European Journal of Control 21 (2015) 1–13

Contents lists available at ScienceDirect

European Journal of Control

journal homepage: www.elsevier.com/locate/ejcon

Control design for nondeterministic input/output automata ☆

Yannick Nke, Jan Lunze *

Ruhr-Universität Bochum, Institute of Automation and Computer Control, Germany

A R T I C L E I N F O

Article history:
Received 19 February 2014
Received in revised form
25 November 2014
Accepted 25 November 2014
Recommended by N. Kazantzis
Available online 9 December 2014

Keywords:
Discrete-event systems
Input/output automata
Supervisory control

A B S T R A C T

This paper presents a new control design approach for discrete-event systems described by Input/Output automata. A formal design method guarantees the fulfillment of the specifications for the closed-loop system including the system safety. Necessary and sufficient conditions for the well-posedness of the control loop and the controllability of the plant with respect to the specification are proved. The control of a batch process is used to illustrate the results.

1. Introduction

1.1. Control aim and literature survey

The aim of this paper is to propose a new control design approach for discrete-event systems modeled by nondeterministic Input/Output (I/O) automata $\mathcal{N}_p$ (Fig. 1). In this framework, the plant $\mathcal{N}_p$ is in the state z_p and the controller $\mathcal{C}$ in the state z_c. For a given input $v_p = w_c$ generated by the controller, the plant reacts with an output $w_p = v_c$ and a state transition to a new state z_p'. For finite number of steps k_e, the specifications $\mathcal{S}$ on the safety and on operating constraints are given in terms of a final state $z_F = z_p(k_e)$, a state sequence $Z_s = (z_p(0), \ldots, z_p(k_e))$ or an output sequence $W_s = (w_p(0), \ldots, w_p(k_e))$. The design aim is to find a controller $\mathcal{C}$ such that the control loop depicted in Fig. 1 satisfies the selected specification z_F, Z_s or W_s, respectively.

Several approaches to control design exist in literature and a comparative overview is given in [32]. All these methods have the common goal that the controller is designed so as to avoid dangerous events or to suppress forbidden states in order to satisfy safety requirements. The differences to the approach proposed in this paper lie in the way how the controller is derived for operating constraints and how the specification $\mathcal{S}$ is modeled.

A widely used control design approach for standard automata was developed in [33], which is often referred to as RW-Theory (RWT). Instead of standard automata, I/O automata are used in this paper because they explicitly describe the action–reaction principle (causality) which is a fundamental property of technological systems. References [3,11,19,23,30] are other examples of references from the discrete-event system literature in which an explicit distinction between inputs and outputs of automata is also made. Some RWT-based approaches for I/O automata were proposed in references [4,30,31], which show important limitations of the RW-Theory for I/O automata regarding the automatic synthesis. Implementation and complexity issues are presented in [9,12,36].

The approach proposed in this paper contributes to the automatic controller synthesis for a given specification and provides sufficient information for the implementation of the controller obtained. Even though the computational complexity is not the main concern of this paper, it is addressed in Section 7.

The main difference between the I/O automata handled by RWT-based approaches and those presented here lies the interpretation of the I/O transitions. References [4,7,33] consider an I/O transition as a succession of an input event σ_i and an output event σ_o in a sequence of three states, e.g. $1 \rightarrow 3 \rightarrow 2$ in Fig. 2(a). The I/O automata used in this paper are more compact because only one state transition and two states are taken into account for each I/O transition labeled by v/w (Fig. 2(b)), where v denotes the input symbol and w the output symbol.

The I/O control law developed in [30,31] is a Moore automaton. In the framework of this paper, it would lead to an open-loop control structure in Fig. 1 at time step k, because the control signals $w_c(k)$ would be generated regardless of the current plant output $w_p(k)$ depending only on the internal state $z_c(k)$ of the controller due to the Moore property.

The approach proposed in this paper handles the plant $\mathcal{N}_p$ and the controller $\mathcal{C}$ as two well distinguished entities. Hence, z_F is the specified goal state for the plant only, whereas Z_s or W_s are state or

☆A short version of this paper was presented at the 18th IFAC World Congress, 2011 in Milano, Italy.

* Corresponding author.
E-mail addresses: yannick.nke@rub.de (Y. Nke), lunze@atp.rub.de (J. Lunze).

http://dx.doi.org/10.1016/j.ejcon.2014.11.001
0947-3580/

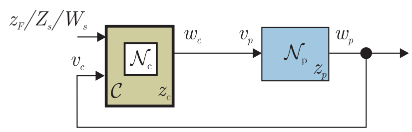

Fig. 1. Control loop of I/O automata.

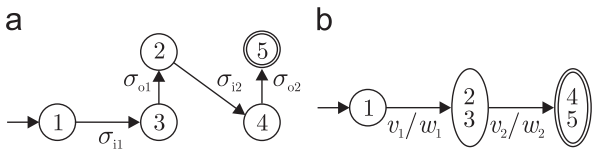

Fig. 2. Difference between a supervisor and a control law. (a) I/O supervisor and (b) I/O control law.

output sequences which have to be executed or generated by the plant, respectively (Fig. 1). However, two key properties are required for the control loop: *determinism* and *nonblockingness*. This comparison of the new method proposed in this paper and RWT-based methods proposed in literature will be extended in Section 5.5.

The strong model-matching problem for deterministic and completely defined I/O automata is studied in [5]. It consists of finding a controller for a given open-loop system with a desired closed-loop behavior. The controller synthesis proposed here is designed for the closed loop (Fig. 1) even though it can run in an open-loop manner e.g. if the control law fulfills the Moore property.

An I/O controller of sequential machines is proposed in [13] under the assumption of a deterministic control loop, whereas this paper uses the notion of weak well-posedness to catch the nondeterminism of the plant with a deterministic controller. Furthermore, the existence condition of a controller is given in [13] by nonempty entries in a Boolean reachability matrix called skeleton matrix. In this paper, the presence of a nonempty entry in such a reachability matrix is not sufficient to guarantee the achievement of the specification by the controller because of the nondeterministic behavior of the plant. The notion of safe feasibility will be introduced to solve this issue.

1.2. Problem definition

For a given plant modeled by an I/O automaton $\mathcal{N}_p$ and a specification $\mathcal{S}$, the problem is to find a controller $\mathcal{C}$ with the following requirements:

1. Fulfillment of the specification $\mathcal{S}$ by the closed-loop system.
2. Nonblockingness of the control loop. This property will be called weak well-posedness of the control loop in Eq. (19).
3. Determinism of the control output w_c at any time step. This property is called W-determinism of the controller in Lemma 1.

A specification $\mathcal{S}$ for which such a controller $\mathcal{C}$ with the control law $\mathcal{N}_c$ exists is called *safely feasible*.

This paper proposes

- a new design method of a discrete-event control law $\mathcal{N}_c$,
- an explicit realization scheme of the feedback controller $\mathcal{C}$,
- a controllability condition for the existence of the controller $\mathcal{N}_c$.

Key issues regarding the feasibility of a specification and the determinism of the control output function were initially addressed in [27]. The feasibility is extended in this paper in the sense that necessary and sufficient conditions are given for each specification type considered. Furthermore, the notion of determinism will be combined with the feasibility property in the controllability analysis.

Section 2 presents basic notions. A batch process introduced in Section 3 is used for illustration. Section 4 presents the specification modeling. The derived specification automaton is used in Section 5 for the control design method. Experimental results illustrate the applicability of the approach in Section 6. An overview of the used notation is given in Appendix Appendix A.

2. Preliminaries

2.1. Nondeterministic automata

A *nondeterministic autonomous automaton*

$$\mathcal{A} = (\mathcal{Z}, \lambda, z_0) \tag{1}$$

is defined by the following components:

- $\mathcal{Z}$ – set of states,λ
- – state transition function,
- z_0 – initial state.

The dynamics of the automaton is given by the characteristic function

$$\lambda : \mathcal{Z} \times \mathcal{Z} \to \{0,1\} \quad \text{with } \lambda(z',z) = \begin{cases} 1, & (z',z)! \\ 0 & \text{else,} \end{cases}$$

where $(z',z)!$ means that the system can carry out a state transition from z to z'.

A *nondeterministic I/O automaton* $\mathcal{N} = (\mathcal{Z}, \mathcal{V}, \mathcal{W}, L, z_0)$ has the following additional elements:

- $\mathcal{V}$ – set of control inputs,
- $\mathcal{W}$ – set of control outputs,
- L – characteristic function.

The dynamics of the automaton is given by the function

$$L : \mathcal{Z} \times \mathcal{W} \times \mathcal{Z} \times \mathcal{V} \to \{0,1\}$$

$$L(z', w, z, v) = \begin{cases} 1 & \text{if } (z',w,z,v)! \\ 0 & \text{else,} \end{cases}$$

where $(z',w,z,v)!$ means that the automaton $\mathcal{N}$ can move from state z with the input v to state z' while generating the output w. $Z(0\cdots k_e) = (Z(0), Z(1), \ldots, Z(k_e))$ represents a state sequence of k_e+1 elements denoted by $Z(k)$ with $k = 0\cdots k_e$. $\mathcal{Z}(0\cdots \boldsymbol{K_e})$ is the set of state sequences $Z_i(0\cdots k_{ei})$, where $\boldsymbol{K_e} = \{k_{e1}, \ldots, k_{e|\mathcal{Z}(0\cdots \boldsymbol{K_e})|}\}$ is the set of the corresponding time horizons. If $\boldsymbol{K_e}$ is a singleton, i.e. $\boldsymbol{K_e} = \{k_e\}$, then all corresponding state sequences have the same length k_e+1. An infinite repetition of a state sequence is characterized by the * symbol as $Z^*(0\cdots k_e)$. The symbols $\wedge$ and $\vee$ represent the Boolean AND and OR operations. Since the characteristic functions L and λ can only have the value 1 or 0, they will be used sometimes with both Boolean and arithmetic operators like $\sum$ and $\prod$.

Definition 1 (*Sub-automata and superautomata*). The I/O automaton $\mathcal{N}_2 =$ $(\mathcal{Z}_2,\ \mathcal{V}_2,\ \mathcal{W}_2,\ L_2,\ \mathcal{Z}_{02})$ is a sub-automaton of an I/O automaton $\mathcal{N}_1 =$ $(\mathcal{Z}_1,\ \mathcal{V}_1,\ \mathcal{W}_1,\ L_1,\ \mathcal{Z}_{01})$ if

$$\mathcal{Z}_2 \subseteq \mathcal{Z}_1, \quad \mathcal{V}_2 \subseteq \mathcal{V}_1, \quad \mathcal{W}_2 \subseteq \mathcal{W}_1$$

hold and if L_2 is a restriction of L_1 to the set $\mathcal{Z}_2 \times \mathcal{W}_2 \times \mathcal{Z}_2 \times \mathcal{V}_2$ in the sense of [14]. The restriction means that the behavior of $\mathcal{N}_2$ is