焊接过程中, 当有人试图打开安全门进入机器人焊接区域时(I12=0), T14 发生, 机器人应立即停止焊接(O9=1), 在故障排

除以后(I12=1 且I13=1), 机器人继续焊接(O7=1), 直到焊接完毕回到原点发出焊接完成信号(I2=1 且 I10=1).

焊接完成后, 定位销下降(O2=1), 转台反转(O6=1). 反转过程与正转过程类似, 也会发生光栅有无信号两种情况, Token 从 P8 移到 P9. 反转到位(I7=1)后, 定位销上升(O1=1)固定转台, 然后夹具松开(O3=1), 当夹具松开到位(I5=1)后, 由工人取下焊接完成的焊件, 并放入新的待焊接的板件, 准备新一轮的焊接工作, 系统的 Token 值回到 P1.

4 模型检测过程

在本文的第二节中, 我们介绍了SIPN具有的与普通Petri网相似的一般性质和自身具有的一些特殊的性质, 控制算法的设计应该能够满足这些特征. 在系统的初始状态, 该 SIPN 各位置的 Token 值均不超过 1, 且 Token 在网中转移过程中, 每个变迁能被再次触发, 能回到初始状态, 因此, 满足基本 Petri 网的安全性、活性、可逆性的特征, 而该SIPN 的特殊性质则有待验证. 本文中我们使用 Cadence SMV 软件进行该模型的检测工作, SMV 基于 “符号模型检测”(Symbolic Model Verifier), 是一种分析有限状态系统的常用工具, 使用时序逻辑来描述模型的所有功能特性[14-16].

为了验证该SIPN的控制算法是确定性的, 我们需要防止所有潜在的冲突. 由该SIPN模型我们可以看出, T4 与 T11, T4 与 T13, T6 与 T14, T6 与 T16, T8 与 T17, T8 与 T19 是可能出现的 6 对冲突, 用时序逻辑可以表达为:

P1a: SPEC AG ~ (T4&T11) ;

P1b: SPEC AG ~ (T4&T13) ;

P1c: SPEC AG ~ (T6&T14) ;

P1d: SPEC AG ~ (T6&T16) ;

P1e: SPEC AG ~ (T8&T17) ;

P1f: SPEC AG ~ (T8&T19) ;

为了保证系统的控制过程具有终止性, 即避免出现无限循环, 我们创建一个变量 eoc (End of Cycle)来声明[17], 表达式为:

eoc:= ~ (T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 | T11 | T12 | T13 | T14 | T15 | T16 | T17) ;

P2: SPEC AG EF eoc ;

在设计中, 还应保证输出形式的正确性, 即输出为 “0”或 “1”. 例如, 夹具松开输出信号(O3), 用时序逻辑形式写出表达式为:

P3: SPEC AG EF ((eoc&(O3=0))|(eoc&(O3=1))) ;

此外, 该系统在实际运行中还应满足一些特殊的功能特性, 有以下几个方面:

定位销的上升和下降, 夹具的夹紧和松开, 转台的正转和反转不可以同时进行. 写成时序逻辑的形式为:

P4a: SPEC AG ~ ((O1=1)&(O2=1)) ;

P4b: SPEC AG ~ ((O3=1)&(O4=1)) ;

P4c: SPEC AG ~ ((O5=1)&(O6=1)) ;

定位销下降过程中转台不能转动, 同样, 转台的转动过程中定位销也不可上升.

P5a: SPEC AG ~ ((O1=1)&(O5=1)) ;

P5b: SPEC AG ~ ((O1=1)&(O6=1)) ;

P5c: SPEC AG ~ ((O2=1)&(O5=1)) ;

P5d: SPEC AG ~ ((O2=1)&(O6=1)) ;

为防止将板件挤坏, 板件放置到位前不能执行夹具的夹紧动作, 同时, 在定位销未上升将转台固定之前, 也不能将夹具松开. 写成时序逻辑的形式为:

P6a: SPEC AG (~I3 → EF ~ (O4=1)) ;

P6b: SPEC AG (~I8 → EF ~ (O3=1)) ;

转台转动的前提是定位销必须是下降的, 而定位销下降的前提是机器人必须位于原点. 用时序逻辑表示为:

P7a: SPEC AG (~I2 → EF ~ (O2=1)) ;

P7b: SPEC AG (~I9 → EF ~ (O5=1)) ;

P7c: SPEC AG (~I9 → EF ~ (O6=1)) ;

启动机器人焊接前, 夹具必须夹紧, 定位销必须处于上升的位置. 用时序逻辑表示为:

P8a: SPEC AG (~I4 → EF ~ (O7=1)) ;

P8b: SPEC AG (~I8 → EF ~ (O7=1)) ;

通过使用SMV软件对模型进行检测, 遍历系统所有可能的状态, 最终可得到 “真”的结果, 从而排除了系统中可能存在的死锁和冲突等事件. 由此表明图 3 的SIPN模型不仅能满足其自身的特殊性质, 同时满足在实际应用中所要求的特性, 所以, 根据此 SIPN 模型设计的 PLC 程序在实际系统中正确、可行.

5 结论

本文介绍了一种利用形式化方法来设计正确、可靠的 PLC 程序的过程. 基于 SIPN 构建一个机器人焊

接单元的例子, 首先是对系统的运行过程进行透彻的分析, 建立初步的形式化模型, 然后, 通过分析 Token 在SIPN中的流动来分析系统的动态行为, 检查系统是否满足基本 Petri 网的安全性、活性和可逆性等基本特征. 除此之外, 利用时序逻辑来表示 SIPN 自身的一些特殊的性质, 如确定性、终止性和输出正确性, 并利用模型检测软件进行验证和确认, 由此来设计系统的 PLC 程序.

基于SIPN建立的系统模型, 以图形方式描述系统的控制算法, 给出一种可视化的控制算法流程的反馈, 使得编程易于更加快速地实现. 同时通过形式化方法的运用, 使用模型检测工具 Cadence SMV 遍历系统的所有状态, 有效避免了系统中可能存在的冲突、死锁等事件, 所以, 由此方法设计出的 PLC 程序较传统方法比起来具有更高的正确性和可靠度.

参考文献

1 齐鹏飞,罗继亮,陈雪琨.PLC 程序形式化的设计与验证.华侨大学学报(自然科学版),2013,34(3):241–244.

2 李俊,戴先中,孟正大.基于信号解释 Petri 网的可重构逻辑控制器分析与设计.东南大学学报(自然科学版),2004, 34(11):101–107.

3 Klein S, Frey G, Litz L. Designing fault-tolerant controllers using SIPN and Model-checking. Fault Detection, Supervision and Safety of Technical Processes2003: A Proc. Volume from the 5th IFAC Symposium. Washington DC. 2003.

4 Minas M, Frey G. Visual PLC-programming using signal interpreted Petri nets. American Control Conference. 2002.

5 高晓锋,钟艳如,黄美发,赵新有.基于 SIPN 的一种控制系统 PLC 程序生成研究.现代制造工程,2005,(12):38–40.

6 Frey G, Litz L. Formal methods in PLC programming. Systems, Man, and Cybernetics. 2000 IEEE International Conference, Nashville, 2000.

7 王瑞利,林忠,袁国兴.科学计算程序的验证和确认.北京理工大学学报,2010,30(3):353–357.

8 John CH. Systems engineering verification and validation. Aerospace America, 2012, 50(10): 62.

9 王常春,董威.在模型检测工具 SMV 中实现进程阻塞.计算机工程与科学,2006,28(3):85–87.

10 Alenljung T, Lennartson B, Hosseono MN. Sensor graphs for discrete event modeling applied to formal verification of PLCs. IEEE Trans. on Control Systems Technology, 2012, 20(6): 1506–1521.

11 胡敏,洪涛,王勇,等.基于总线技术的柔性机器人焊接工作站.电焊机,2010,40(5):95–98.

12 陈国辉,胡国雨,陶渊亮,等.机器人点焊在汽车座椅骨架焊接的应用.汽车零部件,2012,(11):95–97.

13 李芳,杨海澜,华学明,吴毅雄,朱麟.汽车纵梁点焊机器人工作站设计.电焊机,2011,41(7):7–9.

14 Vakili A, Day NA. Using model checking to analyze static properties of declarative models. 2011 26th IEEE/ACM International Conference on Automated Software Engineering. 2011.

15 Clarke EM, Heinle W. Modular translation of statecharts to SMV. Carnegie-Mellon University of Computer Science. 2000.

16 Mertke T, Frey G. Formal verification of PLC-programs generated from signal interpreted Petri nets. Systems, Man, and Cybernetics, 2000 IEEE International Conference. Nashville. 2000.

17 Gergely EI, Husi G, Yildirim S. PLC programs design using signal interpreted petri networks. Journal of Computer Science and Control Systems, 2009, 2(1): 102–106.