基于组件自动机的概率连续行为的形式化模型

卜星晨,曹子宁,王福俊
(南京航空航天大学 计算机科学与技术学院,江苏 南京 211106)

摘 要:信息物理融合系统是由具有离散性的信息系统和具有连续性的物理系统通过端口进行数据、控制信号的通信组合而成。采用形式化方法对信息物理融合系统进行描述和验证,对于提高系统的正确性、可靠性和安全性具有重要的意义。信息物理融合系统的系统特性与组件交互自动机的建模思想相吻合。在现有的组件交互自动机的基础之上,提出了一种新的组件交互自动机——概率混成组件交互自动机,并给出了相关的定义及其进行组合的算法。概率混成组件交互机与之前的组件自动机相比,引入了状态迁移的不确定性以及状态内部的连续动态性,既能够描述系统状态上的不确定状态迁移,又能够对状态内部的连续行为特性进行刻画,便于对信息物理融合系统内部的不确定性和连续性进行很好地描述。

关键词:信息物理融合系统;组件交互自动机;概率迁移;连续动态性;形式化方法
中图分类号:TP301 **文献标识码**:A **文章编号**:1673-629X(2020)11-0001-06
doi:10.3969/j.issn.1673-629X.2020.11.001

Formal Model of Probabilistic Continuous Behavior Based on Component Automata

BU Xing-chen,CAO Zi-ning,WANG Fu-jun
(School of Computer Science and Technology,Nanjing University of Aeronautics and Astronautics,Nanjing 211106,China)

Abstract:The Cyber-physical system (CPS) is composed of discrete information systems and continuous physical systems which communicate data and control signals through ports. It is of great significance to describe and verify the CPS by using formal method for improving the correctness,reliability and security of the system. The system characteristic of the CPS is consistent with the modeling idea of the component interaction automaton. Based on the existing component interaction automata,a new component interaction automata,probabilistic hybrid component interaction automata,is proposed,and its definition and algorithm are given. Compared with the previous component automata,the probabilistic hybrid component interaction automata can model the uncertainty of state transition and the continuous dynamics inside the state. It can not only describe the uncertain state transition on the system state,but also describe the continuous behavior characteristics inside the state,which is convenient to describe the uncertainty and continuity inside the CPS.

Key words:cyber-physical system;component interaction automata;probability transition;continuous dynamic;formal method

0 引 言

基于组件的开发技术正逐渐应用到信息物理融合系统(CPS)[1-2]的开发过程中。现已有许多基于组件的软件系统开始应用到航空航天、军事过程控制等领域当中,这些领域往往对系统的正确性和可靠性要求比较高。系统执行过程中的一个小小的错误都有可能导致巨大的灾难。而这类系统往往同时具有连续性和离散性,且每个子系统之间存在着频繁的数据通信,因而给这类系统的准确建模带来了一定的困难。

基于组件[3-4]的开发技术是一种将系统组件化、然后再将组件化后的系统组件进行组合的技术。采用形式化的方法[5]对CPS进行建模与验证,能够很好地保证系统的正确性和安全性。形式化的验证过程一般采用形式化建模语言对系统进行建模,采用逻辑语言刻画所要验证的性质,最后使用形式化验证工具自动地对模型和逻辑规约进行验证。文献[6]指出在基于组件的开发过程中引入形式化的建模方法和自动化验证技术,是一个重要的研究方向。

收稿日期:2019-10-24 **修回日期**:2020-03-05
基金项目:国家航空科学基金(20185152035,20150652008);中央高校基本科研业务费(NJ2019010)
作者简介:卜星晨(1995-),男,硕士研究生,CCF会员(88989G),研究方向为形式化方法;曹子宁,教授,博导,研究方向为形式化方法、人工智能。

该文针对CPS,在组件自动机的基础上扩展了概率、连续行为特性,提出了概率混成组件交互自动机,并给出了具体的定义及其进行组合验证的算法。提出的概率混成组件交互自动机不仅能够描述组件之间的交互行为,还能能够清楚地刻画出系统的体系机构以及组件内的连续动态性和状态迁移上的概率选择性。

1 背景和动机

CPS是一种将计算过程与物理过程组合在一起的实时反应式系统。其最主要的特征是系统中同时包含离散性和连续性。CPS本身是一种基于组件的系统,每一个软件实体或者物理设备都能够独立运行,每个子系统之间可以通过信号的传输实现相互合作的功能。

现已有许多能够对混成系统进行建模的形式化方法,如混成自动机[7-8]和混成I/O自动机[9-10]。然而混成I/O自动机是输入使能的,即混成I/O自动机中的每一个状态都必须能够接收所有可能的输入动作,而混成自动机并不能够将组件之间的通信组合方式进行准确建模。因此,需要一种新的形式化建模方法对CPS进行建模。

组件交互行为的建模方法可以分为两大类:基于进程代数的方法和基于自动机的方法。基于进程代数的方法主要使用CSP/Timed CSP、Pi演算、CCS/Timed CCS等进程代数语言进行建模。基于自动机的方法主要使用接口自动机[11]、I/O自动机、时间自动机[12]以及时间I/O自动机[13]进行建模。组件交互自动机[14-16]在自动机的基础上根据进程代数的思想扩充了组件之间交互行为的描述,使其能够保留组件之间的交互属性。

1.1 组件交互自动机

组件交互自动机是一种非确定的自动机,其中每一个迁移动作都标注为输入、输出或者内部动作。输入(输出)动作与接收(发出)该动作的组件相关联。内部动作与动作上同步的两个组件相关联。在组件交互自动机的一次组合过程中,只有两个组件可以进行同步操作。组件交互自动机的定义如下:

定义1:组件交互自动机COIA可以写成元组的形式 (Q,Q_0,A,Σ,E,H),其中:

(1) Q 表示非空的有限状态集合;

(2) $Q_0 \subseteq Q$ 表示初始状态集合;

(3) A 表示有限动作集,包括输入动作、输出动作、内部动作 τ 和空动作 ε;

(4) Σ 表示动作约束符号表的集合,$\Sigma = \{(C) \times A \times (C \cup (\pm))\}$,$C$ 表示组件及其接口的名称,$C \in N(H)$,$N(H)$ 表示组件组合过程中涉及到的组件名称的集合,+和-分别表示输入动作和输出动作;

(5) $E \subseteq Q \times \Sigma \times Q$,表示状态迁移的集合;

(6) H 是用来表示组件内部组件组合层次的元组,元组 $H_i = (n_{i1}, n_{i2}, \cdots, n_{im})$ 表示组件 H_i 由 m 个组件在同一层次上组成而成。括号用来表示组合的层次性,最内层的组件最先进行组合,然后依次向外进行组合。

$(A,a,+),(B,a,-),(B,a,A) \in E$ 分别表示输入、输出和内部动作。$(A,a,+)$ 表示组件 A 接收动作 a,$(B,a,-)$ 表示组件 B 发送动作请求 a,(B,a,A) 表示组件 B 输出动作 a,组件 A 接收到动作 a。

1.2 动 机

CPS是基于组件的系统,这为使用COIA对CPS进行建模提供了良好的基础。现有的组件交互自动机只支持离散系统的建模,并不支持概率以及连续行为性质的建模。因此,对组件交互自动机进行了相应的扩展,在组件交互自动机的基础上提出了概率混成组件交互自动机PHCOIA来对CPS进行建模。

2 概率混成组件交互自动机

PHCOIA在COIA的基础上添加了变量集 X,变量集 X 的不同取值表示系统所处的不同状态。PHCOIA在COIA的基础上对状态上的动作变迁,增加了关于变量集合 X 的约束条件Enab,只有满足了相应的约束条件之后,才能执行动作进行状态变迁。

现实生活中总会存在一些不确定因素对系统的执行造成一定的影响,因此在COIA的状态迁移上添加了概率分布 P,表示状态变迁上的不确定选择。除此之外,还定义了赋值函数 F_a 和流函数 F 分别表示对状态内变量的赋值操作以及定义状态内变量的变化速率。通过定义不变式 I 约束每个状态内变量的变化范围。PHCOIA的定义如下:

定义2:PHCOIA可以写成元组的形式 $(Q,Q_0,A,\Sigma,X,X_{\text{init}},F_a,F,\text{Enab},E,I,P,H)$,其中:

(1) Q 表示非空的有限状态集合;

(2) $Q_0 \subseteq Q$ 表示初始状态;

(3) A 表示有限动作集,包括输入动作 A^{in}、输出动作 A^{out}、内部动作 A^{int} 和空动作 ε;

(4) Σ 表示动作约束符号表的集合,$\Sigma = \{(C) \times A \times (C \cup (\pm))\}$,$C \in N(H)$;

(5) X 表示实值变量的有限集合;

(6) X_{init} 表示变量集合 X 在初始状态的取值;

(7) $F_a: Q \rightarrow \text{assign}(X)$ 表示赋值函数,用来对状态中的变量进行赋值;

(8) $F: Q \times \mathbb{R}^{|X|} \rightarrow \mathbb{R}^{|X|}$ 表示状态上变量的流函数,F 函数定义了变量的变化率,通常使用变量对时间

的一阶导数进行表示;

(9) Enab:$Q \times \Sigma \rightarrow$ guard 表示状态执行动作迁移的谓词条件, guard 为变量集合 X 上的布尔表达式;

(10) E 表示状态迁移的集合, $E \subseteq Q \times \Sigma \times Q'$;

(11) I 为状态 $q \subseteq Q$ 上的不变式条件,用来限制变量的变化范围,不变式条件通常使用变量集合 X 上的布尔表达式进行表示;

(12) P 是概率分布, $P:Q \times \Sigma \times Q' \rightarrow [0,1]$,表示状态迁移发生的概率, $\forall q' \in Q$, $\forall \alpha \in \Sigma$, $\sum_{q' \in Q}(q,\alpha,q') = 1$;

(13) H 是描述该组件内组件组合层次的元组。

3　PHCOIA 的迁移系统语义

PHCOIA 中的状态由系统当前所处位置 $q \in Q$ 和变量的取值组合而成。PHCOIA 中有两种迁移方式:动作迁移和时间迁移。动作迁移:当状态内的变量值满足相应的谓词条件 guard 时,当前状态将会执行动作 Σ 跳转到下一个状态;时间迁移:系统停留在当前位置 q ,但系统中的变量随着时间的流逝不断地发生变化。时间迁移的边标记为实数值,表示系统在位置 q 上停留的时间。Δ 表示所有迁移边的集合, $\Delta = \Sigma \cup R_{\geqslant 0}$ 。

定义 3:概率混成组件交互自动机 $M = (Q,Q_0,A,\Sigma,X,X_{\text{init}},F_a,F,\text{Enab},E,I,P,H)$,其迁移系统可以定义为 $\text{TR}(M) = (S,S_0,\Delta,T)$, 其中 S 表示状态集合, $S = \{ < Q,\text{val}(\vec{X}) > | \ \text{val}(\vec{X}) \models I(Q) \}$, $\text{val}(\vec{X})$ 表示变量集 X 上的一组取值, $(q,\vec{v}) \in S$ 表示迁移系统中的状态,其中 $q \in Q$; $S_0 = \{ < Q_0,\text{val}(\vec{X}) > | \ \text{val}(\vec{X}) = X_{\text{init}} \}$, S_0 表示迁移系统的初始状态集合; $\Delta = \Sigma \cup R_{\geqslant 0}$ 表示所有迁移边的集合;PHCOIA 上的状态迁移 T 有两种形式:

动作迁移,状态 $(q,\vec{v})$ 执行动作集合中的动作 $\alpha \in \text{Act}$ 跳转到下一个状态:if($\vec{v} \models \text{Enab}(q,\alpha)$), $(q,\vec{v}) \xrightarrow{\alpha} \{(q',\vec{v}') | \ (P(q,\alpha,q') > 0) \wedge (\vec{v}' = F_a(q)) \wedge (\vec{v}' \models I(q'))\}$;

时间迁移,状态 $(q,\vec{v})$ 在当前位置 q 停留 $d \in \mathbb{R}_{\geqslant 0}$ 时刻: If($(d' \in [0,d]) \wedge (\vec{v}' = \vec{v} + \int_t^{t+d} F(q)\,\text{d}t \ | \ I(q))$), $(q,\vec{v}) \xrightarrow{d} (q,\vec{v}')$;

PHCOIA 的迁移系统从初始状态 S_0 开始执行,如果当前状态满足动作迁移的使能条件 Enab,系统将会根据概率分布 P 从所有可能的后继状态中选择一个状态进行跳转;如果当前状态不满足变迁的使能条件,状态将会执行时间迁移,状态内的变量根据流函数随着时间的变化而发生变化。

PHCOIA 的执行路径是一条有限或无限的状态迁移序列,路径上的状态迁移在时间迁移和动作迁移之间交替执行,如: $\pi = s_0(d_0)s_1(\alpha_1)s_2(d_2)s_3(\alpha_3)s_4\cdots$, d_i 表示时间迁移, α_i 表示动作迁移。$\text{dur}(i)$ 表示到达状态 i 所经历的时间: $\text{dur}_\pi(i) = \sum_{0 \leqslant j \leqslant i} d_j$ 。$\pi(s)$ 表示从状态 s 开始的路径集合。路径 π 上的概率度量 $\Pr(\pi)$ 等于路径上所有动作迁移的概率乘积。

时间延迟只发生在位置上,动作变迁的时间忽略不计。因此,如果不对状态变迁进行进一步的约束,就有可能出现自动机在有限时间内执行无限多动作的情况,即 Zeno 行为。然而现实生活中这样的情况是不存在的,即不存在可以在有限时间内执行无限多指令的计算机。因此,接下来给出结构良好性标准。

定义 4:如果对于 $\text{TR}(M)$ 中的所有状态 s ,满足 $\forall n \in \mathbb{N}$, $\exists \pi \in \pi(s)$, $\exists i \in \mathbb{N}$, $\text{dur}_\pi(i) > n$,即对于任意给定的时间 $n \in \mathbb{N}$,总是存在一条从状态 s 开始的路径 $\pi(s)$,从状态 s 到达路径 $\pi(s)$ 上某一状态的时间大于 n ,则称 PHCOIA 满足结构良好性标准。

该文所定义的 PHCOIA 不包含齐诺行为。只有满足结构良好性标准的 PHCOIA 才能进行有效的组件组合。

4　PHCOIA 的组合与验证

4.1　PHCOIA 组合的相关概念

在交互自动机的组合过程中,需要匹配对应的输入输出动作。在状态的组合过程中,除了需要组合状态内的变量集外,还需要组合状态上的赋值函数、流函数和不变式约束,同时在生成的状态集上生成对应的迁移关系以及对应的概率分布,以构造新的组件自动机。

定义 5:如果两个状态内的赋值函数、流函数和不变式约束中出现的变量集合不相交,则称这两个状态可以进行组合。

定义 6:PHCOIA M 有输出动作 $(C_i,\alpha_i,-) \in \Sigma_M$, $i \in \mathbb{N}$,PHCOIA N 有输入 $(C_k,\alpha_k,+) \in \Sigma_N$, $k \in \mathbb{N}$ 。将 M 和 N 进行组合,如果 α_i 和 α_k 动作名称相同且输出动作 α_i 前后两个状态可以与输入动作 α_k 前后两个状态分别进行组合,则称输出动作 α_i 与输入动作 α_k 匹配。在不考虑内部动作细节的情况下,内部动作可以简记为 (C_i,τ,C_k) 。

定义 7: $D(H)$ 表示组件组合层次的深度。 $H = (C)$,表示 H 内只有一个元素 C ,即该组件是一个原子组件,组件的组合深度为 1。

定义8 :PHCOIA的组合 $S=(Q,Q_0,A,\Sigma,X,X_{\text{init}},F_a,F,\text{Enab},E,I,P,H)$,是由PHCOIA的集合 $\{M_i=(Q_i,Q_{0i},A_i,\Sigma_i,X_i,X_{\text{init}_i},F_{ai},F_i,\text{Enab}_i,E_i,I_i,P_i,H_i)\}$ 组合而成,其中:

(1) $i\in N$ 且参与组合的组件名称集合 $\{(N(H_i))\}$ 互不相交;

(2) Q,Q_0 分别为组合后的状态集合和初始状态集合, $Q\subseteq\coprod_{i\in\mathbb{N}}Q_i$, $Q_0=\coprod_{i\in\mathbb{N}}Q_{0i}$; $A=\bigcup_{i\in\mathbb{N}}A_i$,集合 A 表示组合后的动作集合,若组合过程中两个自动机的输入输出动作匹配,则将原输入输出动作从动作集合中去除,将新产生的内部动作添加到动作集合中,同时更新动作约束符号集合 Σ ; $X=\bigcup_{i\in\mathbb{N}}X_i$, $X_{\text{init}}=\bigcup_{i\in\mathbb{N}}X_{\text{init}}$, X 和 X_{init} 分别表示组合后的变量集合和变量的初始值;

(3)动作约束符号 $\Sigma\subseteq\{C\times A\times(C\cup(\pm))\}$, $C\in N(H_i)\cup N(H_k)$;

(4) $\text{Enab}:\coprod_{i\in\mathbb{N}}Q_i\times\Sigma\rightarrow\text{guard}$ 表示状态执行动作迁移的谓词条件; $E\subseteq\coprod_{i\in\mathbb{N}}Q_i\times\Sigma\times\coprod_{i\in\mathbb{N}}Q_i$ 表示组合后状态上的动作迁移集合; $P:\coprod_{i\in\mathbb{N}}Q_i\times\Sigma\rightarrow[0,1]$ 表示动作迁移的概率分布;

(5) F_a,F,I 分别为组合后状态上的赋值函数、流函数以及不变式约束;

(6) H 为组件内组件组合层次的元组。

在自动机的组合过程中,需要考虑动作的匹配,如果两个自动机存在可以匹配的动作,且这两个动作的使能条件可以同时满足时,这两个动作可以同步生成为一个内部动作;如果两个动作不匹配,那么这两个动作将会交替执行下去。两个自动机在组合过程中既可能存在同步互补动作,也可能存在交替执行动作。

4.2 PHCOIA的组合算法

PHCOIA组合算法,将 M_1,M_2 进行组合, $M_1=(Q_1,Q_{01},A_1,\Sigma_1,X_1,X_{\text{init}_1},F_{a1},F_1,\text{Enab}_1,E_1,I_1,P_1,H_1)$, $M_2=(Q_2,Q_{02},A_2,\Sigma_2,X_2,X_{\text{init}_2},F_{a2},F_2,\text{Enab}_2,E_2,I_2,P_2,H_2)$,输出为组合后的自动机 $S=(Q,Q_0,A,\Sigma,X,X_{\text{init}},F_a,F,\text{Enab},E,I,P,H)$ 。下面给出PHCOIA组合算法的具体步骤:

1、对自动机 M_1 和 M_2 上的输入动作集的交集和输出动作集的交集进行判断,如果其中一个不为空,则返回错误信息; M_1 和 M_2 的同步共享动作集 $\text{Shared}(M_1,M_2)$ 为 $(A_1\cap A_2)\backslash\varepsilon$,自动机 S 上的变量集 X 为自动机 M_1 和 M_2 变量集合的并集 $X_1\cup X_2$,内部动作集合为 $A_1^{\text{int}}\cup A_2^{\text{int}}\cup\text{Shared}(M_1,M_2)$,输入动作集合为 $(A_1^{in}\cup A_2^{in})\backslash\text{Shared}(M_1,M_2)$,输出动作集合为 $(A_1^{\text{out}}\cup A_2^{\text{out}})\backslash\text{Shared}(M_1,M_2)$;

2、从自动机 M_1 和 M_2 的初始状态进行状态组合,如果 M_1 和 M_2 的初始状态不能进行组合,则返回错误信息;否则标记 Q_{01} 和 Q_{02} ,自动机 S 的初始状态 Q_0 为 Q_{01} 和 Q_{02} 的状态组合,将 Q_0 添加到状态集合 Q 中,状态 Q_0 内部的初始变量赋值 X_{init} 等于 Q_{01} 和 Q_{02} 状态内变量的初始值,状态 Q_0 内的赋值函数、流函数以及不变式约束分别为 Q_{01} 和 Q_{02} 上赋值函数的并集、流函数的并集和不变式约束的并集;将状态 S_1,S_2,S_{Pre} 分别置为 Q_{01},Q_{02} 和 Q_0 ,跳转到步骤3;

3、如果 M_1 和 M_2 中的状态都被标记过,则 $H=(H_1,H_2)$,组合完成程序返回;如果 S_1,S_2 都存在后继状态变迁,则跳转到步骤4;如果 S_1 存在状态变迁而 S_2 不存在,则跳转到步骤8;如果 S_1 不存在状态变迁而 S_2 存在,则跳转到步骤9;如果都不存在变迁,程序返回;

4、遍历所有分别以状态 S_1,S_2 为起始节点,形如 (S_1,α_1,S_1'), (S_2,α_2,S_2') 的状态变迁,如果 α_1 和 α_2 中的动作都是同步动作,则跳转到步骤5;如果 α_1 为同步动作而 α_2 不是,则跳转到步骤6;如果 α_1 不是同步动作而 α_2 是同步动作,则跳转到步骤7;如果 α_1 和 α_2 都不是同步动作,分别执行步骤6与步骤7中的操作;

5、如果 α_1 和 α_2 不匹配,则返回错误信息;将这两个同步动作进行组合生成内部动作约束符号 α ,将后继状态 S_1' 和 S_2' 进行组合生成状态tempState,标记状态 S_1' 和 S_2' ;自动机 S 中状态 S_{Pre} 执行 α 的使能条件为状态 S_1 执行 α_1 的使能条件与状态 S_2 执行动作 α_2 的使能条件的交集,将状态变迁 $(S_{\text{Pre}},\alpha,\text{tempState})$ 加入到自动机 S 中的状态变迁集合中,相应的状态迁移的概率为状态 S_1 到达 S_1' 的概率与 S_2 到达 S_2' 的概率的乘积。如果状态tempState不存在状态集 Q 中,则将tempState加入到状态集 Q 中,状态tempState内的赋值函数、流函数以及不变式约束分别为 S_1 和 S_2 上赋值函数的并集、流函数的并集和不变式约束的交集,将 S_1,S_2,S_{Pre} 分别置为 S_1,S_2 和tempState,跳转到步骤3;

6、将后继状态 S_1 和 S_2' 进行组合生成tempState,标记状态 S_2' ,自动机 S 中状态 S_{Pre} 执行动作 α_2 的使能条件为状态 S_2 执行动作 α_2 的使能条件,将状态变迁 $(S_{\text{Pre}},\alpha_2,\text{tempState})$ 加入到 S 中的状态变迁集合中,相应的状态迁移的概率为状态 S_2 到达 S_2' 的概率。如果状态tempState不存在于状态集 Q 中,则将tempState加入到状态集 Q 中,tempState内的赋值函数为 S_2' 上的赋值函数,tempState内的流函数以及不变式约束分别为 S_1 和 S_2' 上流函数的并集和不变式约束的并集,将 S_2,S_{Pre} 分别置为 S_2' 和tempState,跳转到步骤3;

7、步骤7的内容与步骤6的内容相似,不再重复赘述;

8、遍历所有以状态 S_1 为起始节点形如 (S_1,α_1,S_1') 的状态变迁,如果 α_1 中的动作是同步动作,则返回

错误信息;将 S_1' 和 S_2 进行组合生成 tempState,标记状态 S_1',自动机 S 中状态 S_{Pre} 执行动作 α_1 的使能条件为 S_1 执行动作 α_1 的使能条件,将状态变迁 (S_{Pre},α_1, tempState) 加入到自动机 S 中的状态变迁集合中,相应的状态迁移概率为状态 S_1 到达 S_1' 的概率。如果状态 tempState 不存在状态集 Q 中,则将 tempState 加入到状态集 Q 中,tempState 内的赋值函数为 S_1' 上的赋值函数,tempState 内的流函数以及不变式约束分别为 S_1' 和 S_2 上流函数的并集和不变式约束的并集,将 S_2,S_{Pre} 分别置为 S_1' 和 tempState,跳转到步骤 3;

9、步骤 9 的内容与步骤 8 的内容相似,不再重复赘述。

4.3　PHCOIA 的组合验证

PHCOIA 的组合算法通过遍历自动机中的所有迁移来实现两个自动机的兼容性组合。利用该算法可以直接对两个 PHCOIA 进行组合相容性错误检测。如果在组合过程中,需要组合的两个状态内包含的变量有交集,或者在动作的组合过程中,当前自动机中的同步动作与目标自动机在保证动作先后序列的前提下无法找到与之匹配的动作或者两个同步动作不匹配,则判定出现相容性错误,给出错误提醒。

5　基于 PHCOIA 的应用实例

本节使用 PHCOIA 对飞行器的对接过程进行建模。在两个飞行器的对接过程中,后面追赶的飞行器称作追赶者飞行器(chaser),需要对接到的飞行器称为目标飞行器(target),追赶者沿着半径为 r_1 的轨道以角速度 θ_1 运行。为了顺利地与目标飞行器完成对接,追赶者需要先在低轨道上改变自身的角度姿势,然后再逐渐上升到目标飞行器所在的高度进行对接。当上升到一定高度之后,追赶者将发送对接信号 dock 与目标飞行器进行对接。飞行器的每一次对接并不能保证完全的成功,存在一定的失败机率。如果对接失败,目标飞行器将会发送失败信号 error,追赶者接收到信号后将会下降自身高度,重新调整角度,准备下一次的对接;如果对接成功,追赶者将会收到目标飞行器发送的对接成功信号 success,并与目标飞行器以相同的角速度一同运行下去。

追赶者的 PHCOIA 如图 1 所示,目标飞行器的 PHCOIA 如图 2 所示。

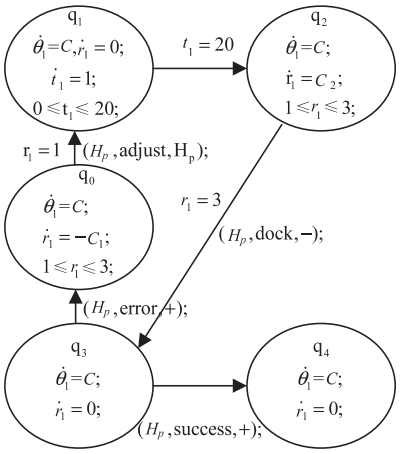

图 1　追赶者飞行器的 PHCOIA

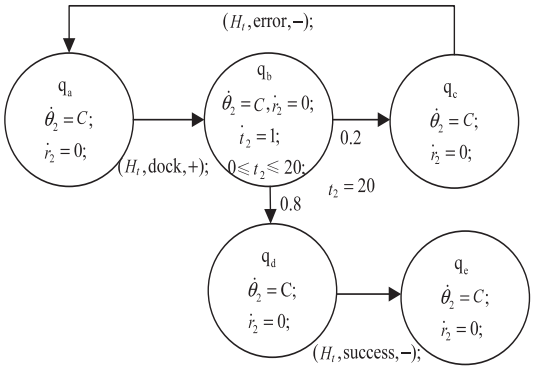

图 2　目标飞行器的 PHCOIA

两个飞行器的组合建模如图 3 所示。为了建模的简单呈现,所有的输入输出动作都组合成了内部动作。从组合后的 PHCOIA 中,可以很直观地看出对接过程中,各个系统内部的连续变化特性以及状态迁移上的概率不确定性,同时系统内部各个组件的交互行为也能够通过组件交互自动机很好地保存下来。

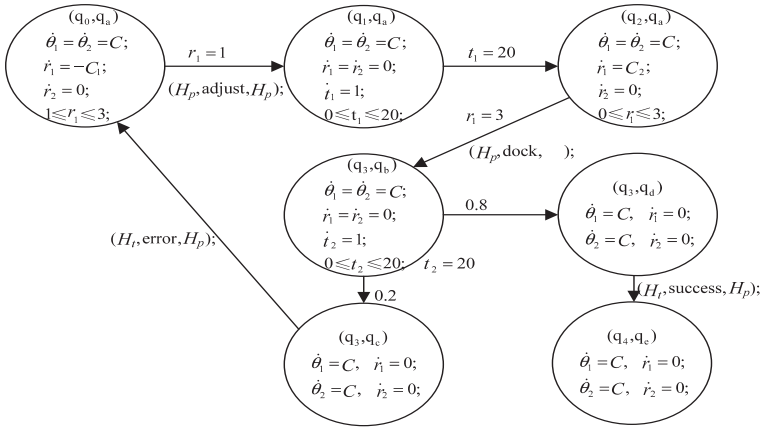

图 3　组合后的 PHCOIA

6 结束语

在COIA的基础上,针对CPS,提出了概率混成组件交互自动机的概念。PHCOIA不仅能够详细地描述组件之间的交互行为,还能够清楚地刻画出组件状态内的连续变化特性以及状态变迁上的概率选择性。可以利用PHCOIA对CPS中的各个组件进行准确建模。在给出PHCOIA的相关定义之后,还给出了其相应的组合算法,根据组合算法可以将表示各个组件的自动机进行组合,从而构建出一个完整的表示原系统的PHCOIA。在接下来的工作中,将着手在PHCOIA上开展形式化验证的研究,使其能够在PHCOIA上验证相关性质。

参考文献:

[1] 张 程,陈付龙,刘 超,等.基于XML的信息物理融合系统组件建模与仿真[J].计算机应用,2019,39(6):1842-1848.

[2] 李洪阳,魏慕恒,黄 洁,等.信息物理系统技术综述[J].自动化学报,2019,45(1):37-50.

[3] 李揭阳,李 勇,张福高.基于构件交互自动机的AADL模型转换方法研究[J].计算机技术与发展,2017,27(7):68-71.

[4] 贾仰理,张振领,李舟军.基于自动机的构件实时交互行为的形式化模型[J].计算机科学,2010,37(9):151-156.

[5] WAN Y,XU Z,MEI M. A formal method for testing reactive system[M]//Artificial intelligence and computational intelligence. Berlin,Heidelberg:Springer,2012.

[6] CHILDS A,GREENWALD J,RANGANATH V P,et al. Cadena:an integrated development environment for analysis,synthesis,and verification of component-based systems[M]//Fundamental approaches to software engineering. Berlin,Heidelberg:Springer,2004.

[7] LYGEROS J,JOHANSSON K H,SIMIC S N,et al. Dynamical properties of hybrid automata[J]. IEEE Transactions on Automatic Control,2003,48(1):2-17.

[8] BAK S,BEG O A,BOGOMOLOV S,et al. Hybrid automata:from verification to implementation[J]. International Journal on Software Tools for Technology Transfer,2019,21(1):1-18.

[9] BARTOCCI E,CORRADINI F,BERARDINI M R,et al. Modeling and simulation of cardiac tissue using hybrid I/O automata[J]. Theoretical Computer Science,2009,410(33):3149-3165.

[10] LYNCH N,SEGALA R,VAANDRAGER F. Hybrid I/O automata revisited[M]//Hybrid systems:computation and control. Berlin,Heidelberg:Springer,2000.

[11] 张 岩,胡 军,于笑丰,等.接口自动机——一种用于组件组合的形式系统[J].计算机科学,2005,32(11):214-219.

[12] 李力行,金 芝,李 戈.基于时间自动机的物联网服务建模和验证[J].计算机学报,2011,34(8):1365-1377.

[13] DAVID A,LARSEN K G,LEGAY A,et al. Real-time specifications[J]. International Journal on Software Tools for Technology Transfer,2015,17(1):17-45.

[14] CERNA I,VAREKOVA P,ZIMMEROVA B,et al. Component substitutability via equivalencies of component-interaction automata[J]. Electronic Notes in Theoretical Computer Science,2007,182(1):39-55.

[15] ZIMMEROVA B,VAREKOVÁ P,BENEŠ N,et al. Component-interaction automata approach (CoIn)[M]. [s. l.]:Springer-Verlag,2008.

[16] QIAN Zhongsheng. Component interaction automata for component-based web application testing[J]. Journal of Chinese Computer Systems,2013,34(8):1813-1818.