

符号化离散控制器的物理实现技术

任 明 明¹, Emil Dumitrescu², Eric Niel², 李 鹏³, 李 蓉³

(1 南京南瑞集团公司 信息通信技术分公司, 江苏 南京 211100;

2 法国国家科学研究中心 安培实验室, 法国 维勒班 F-69100;

3 宁夏电力公司信息通信分公司, 宁夏 银川 750000)

摘 要: 针对基于二进决策树(BDD)技术综合出的符号化离散控制器,提出了一种物理实现技术.该方法通过针对可控变量引入环境变量,以及将监督器方程转换为控制函数向量,同时解决了符号化控制器综合技术带来的控制的非确定性和结构的不兼容性.还尝试将离散控制器技术引入到硬件设计自动化领域,利用离散控制器作系统错误的自动修正.介绍了该技术的理论原理与控制结构特性,并通过一个实际的片上串并转换系统,详细展示了该实现技术.

关键词: EDA;离散控制器综合;形式验证;硬件设计

中图分类号: TN402

文献标识码: A

文章编号: 1000-7180(2014)05-0112-05

Phisycal Implementation Technique for Symbolic Discrete Controller Synthesis

REN Ming-ming¹, Emil DUMITRESCU², Eric NIEL², LI Peng³, LI Rong³

(1 Information Telecommunication Technology Corporation, Nari Group Corporation, Nanjing 211100, China;

2 AMPERE Laboratory, CNRS, Villeurbanne 69100, France; 3 State Grid Ningxia Corporation, Yinchuan 750000, China)

Abstract: A physical hardware implementation technique has been proposed for discrete controllers synthesized by symbolic DCS. By introducing auxiliary variables for the controllable input variables and transforming the supervisor equation to a vector of control functions, this method resolves both the control non-determinism problem and the structural incompatibility problem introduced by the symbolic technique. We also try to introduce the DCS technique into the hardware design automation area, hoping to correct automatically design errors. This paper illustrates this technique as well as the control architecture by a real-life hardware design.

Key words: EDA; discrete controller synthesis; formal verification; hardware design

1 引言

离散控制器综合技术(DCS)[1]已经在EDA等领域得到了应用.由于符号化的离散控制器综合技术^[2-3]能更有效地处理超大状态空间,因此近年来得到了较为深入的研究和应用.本文的控制器实现技术即特别针对符号化综合技术生成的离散监督器.为了避免术语混淆,在本文中监督器(supervisor)特指由符号化综合技术生成的监督方程;而控制器(controller)特指监督器的具体实现.

监督器的实现问题涉及到多个实际问题,例如监督器实现后的效率、自治性、容错性、分布式、鲁棒性及紧凑性等要求.但是,在满足这些要求前,有两个问题必须解决.

(1)控制的非确定性:大多数综合的监督器都是非确定性的,即当同时存在多个控制解时,这些解都是可行的.此时,必须手动或某种方法自动做出选择,解决这种不确定性.离散控制器综合技术极少直接产生确定性监督器;

(2)与模块化设计模式在控制结构上的不兼容

性:该问题只存在于符号化综合技术上.符号化的监督器不具备输入/输出信号,它们只是由布尔方程编码的所有合法的控制解.一种直观的解决方法是在运行时在线解答监督器方程,这需要一个物理上运行在微处理器上的布尔计算软件.这种实现本质上是一种软件架构,在运行效率上有较大局限性.

我们的目标是解决控制的非确定性和结构上的不兼容性问题.解决控制的非确定性保证了一个自治的实现;解决架构的不兼容性本质上是寻找一种方法,将监督器方程表达为一组控制函数.该方法的优点,首先也是最重要的,可以很容易地得到硬件实现;其次,以控制函数向量表达的监督器实现,要远比监督器方程紧凑.另外,我们尝试把传统闭环控制架构引入硬件设计环节.我们提出了一类可以由控制架构解决的硬件设计问题.

2 定义

2.1 可控有限状态机

可控有限状态机 P 定义如下:

$$P = \langle I, S, \delta, s_0, O, \lambda \rangle$$

其中:

I 为布尔输入变量集,且 $I = U \cup C, U \cap C = \emptyset$;

$U = \{u_0, \dots, u_{r-1}\}, r > 0$ 为不可控输入变量集,记 $\mathbf{u} = (u_0, \dots, u_{r-1})$ 为 U 的一个取值;

$C = \{c_0, \dots, c_{p-1}\}, p > 0$ 为可控输入变量集,记 $\mathbf{c} = (c_0, \dots, c_{p-1})$ 为 C 的一个取值;

$S = \{s_0, \dots, s_{n-1}\}, n > 0$ 为状态变量集,记 $\mathbf{s} = (s_0, \dots, s_{n-1})$ 为 S 的一个取值;

$\delta: \mathbb{B}^{p+r} \times \mathbb{B}^n \rightarrow \mathbb{B}^n$ 为 P 的状态转移函数;

$s_0 \in \mathbb{B}^n$ 是 P 的初始状态;

$O = \{o_0, \dots, o_{m-1}\}, m > 0$ 为输出变量集;

$\lambda: \mathbb{B}^{p+r} \times \mathbb{B}^n \rightarrow \mathbb{B}^m$ 为与 O 关联的输出函数.

以 P 表示受控系统. P 的状态集合由特征函数表示.令 $E \subset \mathbb{B}^n$, 集合 E 的特征函数定义为

$$C_E: \mathbb{B}^n \rightarrow \mathbb{B} \text{ 且 } C_E(e) = 1 \Leftrightarrow e \in E.$$

集合运算符都有其对应的布尔运算符:“+”(逻辑或)执行并集运算;“ $\cdot$ ”(逻辑与)执行交集运算;逻辑反 $\overline{C_E}$ 代表集合 E 在 $\mathbb{B}^n$ 上的补集.

令 $S' = \{s'_0, \dots, s'_{n-1}\}$ 为后继状态集合. P 的状态转移关系 T 定义为所有合法状态转移的集合 s a s' 使得 $s' = \delta(s, \mathbf{u}, \mathbf{c})$:

$$T(s, \mathbf{u}, \mathbf{c}, s') = \prod_{i=0}^{n-1} s'_i \Leftrightarrow \delta_i(s, \mathbf{u}, \mathbf{c}).$$

2.2 符号化离散控制器综合

基于形式语言的离散控制器综合技术由 [1] 提出并发展.同时,在设计自动化领域,基于二叉决策树(BDD)的状态集合符号化表示方法显示了极大的成功,某种程度上解决了状态空间的指数膨胀问题[4].基于符号化方法的离散控制综合技术被提出并发展[2,3,5].

我们假设所研究的系统为完全可观.对于一个给定的受控系统 P 和一个控制目标 Q ,符号化综合技术计算出一个监督器 X ,使得 P 总是满足 Q .该控制架构如图 1 所示.

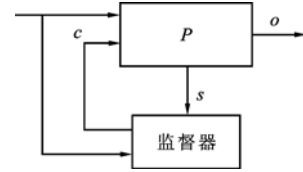


图 1 控制架构

计算过程主要由两步组成:

(1) 计算 P 的状态子集,控制不动点(IUC).只要 P 保持在 IUC 内部,则对 Q 的违反就会被无限期推迟.控制不动点的计算过程超出了本文的范围,具体细节可参考文献[2,5];

(2) 计算监督器 X :即所有指向 IUC 的状态转移 s a 集合.监督器的精确表达式如下:

$$X(s, \mathbf{u}, \mathbf{c}) = \exists s': T(s, \mathbf{u}, \mathbf{c}, s') \cdot \text{IUC}(s').$$

当且仅当 $s_0 \in \text{IUC}$ 时,存在控制解.从动态的观点来看,监督器 X 和系统外部环境,分别通过对系统可控输入 C 和不可控输入 U 的操控进行“博弈”.监督器的目标是系统永远不会进入违反 Q 的状态.因此,对于任意状态 $s \in \text{IUC}$ 和任意不可控输入向量 $\mathbf{u} \in \mathbb{B}^r$,监督器 X 给出可控输入 $\mathbf{c}$ 的合适的值,使得 P 的当前状态转移指向不动点 IUC.

符号化监督器 X 具有两个重要特点:(1) 最大容许/最小限制,即所有指向不动点 IUC 的状态转移都被包含在监督器 X 中,除此之外对系统行为无额外限制;(2) 控制的非确定性,即对于任意给定当前状态和给定不可控输入值向量,有可能在不动点 IUC 中存在多于一个的后继状态.我们的目标是实现监督器 X ,也即解决控制的非确定性问题,同时尽可能保持最大容许性.

3 符号化监督器实现技术

3.1 结构兼容性要求

监督器的实现的目标是解决控制的非确定性.

得到的确定的控制解称为控制器,通常是监督器的一个子集.

另外,符号化综合技术存在架构问题:监督器 X 是一个布尔特征函数,其编码了所有对于控制目标可接受的状态转移. X 的布尔表达式同 P 以及控制架构不兼容,如图 1. 这种不兼容性通常由“在线”或“离线”解答布尔方程解决:

$$X(s, u, c) = 1 \quad (1)$$

本文提出的实现技术同时解决了以上两个问题,如图 2 所示. 控制的非确定性通过为每个 c_i 引入相应的辅助环境变量 c_i^{env} 而被显化;结构的不兼容性通过计算每个可控变量 c_i 的函数表达式得以解决,相当于离线解答方程(1). 由监督器 X 构造由 p 个控制函数组成的控制器:

$$\begin{aligned} f_i: \mathbb{B}^n \times \mathbb{B}^{p+r} \rightarrow \mathbb{B}, i = 0 \cdots p-1, \text{使得} \\ c_i = f_i(s_0, \cdots, s_{n-1}, u_0, \cdots, u_{r-1}, c_0^{\text{env}}, \cdots, c_{p-1}^{\text{env}}) \end{aligned} \quad (2)$$

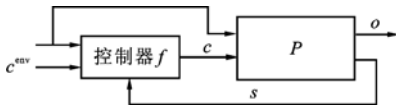


图 2 期望的控制架构

X 与 f 的关系由以下方程确定:

$$X(s, u, c) = \exists c_0^{\text{env}}, \cdots, c_{p-1}^{\text{env}} : \prod_{i=0}^{p-1} (c_i \Leftrightarrow f_i(s, u, c^{\text{env}})) \quad (3)$$

我们希望:(1) 任何被 X 所允许的控制解都能被 f 重现;(2) 任何由 f 计算出的控制解都被 X 所接受.

3.2 监督器实现算法

实现 X 的一个先决条件是它必须是可满足的:

$$\forall s \in \text{IUC}, \forall u \in \mathbb{B}^r, \exists c: X(s, u, c) = 1 \quad (4)$$

我们的实现算法依赖于在可控变量 $c_i, i = 0 \cdots p-1$ 上对 X 作香农分解:

$$X = \overline{c_i} \cdot X|_{c_i=0} + c_i \cdot X|_{c_i=1}.$$

变量 c_i 对 X 的可满足性的影响总结在真值表 1 中. 我们的目标是寻找使得 X 满足的 c_i 值的表达式. 满足 X 的 c_i 的值被总结在表 2 中. 表 2 总结了使得监督器方程(1)成立的可控变量 c_i 的取值. 由表 2 可知, $X|_{c_i=0}$ 与 $X|_{c_i=1}$ 同时为真时,无论 c_i 的取值,所有被 X 容许的状态转移都指向 IUC. 此可看作控制的非确定性在可控变量 c_i 上的投影. 当 c_i 取值不影响 X 的满足时,我们令其取值由环境变量 c_i^{env} 驱动. 从结构上看,变量 c^{env} 为辅助环境(输入)变量,如图 2 所示.

表 1 X 在变量 c_i 上的香农分解真值表

$X _{c_i=1}$	$X _{c_i=0}$	c_i	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

表 2 满足 X 的 c_i 值

$X _{c_i=1}$	$X _{c_i=0}$	c_i
0	1	0
1	0	1
1	1	0 或 1 (c_i^{env})

由表 2 可知,以下布尔表达式计算可控变量 c_i 对应的 f_i 的值:

$$f = \overline{X|_{c_i=0}} \cdot X|_{c_i=1} + c_i^{\text{env}} \cdot X|_{c_i=1} \cdot X|_{c_i=0} \quad (5)$$

式(5)代表了控制器实现算法 Algorithm 1 的基本计算步骤:首先假设 X 成立,计算 f_0 ;将 X 内的 c_0 替换为 f_0 ,得到 X_1 ;然后假设 X_1 成立,计算 f_1 ,依此类推.

算法 1 控制器实现算法

Require: X , 可满足的监督器

{由 X 计算 $f_0, \cdots, f_{p-1}$ }

{中间结果: $X_0, \cdots, X_p$ }

$X_0 \leftarrow X$

for $i = 0 \rightarrow p-1$ do

$f_i = \overline{X_i|_{c_i=0}} \cdot X_i|_{c_i=1} + c_i^{\text{env}} \cdot X_i|_{c_i=1} \cdot X_i|_{c_i=0}$

$X_{i+1} \leftarrow$ 以 f_i 替换 X_i 中的 c_i

end for

由算法 1 得到如下函数向量:

$$f = \begin{bmatrix} f_0(s, u, c_0^{\text{env}}, f_1, \cdots, f_{p-1}) \\ f_1(s, u, c_1^{\text{env}}, f_2, \cdots, f_{p-1}) \\ \cdots \\ f_{p-1}(s, u, c_{p-1}^{\text{env}}) \end{bmatrix}$$

以及一个最后一次替换后的残留表达式 $X_p(s, u)$. 为了确保 f 生成的所有控制解都被 X 接受,我们必须证明 $X_p \equiv 1$, 即 $\forall s \in \text{IUC}, \forall u \in \mathbb{B}^r: X_p = 1$.

定理 1 X_p 永为真.

证明: 通过持续以 f_i 替换 X_i 中的 c_i , 得到 X_p 的如下表达式:

$$X_p = \exists c_{p-1}, \dots, \exists c_0 : X(s, u, c)$$

即, $\forall s \in IUC, \forall u \in \mathbb{B}^r : X_p$ 成立, 这与我们初始的监督器有解的假设(4)一致.

该定理表明通过持续以 f 替换 X 中的 c , 得到一个永为真的残部. 故所有 f 生成的控制解都被包含在 X 内. 相反地, 需要证明:

定理 2 X 包含的所有控制解都能由 f 重现.

证明: 上述陈述等价于

$$\forall s, \forall u, \forall c : (X(s, u, c) \Rightarrow \exists c^{env} : c \Leftrightarrow f(s, u, c^{env}))$$

通过应用存在量化算子及替换 f 表达式, 该蕴含的右侧等价于

$$X + \prod_{i=0}^{p-1} c_i \cdot \overline{X} \mid_{c_i=0} \cdot \overline{X} \mid_{c_i=1},$$

故上述蕴含永远为真.

需要指出的是, 我们得到的控制器 f 依赖于香农分解的变量顺序. 该顺序确立了一个求值优先级: f_{p-1} 首先被计算, 其值被用于计算 $f_{p-2}, \dots, f_0$.

4 实例

4.1 串并转换器

我们通过实例展示该技术的应用, 并将控制器实现技术引入到硬件设计领域, 期望构造一个“构造即正确”(correct-by-construction)的串并转换器. 图3展示了该串并转换器的内部结构. 它由两个组件构成: 一个串行缓冲器(SB)和一个并行字构造缓冲器(WCB). 建模细节可参考文献[6].

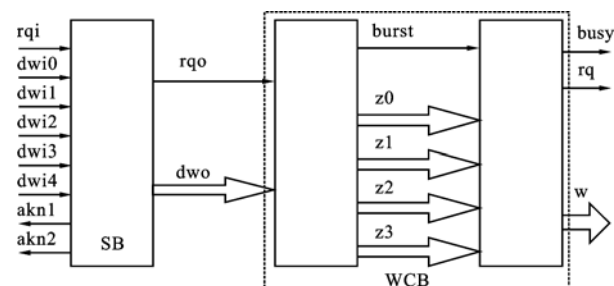


图3 串并转换器结构

组件 SB 接收 4 比特宽度的串行数据并执行奇偶校验. 如校验未通过, SB 通过 akn 信号通知数据发送端. 发送端通过 dw 端口写数据, 并通过 rqi 信号通知系统数据已可用. 一个时钟周期后, SB 通过应答机制将奇偶检查结果通知发送端. 因此, 一次数据接收需要两个时钟周期. 注意该应答机制只被用

来传递奇偶校验结果. 另外, 数据发送端会尝试重新发送数据, 直到收到正确的应答信号.

SB 模块将所有正确接受的串行数据发送给 WCB 模块. WCB 模块:

(1) 将串行数据积累在内部缓冲区, 构造 8 比特字节;

(2) 当内部缓冲区满时, 一次性输出所构造的并行数据字. 在输出期间, WCB 发出 $busy$ 信号, 表明其暂时不能接收新的数据, 输出结束后方可接收新数据.

SB 具有流水线特征, 即在给定时刻 t , 它可以传送串行数据给 WCB, 同时又从数据发送端接收新数据.

4.2 形式验证及错误分析

模块 SB 和 WCB 分别是独立设计及验证的成熟模块, 它们各自具有正确的功能, 但是设计时并未考虑协同工作的情形. 我们希望通过组合这两个已经存在的模块来构造一个串并转换器. 但是当 SB 和 WCB 组合在一起时, 得到的系统不能满足预期功能. 转换器必须具备的一个重要特性就是所有接收到的串行数据最终必须被发送出去, 即不能有数据丢失. 该特性可以由以下 CTL 时态逻辑表示:

$$\text{No data loss: AG } \overline{\text{busy}} \wedge \text{rqo}.$$

因此, 在 WCB 输出并行数据时, SB 不应再向其发送数据. 该特性仅在一种情形下不成立: 在某时刻 t , WCB 的内部缓冲区基本填满, 仅能接收一次串行数据. 此时, SB 向 WCB 发送一次串行数据, 并同时向数据发送端应答接收下一段数据. 在 $t+1$ 时刻, WCB 缓冲区已满, 并开始向外部发送并行数据. 而同时 SB 向 WCB 传送前一时刻收到的数据. 由于 WCB 忙, 该数据丢失. 以上情形被符号化模型检查(model-checking)所证实, 印证了特性“无数据丢失”是不符合实际的.

我们的目标是在不修改模块 SB 和 WCB 的情况下, 确保特性“无数据丢失”成立. 我们利用符号化离散控制器综合技术来生成一个“补丁”, 使得上述特性得以满足.

4.3 离散控制器综合

为了将应用离散控制器综合技术应用在硬件设计上, 设计者必须指定哪些输入变量是可控的. 没有成熟统一的法则来确定可控输入. 就此串并转换器而言, 观察到可以通过在合适的时刻延迟发送端的数据到来, 以避免数据丢失, 即延迟数据输入请求 rqi 信号. 因此, 选择 rqi 作为可控输入变量. 其余输

入变量为不可控变量。

监督器由工具 Sigali^[7] 综合生成, 我们利用实现算法 Algorithm 1 将监督器转换为控制器。完整的受控串并转换器结构如图 4 所示, 对应于改进的闭环控制架构图 2。

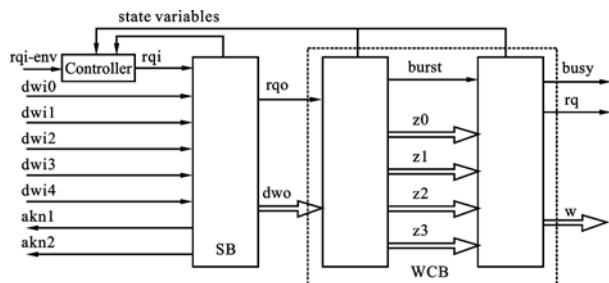


图 4 基于离散控制器的硬件错误自动修正

我们的实现算法针对可控变量 rqi 引入了辅助环境变量 $rqi-env$, $rqi-env$ 取代 rqi 成为外部数据传入请求, 由系统外部环境驱动, 并由控制器读取。根据 $rqi-env$ 的值以及受控系统内部状态, 控制器计算合适的值并赋给 rqi 。

得到的控制器扮演了外部传输请求 $rqi-env$ 过滤器的角色。当外部串行数据抵达时, 由控制器决定是否将请求传送给 SB。如果控制器判断该数据有可能会丢失, 则会阻塞该请求。在任意时刻 t , 当有外部传输请求 $rqi-env$ 时, 存在两种可能的情景:

(1) WCB 内部缓冲区仅剩一个空闲存储位置, 且该位置马上要被 SB 先前收到的数据占用。此时, 如果收到 $rqi-env$ 请求, 在没有控制器的情况下数据会丢失, 而控制器令 $rqi=0$, 从而推迟了外部数据传入 SB, 避免数据丢失;

(2) WCB 内部缓冲区有多于一个的空闲存储位置。此时收到 $rqi-env$ 信号时, 控制器会令 $rqi=1$, 外部数据请求被传递给 SB, 数据被接收。

由此可见, 当外部传入请求被延迟时, SB 对此并不知情。原本的外部应答机制仅用于保证奇偶校验; 通过应用离散控制器, 该机制被扩展用以保证数据不会丢失。

控制器将控制的非确定性转移到系统外部环境。当 rqi 的值存在非确定性选择时 (上述第二情形), 该非确定性由外部环境通过 $rqi-env$ 加以解决。实例中, 环境赋予 $rqi-env$ 的值, 此时被直接传递给 rqi 。

由此实例可知, 错误修正控制器建立的输入过滤机制严重依赖于非确定性假设。事实上, 如果综合出的监督器是完全确定性的, 则本文的实现算法会产生一个真正的闭环系统。这样的系统虽然在自动

控制领域更受欢迎, 但不适合自动错误修正。

我们得到的控制器是由输入变量和状态变量构成的二叉决策树 (BDD), 其包含大约 200 个节点, 故不在此显示。

5 实现

该串并转换器由 Mode Automata 语言及 matou 编译器建模。监督器由 Sigali 符号化综合工具生成。交互式仿真由专用工具 SigalSimu 实现。完整设计流程如图 5 所示。

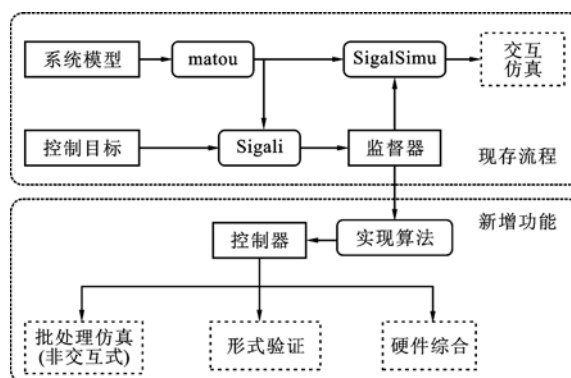


图 5 现存及改进的综合框架

我们的实现算法基于 CUDD 库实现。算法读取 Sigali 生成的监督器并产生一个由二叉决策树表示的若干控制函数组成的控制器。在此之后, 受控系统与实现后的控制器可以一并作批处理仿真, 硬件综合或形式验证。

6 结束语

本文提出了一种针对符号化控制器综合技术的监督器实现技术。该技术同时解决了控制的非确定性问题, 以及监督器同模块化被控系统的结构不兼容性问题。该技术保留了控制解的完整性。实现后的控制器 f 通常远比监督器紧凑, 从而更容易处理。

将传统控制架构进行改进后, 引入硬件设计领域作设计错误的自动修正。对于硬件设计来说, 可控/不可控输入变量的区别并不是很明显。输入变量通常由环境操控, 而不能被简单地封装在控制闭环中。提出的控制架构保持了系统输入/输出接口的不变。可控输入的值实际上由控制器“过滤”, 使得系统的行为不会违反控制目标。

本文提出的实现算法已经被整合在由法国计算机及自动化研究院 (INRIA) 研发的 BZR 语言及其编译仿真系统中, 并在法国科学院 (CNRS) 与庞巴

(下转第 120 页)

电池是否尾端破皮有现象。

4 结束语

本文从机器视觉领域中抽取电池尾端作为研究对象,结合 Matlab 仿真工具,完成了对电池尾端破皮检测。在频谱分析比较困难的情况下,通过仿真确定了以形态学为基础的另一种解决方法,其步骤为:图像抓取、二值化、去噪点、闭操作、统计连通区域。通过连通区域的统计和区域个数阈值的设定,可以检测一个电池尾端是否具有破皮的现象,并且这种检测能够唯一的识别出这种伤痕。在伤痕检测领域中,本文提到的方法还可用于其他平面检测领域,本文只是研究了众多伤痕种类的一种,机器视觉在自动化生产上的应用非常广泛,相关的伤痕判定算法也不断的推陈出新。通过机器视觉不断的发展,一定会将自动化生产带到一个新台阶。

参考文献:

- [1] Rafael C Gonzalez, Richard E Woods. Digital image processing, third edition[M]. US: Publishing House of Electronic Industry, 2011: 402-407.
- [2] Milan Sonka, Vaclav Hlavac, Roger Boyle. Image processing, analysis, and machine vision, Third Edition[M]. [s. l.]: Nelson Engineering, 2007.
- [3] 王磊. 基于机器视觉电池表面缺陷检测技术研究[D].

合肥:中国科学技术大学,2011.

- [4] 宋晓闯. 基于灰度和几何特征的图像匹配算法研究[D]. 天津:河北工业大学,2008.
- [5] 何小利,宋珏. 基于机器视觉技术的物体表面缺陷检测[J]. 洛阳师范学院学报,2011,30(2):64-68.
- [6] 韩芳芳. 表面缺陷视觉在线检测关键技术研究[D]. 天津:天津大学,2011.
- [7] 刘刚,刘增力,陈盛云. 一种改进的基于平移不变小波去噪方法及应用[J]. 云南大学学报:自然科学版,2009,31(2):219-222.
- [8] 靳士利. 基于小波变换的自适应图像去噪方法的研究[D]. 青岛:青岛大学,2010.
- [9] 田进,李言俊,王红梅. 基于平移不变小波变换的图像滤波算法[J]. 弹箭与制导学报,2012,32(6):140-142.
- [10] 张德丰. MATLAB 数字图像处理[M]. 2 版. 北京:机械工业出版社,2011:309-351.
- [11] 李俊. 机器视觉照明光源关键技术研究[D]. 天津:天津理工大学,2006.

作者简介:



朱柯润 男,(1989—),硕士研究生. 研究方向为计算机视觉.

(上接第 116 页)

迪(BOMBARDIER)联合研究项目 FerroCOTS 中得到应用。

参考文献:

- [1] Ramadge PJ G, Wonham W M. The control of discrete event systems[J]. Proceedings of the IEEE, 1989,77(1):81-98.
- [2] Hervé Marchand. Méthode de Synthèse d'automatismes décrits par des systèmes à événements discrets finis[D]. PhD thesis, Université de Rennes I, 1997.
- [3] Arash Vahidi, Martin Fabian, Bengt Lennartson. Efficient supervisory synthesis of large systems[J]. Control Engineering Practice, 2006,14(10):1157-1167.
- [4] Kenneth L McMillan. Symbolic model checking - an approach to the state explosion problem[D]. PhD thesis, Carnegie Mellon University, 1992.
- [5] Eugene Asarin, Oded Maler, Amir Pnueli. Symbolic controller synthesis for discrete and timed systems[J]. Hybrid Systems II lecture Notes in Computer Sci-

ences, 1995(9):1-20.

- [6] Dumitrescu E, Ren M. Automatic error correction based on discrete controller synthesis[C]// The 4th International Federation of Automatic Control Conference on Management and Control of Production and Logistics. Sibiu: IFAC, 2007.
- [7] Marchand H, Bournai P, Le Borgne M, et al. Synthesis of discrete - event controllers based on the Signal environment [J]. Discrete Event Dynamic System: Theory and Applications, 2000,10(4):325-346.

作者简介:



任明明 男,(1981—),博士后. 研究方向为信息系统.