Scientia Iranica D (2022) 29(2), 693–726

SCIENTIA IRANICA

Sharif University of Technology
Scientia Iranica
Transactions D: Computer Science & Engineering and Electrical Engineering
http://scientiairanica.sharif.edu

On the well-posedness, equivalency, and low-complexity translation techniques of discrete-time hybrid automaton and piecewise affine systems

M. Hejri[a,*] and H. Mokhtari[b]

a. *Department of Electrical Engineering, Sahand University of Technology, Sahand New Town, Tabriz, P.O. Box 51335-1996, Iran.*
b. *Department of Electrical Engineering, Sharif University of Technology, Tehran, P.O. Box 11365-9363, Iran.*

Received 20 April 2019; received in revised form 26 August 2019; accepted 7 October 2019

KEYWORDS
Automaton-based discrete-time hybrid automaton;
Piecewise affine (PWA) systems;
Well-posedness;
Complexity;
Equivalency and translation techniques.

Abstract. The main contribution of this paper is to present the systematic and low-complexity translation techniques between a class of hybrid systems referred to as automaton-based Discrete-time Hybrid Automata (DHA) and piecewise affine (PWA) systems. As a starting point, the general modeling framework of the automaton-based DHA is represented, which models the controlled and uncontrolled switching phenomena between linear continuous dynamics including discrete and continuous states, inputs and outputs. The basic theoretical definitions on the state trajectories of the proposed DHA with forward and backward evolutions that yield forward and backward piecewise affine (FPWA and BPWA) systems are given. Next, the well-posedness and equivalency properties are proposed and the sufficient conditions under which the wellposedness property is achieved with the automaton-based DHA and PWA systems are given. It is shown that the graphical structure of the proposed automaton-based DHA makes it possible to obtain analytically the equivalent PWA system with polynomial complexity in contrast to the existing numerical translation techniques via decomposed structure of the DHA with exponential complexity. Examples are presented to confirm the effectiveness of the proposed translation techniques.

1. Introduction

1.1. Motivation and literature review

In the dynamical systems theory, the systems that combine time-driven and event-driven dynamics are called *hybrid systems* [1]. In recent years, hybrid systems have attracted much attention in both academia and industry, largely due to the embedding of event-driven microprocessors in complex automated time-driven dynamics such as automobile [2], aircrafts [3], air traffic control systems [4], process control systems [5], communication networks [6], robotics [7], biology [8], circuits and electronics [9], networked control systems [10], and power systems [11], to list just a few.

The first step in the analysis, design and synthesis, control, performance evaluation, and optimization of hybrid systems is to develop suitable mathematical models [1]. In hybrid systems, the analytical complexity of continuous dynamic systems merges with the combinatorial complexity of the discrete-event systems, leading to the fact that the analysis and synthesis of these classes of systems are very difficult. Another reason for the difficulty of mathematical treatment of

*. *Corresponding author. Tel.: +98 4133459363; Fax: +98 4123444322
E-mail addresses: hejri@sut.ac.ir (M. Hejri); mokhtari@sharif.edu (H. Mokhtari)*

doi: 10.24200/sci.2019.53308.3177

hybrid systems is the nature of their state sets as the product of their purely continuous and purely discrete subsystems state sets. Therefore, in general, it is not possible to use the rich set of analysis and synthesis approaches that work for individual continuous or discrete subsystems.

In this paper, our emphasis is on the models of hybrid systems in the discrete-time setting. It should be noted that the discretization in the time does not imply the discretization of the state space. Discretization in space is out of the scope of the present work and the interested readers are referred to [12,13] and references therein. The discrete-time sample path is piecewise constant, while the states can still take values from a set of real numbers. Of the merits of discrete-time models, one can mention their suitability for solving optimization problems and similar mathematical manipulations such as identification problems, which would be much more complex in a continuous-time setting [14,15]. Another advantage is the elimination of the zeno phenomenon which is prone to occur in continuous-time hybrid systems models. The zeno behavior is a phenomenon that is quite rare in real-life hybrid systems and arises due to the modeling abstraction [16].

In contrast, discrete-time models are subject to some limitations. Because of the continuous nature of the time and the concept of discretization, a level of approximation should be adopted in the process of passing from the original hybrid system model in the continuous-time domain to the discrete-time setting. Another issue is that when the discrete-time hybrid system model is used in the case of an optimization problem, the size of the resulting mathematical programming increases due to the introduction of a large number of binary variables, associated with each discrete-time interval in a time horizon. On the other hand, the accuracy of the discrete-time hybrid model and the size of the corresponding mathematical programming are related to each other in an opposite manner. This means that to achieve a suitable approximation of the original continuous-time hybrid model, it is usually required to use small discretization time, which in turn leads to large combinatorial problems of intractable size [17,18]. Nevertheless, when a continuous-time hybrid system model reaches the point of implementation, a part of the requirements will be realized using software in a discrete-time fashion, inevitably. In other words, the possibility of the direct design and implementation of continuous-time hybrid systems has been tailored to the discrete-time hybrid models. Therefore, an interesting and challenging problem is to develop some conditions under which the vital properties established and valid for the continuous-time hybrid models are also satisfied for the discrete-time hybrid models, and vice versa [19,20]. These are good reasons why the theory on discrete-time hybrid dynamical systems should be developed. In this regard, several discrete-time models of hybrid systems have been proposed in the literature.

In [21], the equivalency relations among some of discrete-time hybrid modeling frameworks such as Mixed Logical Dynamical (MLD) systems [22,23], Linear Complementarity (LC) systems [24], Extended Linear Complementarity (ELC) systems [25], Min-Max-Plus-Scale (MMPS) systems [26], and Piecewise Affine (PWA) systems [27,28] were presented. The existence of many equivalent modeling frameworks for a hybrid system confirms the significance of the equivalency concept and translation techniques among these modeling frameworks. A reason behind this importance is the fact that each modeling framework is suitable for a specific problem at hand. For instance, finding stability criteria and controller synthesis for PWA systems is made easily in this modeling framework [29–31], while the existence and uniqueness of the solutions can be investigated easier in LC systems [24,32] and hybrid automata [23,33]. Therefore, the study on the equivalency relations among different classes of hybrid systems is of particular importance, because it provides the possibility of transferring the theoretical properties and tools from one modeling framework to another.

Among the available discrete-time hybrid systems, PWA, MLD, and Discrete-time Hybrid Automata (DHA) have received much attention in the literature [34]. In [35], DHA was proposed as a general modeling framework to obtain hybrid models oriented toward the solution of analysis and synthesis problems. The proposed DHA has a *decomposed* structure in the sense that the continuous dynamics are described by affine difference equations, while the discrete dynamics are described by Finite State Machines (FSM). These dynamics are interfaced by Mode Selector (MS) and Event Generator (EG) elements in the decomposed structure of the proposed DHA. Furthermore, a software tool called HYbrid Systems DEscription Language (HYSDEL) is developed to obtain equivalent MLD and PWA representations [36]. While DHA is rich in its expressiveness and is, therefore, the starting point of modeling, simulation, and composition of a wide range of hybrid dynamical systems, it is not suitable to solve controller synthesis problems because of its heterogeneous discrete and continuous nature [36]. Motivated by this fact, the translation of a DHA to its equivalent MLD and PWA systems has received much attention in the literature [35,37–41]. While model predictive control techniques on the basis of online mixed integer optimization are more computationally tractable for MLD systems [42], PWA formulation is more suitable for other analysis and synthesis techniques of hybrid systems such as stability and stabilization [28,43–45]. The analysis and synthesis of PWA systems have received much attention in the literature, and one can refer to [46,47] and the references therein. Depending on the structure of the physical hybrid system, it is often a challenging task to directly obtain a PWA model of the system [48]. Therefore, it is a common practice that the original hybrid system is modeled in a convenient modeling framework such as DHA and then, is translated to the corresponding PWA system.

There are two different approaches to translating a decomposed DHA into its equivalent PWA form. In the first method, DHA can be translated into its equivalent MLD representation and then, from MLD into its equivalent form by using the proposed algorithms presented in [37,40]. As another method, the DHA is directly translated into its equivalent PWA form by using the translation techniques proposed in [38,39,41]. However, due to the decomposed structure of the original DHA model, all these translation techniques need complex and time-consuming cell enumeration and feasibility tests and their complexity grows exponentially with an increase in the dimension of the Euclidian space in which the hyperplanes of the EG element of the decomposed DHA are defined. One reason for such high complexity is that when a hybrid system is formulated in the decomposed DHA, the information associated with the structure of the hybrid system becomes hidden and cannot be exploited using the translation techniques. However, in automaton-based DHA [23], this structural information is explicitly available from the graphical net of the DHA and can be efficiently utilized to solve DHA to PWA translation problem in a fast and low-complexity manner without any need for feasibility tests and applying mixed integer programming.

Although obtaining PWA representation from decomposed DHA and MLD systems has received much attention in the literature, to the best of our knowledge, only few works [49,50] have addressed the problem of equivalency relations and translation techniques between PWA systems and *automaton-based* DHA. In [49], the equivalency between discrete-time PWA systems and the set of linear systems combined with the finite automaton was represented. However, the notions of the controlled and uncontrolled switching phenomena, the role of reset dynamics, DHA trajectories, and conditions under which such translations are valid were not discussed. In [50], the relationship between an autonomous continuous-time Linear Hybrid Automaton (LHA) and PWA systems with disturbance inputs was examined. Modeling of the uncertainty associated with LHA transitions was performed by considering input disturbances in a PWA model. However, the role of discrete inputs and controlled switching phenomena has not been addressed.

1.2. Objectives and contributions

Motivated by the issues mentioned above, the goal of this paper is to solve the translation problem of an automaton-based DHA into its equivalent PWA form with an efficient and low-complexity manner. In the existing literature, this problem is solved using traditional decomposed DHA [35] with exponential complexity through the numerical solution of the cell enumeration problem and mixed integer programming techniques [38,39,41]. This problem is solved easier by using an automaton-based DHA with polynomial complexity and in an analytical manner rather than traditional numerical techniques. In this regard, first, a general discrete-time modeling framework of hybrid systems called automaton-based DHA is formally defined and represented. This modeling framework is established based on the continuous-time hybrid automata in the literature [33,51-55] with some modifications in their modeling structures. The inputs, outputs, and controlled switching phenomena were not considered in the proposed continuous-time hybrid automaton models in [33,51–54]. The proposed model in this paper is a generalization of the HA model in [33,51–54], in which inputs, outputs, and controlled switching phenomena are considered, as well. In some aspects, the proposed model in the present work is related to the existing continuous-time version of hybrid automata with the inputs and outputs in [55]. However, in our work, the transition guards are divided into the controlled and uncontrolled types and the related notions are defined in more detail. Another difference is that in [55], the discrete inputs affect both switching between submodels and continuous dynamics; however, in our work, discrete inputs affect only the switching between modes rather than the continuous dynamics. As compared to our earlier work in [23], the proposed model in the present work is a somewhat modified and generalized as to be aligned well to the system theoretical discussions on the DHA well-posedness and equivalence relations. For instance, all regions, invariants, and guards are defined as not necessarily closed polyhedra considering both strict and non-strict inequalities. Two types of state evolutions namely the traditional backward evolution [35] and forward evolution [23] are defined for the proposed automaton-based DHA. Sufficient conditions are provided for the well-posedness of the proposed automaton-based DHA. The constructive conversion of the proposed automaton-based DHA into its equivalent PWA models and vice versa are formally represented based on two different types of state evolutions (backward and forward).

1.3. Organization of the paper

The remainder of the paper is organized as follows. In Section 2, three main classes of discrete-time hybrid systems are represented including the traditional decomposed DHA, the proposed automaton-based DHA, and PWA systems. The concepts of forward and backward evolutions for hybrid states are presented and the well-posedness properties of the DHA and PWA systems are investigated in this section. Section 3 describes the equivalency relation between automaton-based DHA and PWA systems. The concepts of Forward and Backward PWA (FPWA and BPWA) systems corresponding to forward and backward evolutions of the automaton-based DHA are represented and efficient algorithms for the translation of a DHA into its equivalent FPWA and BPWA systems and vice versa are represented. The effectiveness of the proposed translation techniques is applied to two examples in Section 4. Finally, concluding remarks are made in Section 5.

Notation

$\mathbb{R}$, $\mathbb{Z}_{\geq 0}$ and $\mathbb{N}$ are used to denote the set of real, nonnegative integer and positive integer numbers, respectively. We use $\{0,1\}^n$ and $\{0,1\}^{m\times n}$ to denote the set of n-dimensional column vectors and $m \times n$ matrices whose elements are 0 or 1, respectively. $\mathbb{R}^k$ and $\mathbb{R}^{m\times n}$ denote real-valued k-dimensional column vectors and $m \times n$ matrices, respectively. We use I_n and $0_{m\times n}$ to denote the $n \times n$ identity matrix and the $m \times n$ zero matrix respectively. If $x \in \mathbb{R}^k$ is a vector, then x_i is the ith element of x. Equalities for real vectors must be understood componentwise. We use $\forall$ and $\exists$ to denote "for all" and "there exists", respectively. For real vectors $x, y \in \mathbb{R}^k$, $x \neq y$ if and only if $\exists i \in \{1, 2, ..., k\}$ such that $x_i \neq y_i$. Given a set A, $P(A)$ is the power set of A, i.e., the set of all subsets of A. Let φ be a collection of sets A_i where $i \in \{1, \ldots, N\}$. The general union of the sets in this collection is defined as: $\bigcup_{i=1}^{N} A_i = A_1 \cup A_2 \cup \ldots \cup A_N$. We distinguish a *function* or *mapping* $f(\cdot) : A \to B$ from a *set-valued mapping* $g(\cdot) : A \to P(B)$ by the condition that each $a \in A$ is related to a unique element $b \in B$ by $f(\cdot)$. In contrast, a set-valued mapping $g(\cdot)$ associates for each $a \in A$ a subset $g(a)$ of B. In logical expressions, $\wedge$ and $\vee$ are used to denote the logical "AND", "OR", respectively.

2. Discrete-time models of hybrid systems

There are different approaches to the development of hybrid modeling frameworks [12]:

- One modeling approach is to employ the existing discrete-event systems modeling frameworks such as automata and Petri Nets, and the existing continuous dynamic modeling tools such as difference equations as they are, and then couple them in a decomposed structure by appropriate interfaces (discrete-to-continuous and continuous-to-discrete). The traditional decomposed DHA proposed in [35] falls into this category of the modeling style.
- The next option is to start from existing discrete modeling frameworks of the discrete-event systems such as automata, Petri Nets and extend them by the injection of the continuous dynamics to each discrete state. In this modeling approach, discrete transitions occur based on the invariants of the discrete states and the guards between discrete modes, and reset dynamics are implemented during transition between discrete states. The automaton-based DHA in [23] falls into this category of modeling technique.

Contrary to the decomposed DHA, in automaton-based DHA, similar to continuous-time hybrid automata, the continuous dynamic is a property of the state of the automaton; however, in the decomposed DHA, the continuous dynamic is not a property of the state of the automaton and is selected by an interface element called MS according to discrete inputs, states, and events [35]. The choice of these modeling approaches depends on the analysis and synthesis problem at hand. In the sequel, more detailed analysis is provided on the structure and relations between these two different modeling frameworks.

2.1. The DHA with a decomposed structure

As shown in Figure 1, the DHA proposed in [35] decomposes the hybrid system into two interacting sub-

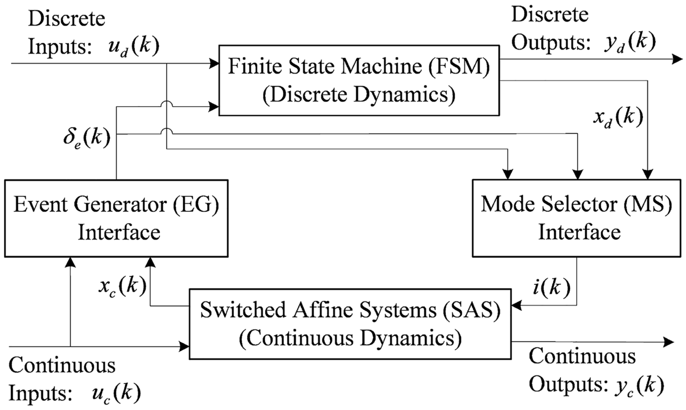

Figure 1. Discrete-time Hybrid Automata (DHA) with a decomposed structure.

systems: (a) a continuous system with the continuous input $u_c(k) \in U_c \subseteq \mathbb{R}^{m_c}$, the output $y_c(k) \in Y_c \subseteq \mathbb{R}^{p_c}$ and the state $x_c(k) \in X_c \subseteq \mathbb{R}^{n_c}$ signals; and (b) a discrete system with discrete inputs $u_d(k) \in U_d \subseteq \{0,1\}^{m_d}$, outputs $y_d(k) \in Y_d \subseteq \{0,1\}^{p_d}$, and states $x_d(k) \in X_d \subseteq \{0,1\}^{n_d}$. These continuous and discrete systems are called Switched Affine System (SAS) and FSM, respectively. The interaction between these two subsystems is realized by two interfaces, called as EG and MS, respectively. The former maps continuous-valued signals into discrete-valued signals $\delta_e(k)$; the latter uniquely translates discrete-valued signals into a discrete-valued signal $i(k)$ that selects a mode of SAS for continuous-state evolution.

In the sequel, each of these components is defined. SAS is a collection of affine systems:

$$x_c(k+1) = A_{i(k)}x_c(k) + B_{i(k)}u_c(k) + f_{i(k)},$$

$$y_c(k) = C_{i(k)}x_c(k) + D_{i(k)}u_c(k) + g_{i(k)}, \tag{1}$$

where $k \in \mathbb{Z}_{\geq 0}$ is the time indicator, $\{A_i, B_i, f_i, C_i, D_i, g_i\}_{i\in\mathcal{I}}$ is a set of matrices of suitable dimensions and $i(k) \in \mathcal{I} \triangleq \{1,\ldots,s\}$ is an input signal that chooses the affine state update dynamics. The FSM (or automaton) is described by the following discrete state-update functions:

$$x_d(k+1) = f_D(x_d(k), u_d(k), \delta_e(k)),$$

$$y_d(k) = g_D(x_d(k), u_d(k), \delta_e(k)), \tag{2}$$

where $f_D(.,.,.) : X_d \times U_d \times \mathcal{D} \rightarrow X_d$ and $g_D(.,.,.)$:$X_d \times U_d \times \mathcal{D} \rightarrow Y_d$ are deterministic discrete functions and $\mathcal{D} \subseteq \{0,1\}^{n_e}$. The EG generates a binary event signal $\delta_e(k)$ according to the satisfaction of the affine constraints or thresholds:

$$\delta_e(k) = f_H(x_c(k), u_c(k)), \tag{3}$$

where $f_H(.,.) : X_c \times U_c \rightarrow \mathcal{D}$ is a vector of descriptive functions of a linear hyperplane. The MS interface is described by:

$$i(k) = f_M(x_d(k), u_d(k), \delta_e(k)), \tag{4}$$

where $f_M(.,.,.) : X_d \times U_d \times \mathcal{D} \rightarrow \mathcal{I}$ is a deterministic discrete function. A mode switch occurs at the time instant k if $i(k-1) \neq i(k)$. In accordance with a mode switch $i(k) = j$, $i(k-1) = h \neq j$, $h, j \in \mathcal{D}$, instead of evolving $x_c(k+1) = A_j x_c(k) + B_j u_c(k) + f_j$, it is possible to associate a reset of the continuous state vector $x_c(k+1) = A_{hj}x_c(k) + B_{hj}u_c(k) + f_{hj}$. Such an state evolution type in the decomposed DHA in [35] is closely related to the backward evolution of the automaton-based DHA in [23]. On the other hand, in the decomposed DHA, there is no counterpart for the reset dynamics of the self-loop edges in the automaton-based DHA. From this point of view, the modeling power and expressiveness of the automaton-based DHA is greater than that of the decomposed DHA. Moreover, the graphical representation of the automaton-based DHA facilitates an easier communication with the model to determine the solution to some of the problems such as the translation of the model into its equivalent PWA system. Nevertheless, for the decomposed DHA, a tool called HYSDEL has been developed that provides a convenient textual representation as an input to the tool and allows describing the hybrid dynamics in a textual form. The HYSDEL compiler then translates this form into the corresponding PWA or MLD models [36], thus providing a multi-modeling capability. Moreover, the decomposition of hybrid systems into continuous and discrete subsystems points to the hybrid nature of the system explicitly and makes it possible to use the methods available for continuous and discrete systems to separate subsystems, although the obtained results for each isolated subsystem are not valid for the overall hybrid system [12]. In Subsection 3.2, we return to the issue of the decomposed DHA model where complexity analysis is to be performed on the translation techniques from the decomposed and automaton-based DHAs to PWA systems.

2.2. Automaton-based realization of a DHA

In [23], a general modeling framework called automaton-based DHA was proposed on the basis of the extended discrete event systems.

Definition 1. *A discrete-time hybrid automaton DHA is a collection:*

$$\begin{aligned} DHA = (&X_c, X_d, Y_c, Y_d, U_c, U_d, f_{x_c}(\cdot,\cdot,\cdot), f_{y_c}(\cdot,\cdot,\cdot),\\ &f_{y_d}(\cdot,\cdot), Init, Inv(\cdot), E_c, E_{uc}, G_c(\cdot), G_{uc}(\cdot),\\ &R_{x_c}(\cdot,\cdot,\cdot)), \end{aligned} \tag{5}$$

where:

- $X_c \subseteq \mathbb{R}^{n_c}$ *is a set of admissible continuous states;*
- $X_d = \{x_{d_1}, x_{d_2}, \ldots, x_{d_{N_d}}\} \subseteq \{0,1\}^{n_d^E}$ *is a finite set of discrete states where* $N_d \in \mathbb{N}$ *is the number of discrete states of the DHA;*
- $Y_c \subseteq \mathbb{R}^{p_c}$ *is a set of admissible continuous outputs;*
- $Y_d \subseteq \{0,1\}^{p_d}$ *is a set of admissible discrete outputs;*
- $U_c \subseteq \mathbb{R}^{m_c}$ *is a set of admissible continuous inputs;*
- $U_d \subseteq \{0,1\}^{m_d}$ *is the set of admissible discrete inputs to activate controlled switching events;*
- $E_c \subseteq X_d \times X_d$ *is a set of controlled or non-autonomous switching events;*
- $E_{uc} \subseteq X_d \times X_d$ *is a set of uncontrolled or autonomous switching events;*
- $f_{x_c}(\cdot,\cdot,\cdot) : X_c \times U_c \times X_d \rightarrow X_c$ *is a function*

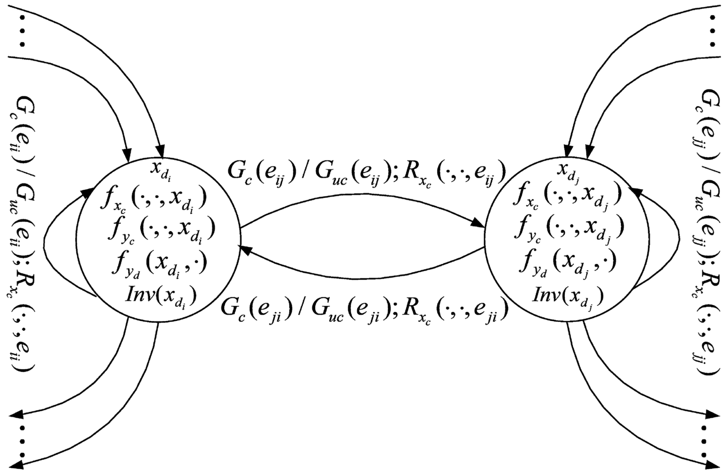

Figure 2. Automaton associated with a Discrete-time Hybrid Automata (DHA).

that determines the evolution of the continuous state $x_c(k)$ in each discrete state;

- $f_{y_c}(\cdot,\cdot,\cdot) : X_c \times U_c \times X_d \to Y_c$ *is a function which determines the evolution of the continuous output $y_c(k)$ in each discrete state;*
- $f_{y_d}(\cdot,\cdot) : X_d \times U_d \to Y_d$ *is a function that specifies the discrete output $y_d(k)$ evolution corresponding to each discrete state;*
- $Init \subseteq X_c \times X_d$ *is a set of initial states;*
- $Inv(\cdot) : X_d \to P(X_c \times U_c \times U_d)$ *is a set-valued function that describes the invariants or domains of the DHA, i.e., the valid continuous states, inputs and discrete inputs associated with discrete states of the DHA;*
- $G_c(\cdot) : E_c \to P(U_d)$ *is a controlled guard, i.e., a condition for controlled switching events;*
- $G_{uc}(\cdot) : E_{uc} \to P(X_c \times U_c)$ *is an uncontrolled guard, i.e., a condition for uncontrolled switching events;*
- $R_{x_c}(\cdot,\cdot,\cdot) : X_c \times U_c \times E \to X_c$ *is a reset map for continuous state $x_c(k)$ where $E = E_c \cup E_{uc}$.*

The proposed DHA can be represented as a directed graph shown in Figure 2.

Each discrete state of the DHA is shown with a *node* or a vertex in the graph, while the *edges* represent possible transitions between the discrete states. A discrete-time affine dynamical system and an optional discrete output are assigned to each node as follows:

$$x_c(k+1) = f_{x_c}(x_c(k), u_c(k), x_{d_i}) = A_{c_i}x_c(k) + B_{c_i}u_c(k) + h_{c_i}, \tag{6}$$

$$y_c(k) = f_{y_c}(x_c(k), u_c(k), x_{d_i}) = C_{c_i}x_c(k) + D_{c_i}u_c(k) + g_{c_i}, \tag{7}$$

$$y_d(k) = f_{y_d}(x_{d_i}, u_d(k)) = C_{d_i}x_{d_i} + D_{d_i}u_d(k) + g_{d_i}, \tag{8}$$

where $i \in \{1, 2, \ldots, N_d\}$, $A_{c_i} \in \mathbb{R}^{n_c \times n_c}$, $B_{c_i} \in \mathbb{R}^{n_c \times m_c}$, $h_{c_i} \in \mathbb{R}^{n_c}$, $C_{c_i} \in \mathbb{R}^{p_c \times n_c}$, $D_{c_i} \in \mathbb{R}^{p_c \times m_c}$, $g_{c_i} \in \mathbb{R}^{p_c}$, $C_{d_i} \in \mathbb{R}^{p_d \times n_d^E}$, $D_{d_i} \in \mathbb{R}^{p_d \times m_d}$, $g_{d_i} \in \mathbb{R}^{p_d}$. $k \in \mathbb{Z}_{\geq 0}$ is the discrete-time indicator, $x_c(k) \in X_c$ the continuous state vector, $u_c(k) \in U_c$ the exogenous continuous input vector, $u_d(k) \in U_d$ the exogenous discrete input vector, $y_d(k) \in Y_d$ the discrete output vector, and $y_c(k) \in Y_c$ the continuous output vector.

To each discrete state $x_d(k) = x_{d_i} \in X_d$, an invariant set $Inv(x_{d_i})$ is assigned as:

$$Inv(x_{d_i}) = \{(x_c(k), u_c(k), u_d(k)) \in X_c \times U_c \times U_d | H_i x_c(k) + J_i u_c(k) \leq K_i, \tilde{H}_i x_c(k) + \tilde{J}_i u_c(k) < \tilde{K}_i \wedge u_d(k) = u_{d_i}\}, \tag{9}$$

where H_i, J_i, K_i, $\tilde{H}_i$, $\tilde{J}_i$, and $\tilde{K}_i$ are real-valued matrices with suitable dimensions and $i \in \{1, \ldots, N_d\}$.

Each edge $e_{ij} = (x_{d_i}, x_{d_j}) \in E$ is labeled by an appropriate *guard* or switching condition for controlled or uncontrolled transitions. The uncontrolled switching condition is a polyhedral partition in the continuous state-input space as:

$$G_{uc}(e_{ij}) = \{(x_c(k), u_c(k)) \in X_c \times U_c | H_{ij}x_c(k) + J_{ij}u_c(k) \leq K_{ij}, \tilde{H}_{ij}x_c(k) + \tilde{J}_{ij}u_c(k) < \tilde{K}_{ij}\}, \tag{10}$$

where H_{ij}, J_{ij}, K_{ij}, $\tilde{H}_{ij}$, $\tilde{J}_{ij}$, and $\tilde{K}_{ij}$ are real-valued matrices with suitable dimensions and that $\forall i, j \in \{1, \ldots, N_d\}$; then, we have $e_{ij} \in E_{uc}$. The controlled switching condition is specified by the values of the discrete input as:

$$G_c(e_{ij}) = \left\{u_d(k) \in U_d | u_d(k) = u_{d_{ij}}\right\}. \tag{11}$$

To each controlled and uncontrolled edge $e_{ij} = (x_{d_i}, x_{d_j})$ of the graph, a reset function is assigned to update the continuous state during switching between the subsystems as follows:

$$x_c(k+1) = R_{x_c}(x_c(k), u_c(k), e_{ij}) = A_{c_{ij}} x_c(k) + B_{c_{ij}} u_c(k) + h_{c_{ij}}, \tag{12}$$

where $A_{c_{ij}} \in \mathbb{R}^{n_c \times n_c}$, $B_{c_{ij}} \in \mathbb{R}^{n_c \times m_c}$, $h_{c_{ij}} \in R^{n_c}$, $\forall i, j \in \{1, \ldots, N_d\}$ such that $e_{ij} \in E$.

Remark 1. *In Definition 1, in a DHA, each transition edge from discrete state x_{d_i} to x_{d_j} whose corresponding reset dynamic is equal to the assigned continuous dynamic in the incoming discrete state x_{d_j}, as in Eq. (13), is called a transition edge without a reset dynamic.*

$$A_{c_{ij}} x_c(k) + B_{c_{ij}} u_c(k) + h_{c_{ij}} = A_{c_j} x_c(k) + B_{c_j} u_c(k) + h_{c_j}. \tag{13}$$

The reverse statement also holds, i.e., for any transition edge without any specified reset dynamic, the equality condition Eq. (13) is valid.

Remark 2. *In Figure 2, the self-loop edges without reset dynamics are not shown in the directed graph of the DHA. Given that edge e_{ij} is a controlled or uncontrolled edge, the symbol $G_c(e_{ij})/G_{uc}(e_{ij})$ is used to denote the respective controlled or uncontrolled switching guard in the DHA graph. Accordingly, $u_d(k) \in G_c(e_{ij})/(x_c(k), u_c(k)) \in G_{uc}(e_{ij})$ and $u_d(k) \notin G_c(e_{ij})/(x_c(k), u_c(k)) \notin G_{uc}(e_{ij})$ are used to specify the activation and inactivation of the edge e_{ij} at instant k, respectively. In other words, when the continuous state and input satisfies a manifold characterized by the uncontrolled guard conditions, an uncontrolled edge is activated. Accordingly, when the discrete input satisfies a controlled guard condition, a controlled edge or event is activated. A controlled or uncontrolled edge is inactivated when it is not activated according to the preceding discussion. Obviously, since any edge in the DHA graph can only be of controlled or uncontrolled type from $u_d(k) \in G_c(e_{ij})/(x_c(k), u_c(k)) \in G_{uc}(e_{ij})$, we mean that only one of the conditions $u_d(k) \in G_c(e_{ij})$ or $(x_c(k), u_c(k)) \in G_{uc}(e_{ij})$ is satisfied depending on whether the respective edge is a controlled or uncontrolled edge, respectively.*

The evolution of the DHA state, input and output is defined based on Definition 2.

Definition 2. *An execution or run of a DHA over a discrete-time interval $K = [0, k_0] = 0, 1, \ldots, k_0$ where $k_0 \in \mathbb{Z}_{\geq 0}$ is a collection $(K, x_c(\cdot){:}K \to X_c, x_d(\cdot){:}K \to X_d, u_c(\cdot){:}K \to U_c, y_c(\cdot){:}K \to Y_c, y_d(\cdot){:}K \to Y_d)$ satisfying the following items:*

- *Initialization: Let us consider discrete state x_{d_i} of a DHA graph that may be connected to other nodes of the overall system through incoming and outgoing edges, as shown in Figure 3. It is supposed that at instant $k = 0$, $(x_c(0), x_{d_i}) \in Init$ where $i \in \{1, \ldots, N_d\}$.*
- *Discrete state evolution:*

  a) a transition from $x_d(k) = x_{d_i}$ to $x_d(k+1) = x_{d_j}$ where $(x_{d_i}, x_{d_j}) \in E_c \cup E_{uc}$ may occur at the instant $k+1$ if and only if $(x_c(k), u_c(k), u_d(k)) \in Inv(x_{d_i})$ and $(x_c(k), u_c(k)) \in G_{uc}(e_{ij})/u_d(k) \in$

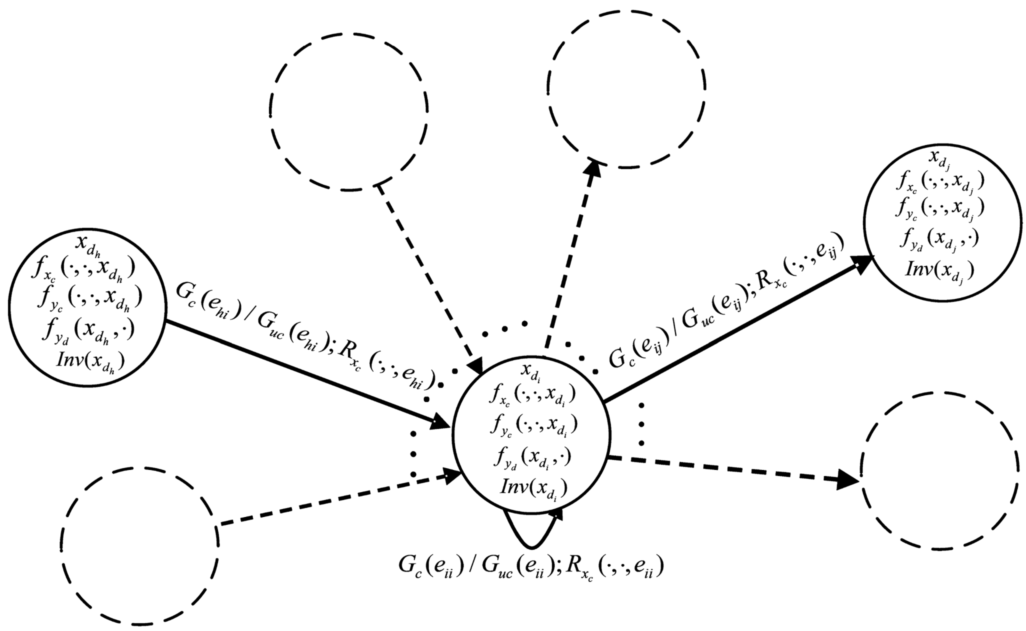

Figure 3. A node of a Discrete-time Hybrid Automata (DHA) with incoming and outgoing edges.

$G_c(e_{ij})$ (see Remark 2). In other words, there is a choice between further staying in node i or a discrete transition.

b) a transition from $x_d(k) = x_{d_i}$ to $x_d(k+1) = x_{d_j}$ where $(x_{d_i}, x_{d_j}) \in E_c \cup E_{uc}$ must occur at the instant $k+1$ if and only if $(x_c(k), u_c(k), u_d(k)) \notin Inv(x_{d_i})$ and $(x_c(k), u_c(k)) \in G_{uc}(e_{ij})/u_d(k) \in G_c(e_{ij})$ (see Remark 2).

- *Continuous state evolution: The evolution of the continuous state $x_c(k+1)$ can be described using two different techniques: backward or forward. These names are chosen to show the order of discrete states, in terms of the time through which the continuous dynamics are selected. At each step time k and in the backward evolution, $x_d(k)$ and $x_d(k-1)$ is used while $x_d(k)$ and $x_d(k+1)$ are employed in the forward evolution. A detailed discussion is presented by Definitions 3 and 4.*

Definition 3. *In a backward evolution, at each sample time k, the selection of the continuous dynamic is made based on the present and previous values of the discrete state, i.e., $x_d(k)$ and $x_d(k-1)$ and the activation of the switching guards of the self-loop edges, i.e., $G_c(e_{ii})/G_{uc}(e_{ii})$. Given $(x_c(k), x_{d_i}) \in X_c \times X_d$ and $u(k) = [u_c(k)^T, u_d(k)^T]^T$ where $(u_c(k), u_d(k)) \in U_c \times U_d$ the continuous state $x_c(k+1)$, $k \in \mathbb{Z}_{\geq 0}$ of the DHA in the backward evolution is computed as follows. Assuming $x_d(-1) = x_d(0)$, compare $x_d(k-1) = x_{d_h}$ with $x_d(k) = x_{d_i}$. If $x_d(k-1) = x_d(k)$, $(x_c(k), u_c(k), u_d(k)) \in Inv(x_{d_i})$ and $u_d(k) \notin G_c(e_{ii})/(x_c(k), u_c(k)) \notin G_{uc}(e_{ii})$, i.e., e_{ii} is inactivated, then compute $x_c(k+1)$ by the continuous dynamic associated to the node i, namely:*

$$x_c(k+1) = A_{c_i} x_c(k) + B_{c_i} u_c(k) + h_{c_i},$$

else if $x_d(k-1) = x_d(k)$ and $u_d(k) \in G_c(e_{ii})/(x_c(k), u_c(k)) \in G_{uc}(e_{ii})$, i.e., e_{ii} is activated, then compute $x_c(k+1)$ through the reset dynamic associated to the edge e_{ii}, i.e.:

$$x_c(k+1) = A_{c_{ii}} x_c(k) + B_{c_{ii}} u_c(k) + h_{c_{ii}},$$

else if $x_d(k-1) \neq x_d(k)$ then compute $x_c(k+1)$ through the reset dynamic associated to the edge e_{hi}, namely:

$$x_c(k+1) = A_{c_{hi}} x_c(k) + B_{c_{hi}} u_c(k) + h_{c_{hi}}$$

Remark 3. *If all edges incoming to the discrete state x_{d_i} are without a reset dynamic (see Remark 1), then $x_{d_h} = x_{d_i}$ and $x_{d_h} \neq x_{d_i}$ in Definition 3 become superfluous and only $x_d(k) = x_{d_i}$ decides for $x_c(k+1)$, namely, $x_c(k+1) = A_{c_i} x_c(k) + B_{c_i} u_c(k) + h_{c_i}$. This type of backward evolution has been used in some pieces of literature such as [56–59].*

Definition 4. *In a forward evolution, selection of the continuous dynamic at each sample time is made based on the current and the next values of the discrete state, i.e., $x_d(k)$ and $x_d(k+1)$ and also the activation of the switching guards of the self-loop edges, i.e., $G_c(e_{ii})$ or $G_{uc}(e_{ii})$. Note that $x_d(k+1) = x_{d_j}$ is available at instant k according to the occurrence of $e_{ij} = (x_{d_i}, x_{d_j}) \in E_c \cup E_{uc}$ (see Definition 2, discrete state evolution item). The continuous state $x_c(k+1)$ is determined by using the following algorithm:*

Compare $x_d(k) = x_{d_i}$ with $x_d(k+1) = x_{d_j}$:
If $x_d(k) = x_d(k+1)$, and $u_d(k) \notin G_c(e_{ii})/(x_c(k), u_c(k)) \notin G_{uc}(e_{ii})$, i.e., e_{ii} is inactivated, then compute $x_c(k+1)$ by using the continuous dynamic associated to the node i, namely:

$$x_c(k+1) = A_{c_i} x_c(k) + B_{c_i} u_c(k) + h_{c_i},$$

else if $x_d(k) = x_d(k+1)$ and $u_d(k) \in G_c(e_{ii})/(x_c(k), u_c(k)) \in G_{uc}(e_{ii})$, i.e., e_{ii} is activated, then compute $x_c(k+1)$ via the reset dynamic associated to the self-loop edge e_{ii}, i.e.:

$$x_c(k+1) = A_{c_{ii}} x_c(k) + B_{c_{ii}} u_c(k) + h_{c_{ii}},$$

else if $x_d(k) \neq x_d(k+1)$, then compute $x_c(k+1)$ through the reset dynamic associated to the edge e_{ij}, namely:

$$x_c(k+1) = A_{c_{ij}} x_c(k) + B_{c_{ij}} u_c(k) + h_{c_{ij}}$$

Remark 4. *If all edges outgoing from the discrete state x_{d_i} are without a reset dynamic (based on Remark 1), then $x_{d_i} = x_{d_j}$ and $x_{d_i} \neq x_{d_j}$ in Definition 4 become redundant and only $x_d(k+1) = x_{d_j}$ decides for $x_c(k+1)$, namely $x_c(k+1) = A_{c_j} x_c(k) + B_{c_j} u_c(k) + h_{c_j}$. This type of forward evolution has been used in some works of the literature such as [60,61].*

- *Output evolution: the values of the continuous and discrete outputs are computed by Eqs. (7) and (8).*

It should be noted that the DHA model in Definition 1 is not a single model. In reality, it represents two different types of models that depend on the type of evolutions (forward or backward). This issue is much highlighted when we want to define the hybrid state $x(k)$ for these two different systems. The state of a system is loosely defined as a set of variables such that knowledge of them at some time together with the future inputs is sufficient to allow for the determination of the system future behavior.

Definition 5. *In the DHA with backward evolution, according to Definition 3, knowledge $x_d(k)$, $x_c(k)$, and $u(k)$ is not enough to determine $x_c(k+1)$ and, $x_d(k-1)$ needs to be known. As a result, the state of the DHA is defined as $x(k) = [x_c(k)^T, x_d(k)^T, x_d(k-1)^T]^T$. On the other hand, based on Definition 4, the state of* *the DHA with forward evolution is defined as* $x(k) = [x_c(k)^T, x_d(k)^T]^T$.

Rmark 5. *It should be noted that remembering the previous discrete state* $x_d(k-1)$ *in the DHA with backward evolution is only required when the DHA contains reset dynamics. According to Remark 3, when the DHA is without reset dynamics, the knowledge of* $x_d(k-1)$ *is not required to determine the future behavior of the system. In this case the augmentation of the DHA state via* $x_d(k-1)$ *in backward evolution is not required and the DHA state can be constantly defined as* $x(k) = [x_c(k)^T, x_d(k)^T]^T$ *for both forward and backward evolutions.*

$E_{DHA}(x_c(0), x_d(0))$ is used to denote the set of all executions of a DHA with the initial condition $(x_c(0), x_d(0)) \in Init$. E_{DHA} is used to denote the union of $E_{DHA}(x_c(0), x_d(0))$ over all $(x_c(0), x_d(0)) \in Init$. The set $Reach_{DHA}$ in Eq. (14) is defined as the set of all reachable states and outputs with their corresponding inputs:

$$\begin{aligned} Reach_{DHA} = \{(&\tilde{x_c}(k), \tilde{x_d}(k), \tilde{u_c}(k), \tilde{u_d}(k), \tilde{y_c}(k), \\ &\tilde{y_d}(k)) \in X_c \times X_d \times U_c \times U_d \times Y_c \times Y_d | \\ &\exists K = [0, k_0] \text{ with } k_0 \in \mathbb{Z}_{\geq 0} \\ &\textit{such that } \ \exists(K, x_c(\cdot), x_d(\cdot), u_c(\cdot), \\ &u_d(\cdot), y_c(\cdot), y_d(\cdot)) \in E_{DHA} \\ &\textit{such that } \ (x_c(k_0), x_d(k_0), u_c(k_0), \\ &u_d(k_0), y_c(k_0), y_d(k_0)) = (\tilde{x_c}(k), \tilde{x_d}(k), \\ &\tilde{u_c}(k), \tilde{u_d}(k), \tilde{y_c}(k), \tilde{y_d}(k))\}. \end{aligned} \tag{14}$$

Remark 6. *In particular cases in which the uncontrolled switching conditions in Eq. (10) are independent of the continuous inputs, one can use a one-step predictive reset. In this situation, the switching between the discrete states is made one sampling step earlier. In other words, a switching occurs before the related guard or switching boundary is actually crossed. An uncontrolled guard between nodes i and j of the DHA is crossed at instant k if* $(x_c(k), u_c(k)) \in G_{uc}(e_{ij})$. *The switching condition is a polyhedral partition in the continuous state space as follows:*

$$G_{uc}(e_{ij}) = \left\{ x_c^{p1}(k) \in X_c | H_{ij} x_c^{p1}(k) \leq K_{ij}, \tilde{H}_{ij} x_c^{p1}(k) < \tilde{K}_{ij} \right\}, \tag{15}$$

where $x_c^{p1}(k) = x_c(k+1) = A_{c_i} x_c(k) + B_{c_i} u_c(k) + h_{c_i}$ *is the one step predicted value for the continuous state at instant* k. *If the continuous dynamic at node i is independent of the continuous inputs, in these conditions i.e.,* $x_c(k+1) = A_{c_i} x_c(k) + h_{c_i}$, *then the idea of one-step predictive reset can be extended to a multi-step predictive reset. In this case, the switching condition can be written as follows:*

$$G_{uc}(e_{ij}) = \left\{ x_c^{pN}(k) \in X_c | H_{ij} x_c^{pN}(k) \leq K_{ij}, \tilde{H}_{ij} x_c^{pN}(k) < \tilde{K}_{ij} \right\}, \tag{16}$$

where $x_c^{pN}(k) = x_c(k+N) = A_{c_i}^N x_c(k) + (\sum_{l=0}^{N-1} A_{c_i}^l) h_{c_i}$ *is* N*-step predicted value for the continuous state at instant* k *and is obtained by recursive implementation of* $x_c(k+1) = A_{c_i} x_c(k) + h_{c_i}$.

Remark 7. *Based on the preceding discussions, the combination of the predictive reset mentioned in Remark 6 and the backward evolution in Definition 3 is called as backward predictive reset. On the other hand, the forward predictive reset is the combination of the predictive reset in Remark 6 and the forward evolution in Definition 4. In cases in which the multi-step predictive reset is impossible in a DHA, the forward predictive reset has the advantage of one-step prediction more with respect to the backward predictive reset, hence better modeling power. This is because in backward evolutions, there is a one-step delay between the change of the continuous dynamics and the discrete states.*

Sometimes, as it occurs during modeling of DC-DC converters in both continuous and discontinuous conduction modes, due to the discrete-time nature of DHA modeling framework and the inability of exact detecting of the uncontrolled switching surfaces because of technical limitation in selecting small discretization sample times, it is necessary to use predictive reset dynamics during the transition between different modes. This should be done to avoid the state variables such as inductor current to take unrealistic negative values and obtain the most exact discrete-time models. See [23,35,62–64] for more details and numerical examples.

Definition 5 represents the concept of the well-posedness property of a hybrid system model which is of particular importance for its usability.

Definition 5. *A DHA is well-posed on* $X_c \times X_d$, $U_c \times U_d$, $Y_c \times Y_d$, *if* $\forall x(0) = [x_c(0)^T, x_d(0)^T]^T$, $(x_c(0), x_d(0)) \in Init$, *and* $\forall k \in \mathbb{Z}_{\geq 0}$, *and* $\forall u(k) = [u_c(k)^T, u_d(k)^T]^T$, $(u_c(k), u_d(k)) \in U_c \times U_d$, *the state trajectory* $x(k) = [x_c(k)^T, x_d(k)^T]^T$, $(x_c(k), x_d(k)) \in X_c \times X_d$ *and the output trajectory* $y(k) = [y_c(k)^T, y_d(k)^T]^T$, $(y_c(k), y_d(k)) \in Y_c \times Y_d$, *are uniquely defined.*

It is noted that the state notation in Definition 5 is for the forward evolution. In backward evolution, according to Definition 5, one can use the augmented vectors $x(0) = [x_c(0)^T, x_d(0)^T, x_d(-1)^T]^T$ and $x(k) = [x_c(k)^T, x_d(k)^T, x_d(k-1)^T]^T$ where $x_d(-1)^T = x_d(0)^T$ (see Definition 3).

Theorem 1 states under which conditions the DHA is well-posed, and therefore, evolves with unique state and output trajectories over the infinite time horizon for all initial conditions in $Init$. In this theorem, $CInv_i$, $DInv_i$, and C_i are defined as follows:

$$CInv_i = \{(x_c(k), u_c(k)) \in X_c \times U_c | (x_c(k), u_c(k), u_d(k)) \in Inv(x_{d_i})\} = \{(x_c(k), u_c(k)) \in X_c \times U_c | H_i x_c(k) + J_i u_c(k) \leq K_i, \tilde{H}_i x_c(k) + \tilde{J}_i u_c(k) < \tilde{K}_i\}, \quad (17)$$

$$DInv_i = \{u_d(k) \in U_d | (x_c(k), u_c(k), u_d(k)) \in Inv(x_{d_i})\} = \{u_d(k) \in U_d | u_d(k) = u_{d_i}\}, \quad (18)$$

$$C_i = \{(x_c(k), u_c(k), u_d(k)) \in X_c \times U_c \times U_d | u_d(k) \in \bigcup_j G_c(e_{ij}) \wedge (x_c(k), u_c(k)) \in \bigcup_l G_{uc}(e_{il}), j \neq l\}. \quad (19)$$

$\forall i, j, l \in \{1, \ldots, N_d\}$ such that $e_{ij} \in E_c$ and $e_{il} \in E_{uc}$. The sets $CInv_i$ and $DInv_i$ determine the continuous and discrete parts of $Inv(x_{d_i})$, respectively. The set C_i determines all the values of $(x_c(k), u_c(k), u_d(k))$ for which at least one of the controlled and uncontrolled outgoing edges of the node i to different discrete state successors is simultaneously enabled. Furthermore, the sets $CInv_i^c$, $DInv_i^c$ are the complementary sets of $CInv_i$ and $DInv_i$, respectively. The set $Reach_{DHA}^{X_c U_c U_d}$ is defined as follows:

$$Reach_{DHA}^{X_c U_c U_d} = \{(\tilde{x_c}(k), \tilde{u_c}(k), \tilde{u_d}(k)) \in X_c \times U_c \times U_d | \exists \tilde{x_d}(k) \in X_d, \exists \tilde{y_c}(k) \in Y_c, \exists \tilde{y_d}(k) \in Y_d \ \ such\ that \ \ (\tilde{x_c}(k), \tilde{x_d}(k), \tilde{u_c}(k), \tilde{u_d}(k), \tilde{y_c}(k), \tilde{y_d}(k)) \in Reach_{DHA}\}.$$

In fact, $Reach_{DHA}^{X_c U_c U_d}$ is the projection of the set $Reach_{DHA}$, defined in Eq. (14), onto $X_c \times U_c \times U_d$.

Theorem 1. *The DHA of Definition 1 is well-posed if the following conditions are satisfied.*

(a) $X_c = \mathbb{R}^{n_c}$, $Y_c = \mathbb{R}^{p_c}$,

(b) $C_{d_i} = 0_{p_d \times n_d^E}$, $D_{d_i} = 0_{p_d \times m_d}$, $g_{d_i} \in Y_d$,

(c) $CInv_i^c = \bigcup_j G_{uc}(e_{ij}), \forall i, j \in \{1, \ldots, N_d\}, e_{ij} \in E_{uc}$,

(d) $DInv_i^c = \bigcup_j G_c(e_{ij}), \forall i, j \in \{1, \ldots, N_d\}, e_{ij} \in E_c$,

(e) $G_{uc}(e_{ij}) \cap G_{uc}(e_{ik}) = \emptyset, \forall i, j, k \in \{1, \ldots, N_d\}$ *such that* $j \neq k$, *and* $e_{ij}, e_{ik} \in E_{uc}$,

(f) $G_c(e_{ij}) \cap G_c(e_{ik}) = \emptyset, \forall i, j, k \in \{1, \ldots, N_d\}$ *such that* $j \neq k$ *and* $e_{ij}, e_{ik} \in E_c$,

(g) $C_i = \emptyset, \quad \forall i \in \{1, \ldots, N_d\}$.

Proof. Condition (a) provides an invariant property for both continuous state $x_c(k)$ and continuous output $y_c(k)$ and implies that $x_c(k) \in X_c$ and $y_c(k) \in Y_c$, $\forall k \in \mathbb{Z}_{\geq 0}$, when $x_c(0) \in X_c$ and $y_c(0) \in Y_c$. Condition (b) guarantees that the discrete output update $y_d(k)$ is a piecewise binary function belonging to its corresponding set Y_d over infinite time horizon.

Let us consider an initial condition $(x_c(0), x_d(0)) \in Init$ such that $x_d(0) = x_{d_i}$. The continuous state $x_c(k)$, continuous and discrete outputs, $y_c(k)$ and $y_d(k)$ of the DHA are extended uniquely by Eqs. (6)–(8) to $(x_c(k), u_c(k), u_d(k)) \in Inv(x_{d_i})^c \cap Reach_{DHA}^{X_c U_c U_d}$ at instant k. This implies that $(x_c(k), u_c(k), u_d(k)) \notin Inv(x_{d_i})$. Therefore, one can conclude that $(x_c(k), u_c(k)) \notin CInv_i$ or $u_d(k) \notin DInv_i$. Equivalently, this implies that $(x_c(k), u_c(k)) \in CInv_i^c$ or $u_d(k) \in DInv_i^c$. In Case 1, if $(x_c(k), u_c(k)) \in CInv_i^c$, then, one can conclude according to Condition (c) that $(x_c(k), u_c(k)) \in \bigcup_j G_{uc}(e_{ij})$. This and Condition (e) imply that definitely only one of the uncontrolled outgoing edges $e_{ij} \in E_{uc}$ is enabled, and the discrete state of the DHA is changed from x_{d_i} to x_{d_j}. Therefore, the continuous state trajectory of the system $x_c(k)$ is extended uniquely by the reset dynamic in Eq. (12) for forward evolution or by Eq. (6) in backward evolution. In both types of evolutions, $y_c(k)$ and $y_d(k)$ evolve uniquely via Eqs. (7) and (8). In Case 2, if $u_d(k) \in DInv_i^c$, then one can conclude according to (d) that $u_d(k) \in \bigcup_j G_c(e_{ij})$. This and Condition (f) imply only one of the outgoing controlled edges $e_{ij} \in E_c$ is enabled, and the discrete state is changed from x_{d_i} to x_{d_j} and $x_c(k)$ is evolved by the reset dynamic in Eq. (12) in the forward evolution or by Eq. (6) in the backward evolution uniquely. Again, $y_c(k)$ and $y_d(k)$ change uniquely via Eqs. (7) and (8). Case 3 takes place when both $(x_c(k), u_c(k)) \in CInv_i^c$ and $u_d(k) \in DInv_i^c$ are simultaneously valid. However, Condition (g) prevents from occurring of this case. This can be shown by a contradiction: Let $(x_c(k), u_c(k)) \in CInv_i^c$ and $u_d(k) \in DInv_i^c$. Then, according to (a) and (b), one can conclude that $(x_c(k), u_c(k)) \in \bigcup_j G_{uc}(e_{ij})$ and $u_d(k) \in \bigcup_j G_c(e_{ij})$. This means that there is a possibility of simultaneous controlled and uncontrolled transition from node i to a different discrete state successor. Equivalently, this implies that $\exists(x_c(k), u_c(k), u_d(k)) \in C_i$. In other words, $C_i \neq \emptyset$ and this contradicts Condition (g). Therefore, whenever the continuous evolution is impossible in a node of the DHA, say node i, a discrete transition to a new node, say node j, is possible and $x_c(k)$ is modified uniquely by the reset dynamics in Eq. (12) associated with the edge e_{ij} in forward evolution, or by the affine dynamic of Eq. (6) associated to the node i of the DHA graph in backward evolution. Besides, $y_c(k)$ and $y_d(k)$ change uniquely via Eqs. (7) and (8). Consequently, the DHAis well posed.□

Note that the reverse statement does not hold, since a well-posed DHA may generally be defined on non-satisfying conditions (a) - (g) of Theorem 1. Moreover, Conditions (a) - (g) are not so restrictive and are satisfied when real plants are described in this modeling framework. This statement is confirmed in other related works such as [22] when the well-posedness property has been defined for the MLD systems. Since the DHA has been proven to be equivalent to MLD systems [23,35], one can conclude that the proposed well-posedness conditions in Theorem 1 are less conservative and are applicable for real plants as well.

2.3. Piecewise affine (PWA) systems

The state-space representation of discrete-time PWA systems is described by [35,36,39,43,65]:

$$\begin{cases} x(k+1) = \tilde{A}_i x(k) + \tilde{B}_i u(k) + \tilde{f}_i \\ y(k) = \tilde{C}_i x(k) + \tilde{D}_i u(k) + \tilde{g}_i \end{cases}$$

$$\text{for} \quad (x(k), u(k)) \in \tilde{\Omega}_i \tag{20}$$

where $x(k) = [x_c(k)^T, x_d(k)^T]^T$ is the state, $u(k) = [u_c(k)^T, u_d(k)^T]^T$ the input, and $y(k) = [y_c(k)^T, y_d(k)^T]^T$ is the output. The state, input and output are partitioned in the continuous components $x_c(k) \in X_c \subseteq \mathbb{R}^{n_c}$, $u_c(k) \in U_c \subseteq \mathbb{R}^{m_c}$, $y_c(k) \in Y_c \subseteq \mathbb{R}^{p_c}$ and discrete components $x_d(k) \in X_d \subseteq \{0,1\}^{n_d}$, $u_d(k) \in U_d \subseteq \{0,1\}^{m_d}$, $y_d(k) \in Y_d \subseteq \{0,1\}^{p_d}$. Each affine subsystem described by the 6-tuple $(\tilde{A}_I, \tilde{B}_i, \tilde{f}_i, \tilde{C}_i, \tilde{D}_i, \tilde{g}_i)$, $i = 1, \cdots, \tilde{s}$ is defined on a cell $\tilde{\Omega}_i \subset X_c \times X_d \times U_c \times U_d$, $i = 1, \cdots, \tilde{s}$ which is a (not necessarily closed) polyhedron on the state-input space defined by a system of inequalities as:

$$\tilde{\Omega}_i = \{(x(k), u(k)) | \tilde{H}_{x,i} x(k) + \tilde{H}_{u,i} u(k) \leq \tilde{k}_i, \tilde{\tilde{H}}_{x,i} x(k) + \tilde{\tilde{H}}_{u,i} u(k) < \tilde{\tilde{k}}_i\}, \tag{21}$$

where $\tilde{H}_{x,i}$, $\tilde{H}_{u,i}$, $\tilde{k}_i$, $\tilde{\tilde{H}}_{x,i}$, $\tilde{\tilde{H}}_{u,i}$, and $\tilde{\tilde{k}}_i$ are matrices with suitable dimensions. Moreover, $\tilde{A}_i, \tilde{B}_i, \tilde{f}_i, \tilde{C}_i, \tilde{D}_i$, and $\tilde{g}_i$ are real matrices of suitable dimensions that define the affine dynamics for all i's. For PWA systems in Eq. (20), the concept of well posedness is defined as in Definition 5. A sufficient condition for the PWA system (Eq. (20)) to be well-posed is given by Lemma 1.

Lemma 1. *The PWA system in Eq. (20) is well-posed on $\Omega \triangleq X_c \times X_d \times U_c \times U_d$ if the following conditions are satisfied:*

(a) *rows and columns of matrices $\tilde{A}_i, \tilde{B}_i, \tilde{C}_i$, and $\tilde{D}_i$ corresponding to the discrete (binary) states and outputs are zero such that:*

$$\tilde{A}_i = \begin{bmatrix} \tilde{A}_{c_i} & \tilde{\tilde{A}}_{c_i} \\ 0_{n_d \times n_c} & 0_{n_d \times n_d} \end{bmatrix},$$

$$\tilde{B}_i = \begin{bmatrix} \tilde{B}_{c_i} & \tilde{\tilde{B}}_{c_i} \\ 0_{n_d \times m_c} & 0_{n_d \times m_d} \end{bmatrix},$$

$$\tilde{f}_i = \begin{bmatrix} \tilde{f}_{c_i} \\ \tilde{f}_{d_i} \end{bmatrix},$$

$$\tilde{C}_i = \begin{bmatrix} \tilde{C}_{c_i} & \tilde{\tilde{C}}_{c_i} \\ 0_{p_d \times n_c} & 0_{p_d \times n_d} \end{bmatrix},$$

$$\tilde{D}_i = \begin{bmatrix} \tilde{D}_{c_i} & \tilde{\tilde{D}}_{c_i} \\ 0_{p_d \times m_c} & 0_{p_d \times m_d} \end{bmatrix},$$

$$\tilde{g}_i = \begin{bmatrix} \tilde{g}_{c_i} \\ \tilde{g}_{d_i} \end{bmatrix}, \tag{22}$$

where $\tilde{f}_{d_i} \in X_d$, $\tilde{g}_{d_i} \in Y_d$, $\tilde{A}_{c_i} \in \mathbb{R}^{n_c \times n_c}$, $\tilde{\tilde{A}}_{c_i} \in \mathbb{R}^{n_c \times n_d}$, $\tilde{B}_{c_i} \in \mathbb{R}^{n_c \times m_c}$, $\tilde{\tilde{B}}_{c_i} \in \mathbb{R}^{n_c \times m_d}$, $\tilde{f}_{c_i} \in \mathbb{R}^{n_c}$, $\tilde{C}_{c_i} \in \mathbb{R}^{p_c \times n_c}$, $\tilde{\tilde{C}}_{c_i} \in \mathbb{R}^{p_c \times n_d}$, $\tilde{g}_{c_i} \in \mathbb{R}^{p_c}$, $\tilde{C}_{c_i} \in \mathbb{R}^{p_c \times n_c}$, $\tilde{\tilde{C}}_{c_i} \in \mathbb{R}^{p_c \times n_d}$, $\tilde{g}_{c_i} \in \mathbb{R}^{p_c}$, $\tilde{D}_{c_i} \in \mathbb{R}^{p_c \times m_c}$, $\tilde{\tilde{D}}_{c_i} \in \mathbb{R}^{p_c \times m_d}$.

(b) $X_c = \mathbb{R}^{n_c}$, $Y_c = \mathbb{R}^{p_c}$.

(c) $\tilde{\Omega}_i \cap \tilde{\Omega}_j = \emptyset$, $\forall i, j \in \{1, \cdots, \tilde{s}\}$ *and* $i \neq j$.

(d) $\tilde{\Omega}_i^c = \bigcup_{j=1, j\neq i}^{j=\tilde{s}} \tilde{\Omega}_j$, $\forall i \in \{1, \cdots, \tilde{s}\}$.

Proof. Condition (a) implies that the discrete (binary) state and output trajectories are always binary piecewise constant functions. In other words, over infinite time horizon, the discrete state $x_d(k)$ and output $y_d(k)$ remain in their discrete sets X_d and Y_d, respectively. Condition (b) provides the invariance property for continuous state $x_c(k)$ and output $y_c(k)$; it implies that over infinite time horizon, the continuous state and output will remain in their sets if they start from them. Condition (c) implies that as long as $(x(k), u(k)) \in \tilde{\Omega}_i$, the state and output trajectory $[x(k)^T, y(k)^T]^T$ of the PWA system is extended uniquely by the corresponding affine subsystem $(\tilde{A}_i, \tilde{B}_i, \tilde{f}_i, \tilde{C}_i, \tilde{D}_i, \tilde{g}_i)$ in Eq. (20). Condition (d) guarantees the continuous and discrete state evolutions over infinite time horizon and avoids existing regions in the state space that do not belong to any region $\tilde{\Omega}_i$. In other words, the union of the sets $\tilde{\Omega}_i$ spans the whole state space, i.e., $\Omega \triangleq \bigcup_{i=1}^{\tilde{s}} \tilde{\Omega}_i = X_c \times X_d \times U_c \times U_d$ where $X_c = \mathbb{R}^{n_c}$ according to Condition (b). Therefore, through the satisfaction of the Conditions (a)–(d), in all cases, the state and output trajectory of the PWA system in Eq. (20) is uniquely defined $\forall k \in \mathbb{Z}_{\geq 0}$, and as a result, it is well-posed.

Similar to Theorem 1, regarding the well-posedness of the DHA, the given conditions in Lemma 1 for the well-posedness of the PWA systems are not necessary. In other words, a well-posed PWA system may be defined while Conditions (a)–(d) are not satisfied. It is noted that in the *continuous-time* domain, there are some pieces of the literature that propose necessary and sufficient conditions for the well-posedness property of PWA and Piecewise Linear (PWL) systems [66,67]. However, in these works some of the general aspects of the PWA systems in Eq. (20) are ignored, of which one can mention the elimination of the discrete states and their corresponding dynamics, continuous and discrete control inputs, controlled switching phenomena, and continuous and discrete output signals.

The following class of hybrid systems called PWA system in a logic canonical form (PWA-LC system) has a fundamental role in developing Lagrange stability and building on performance analysis based on linear matrix inequalities [43,68]. Therefore, in terms of system analysis, it is of particular importance to represent a hybrid system model in the form of PWA-LC modeling framework.

Definition 6. *A PWA system in the logic canonical form (PWA-LC system) is described by the state-space equations:*

$$\begin{cases} x(k+1) = A_i x(k) + B_i u(k) + f_i \\ y(k) = C_i x(k) + D_i u(k) + g_i \end{cases}$$

$$\text{for} \quad (x(k), u(k)) \in \Omega_i \tag{23}$$

where:
$i = 1, \ldots, s,$

$$\Omega_i = \{(x(k), u(k)) | H_{x,i} x_c(k) + H_{u,i} u_c(k) \leq k_i, \bar{H}_{x,i} x_c(k) + \bar{H}_{u,i} u_c(k) < \bar{k}_i, u_d(k) = u_{d_i}, x_d(k) = x_{d_i}\}, \tag{24}$$

$$A_i = \begin{bmatrix} A_{c_i} & 0_{n_c \times n_d} \\ 0_{n_d \times n_c} & 0_{n_d \times n_d} \end{bmatrix},$$

$$B_i = \begin{bmatrix} B_{c_i} & 0_{n_c \times m_d} \\ 0_{n_d \times m_c} & 0_{n_d \times m_d} \end{bmatrix},$$

$$f_i = \begin{bmatrix} f_{c_i} \\ f_{d_i} \end{bmatrix},$$

$$C_i = \begin{bmatrix} C_{c_i} & 0_{p_c \times n_d} \\ 0_{p_d \times n_c} & 0_{p_d \times n_d} \end{bmatrix},$$

$$D_i = \begin{bmatrix} D_{c_i} & 0_{p_c \times m_d} \\ 0_{p_d \times m_c} & 0_{p_d \times m_d} \end{bmatrix},$$

$$g_i = \begin{bmatrix} g_{c_i} \\ g_{d_i} \end{bmatrix}, \tag{25}$$

$f_{d_i} \in \{0,1\}^{n_d}$, $g_{d_i} \in \{0,1\}^{p_d}$, $A_{c_i} \in \mathbb{R}^{n_c \times n_c}$, $B_{c_i} \in \mathbb{R}^{n_c \times m_c}$, $f_{c_i} \in \mathbb{R}^{n_c}$, $C_{c_i} \in \mathbb{R}^{p_c \times n_c}$, $D_{c_i} \in \mathbb{R}^{p_c \times m_c}$, $g_{c_i} \in \mathbb{R}^{p_c}$.

Similar to Lemma 1, a set of sufficient conditions can be developed for well-posedness of PWA-LC systems via Lemma 2.

Lemma 2. *The PWA-LC system in Eq. (23) is well-posed on* $\Omega \triangleq X_c \times X_d \times U_c \times U_d$ *if the following conditions are satisfied:*

(a) $f_{d_i} \in X_d$, $g_{d_i} \in Y_d$,

(b) $X_c = \mathbb{R}^{n_c}$, $Y_c = \mathbb{R}^{p_c}$,

(c) $\Omega_i \cap \Omega_j = \emptyset$ *for all* $i, j \in \{1, \cdots, s\}$ *and* $i \neq j$,

(d) $\Omega_i^c = \bigcup_{j=1, j \neq i}^{j=s} \Omega_j$, $\forall i \in \{1, \cdots, s\}$.

Proof. The proof of this Lemma is similar to that of Lemma 1 and is omitted for the sake of brevity.

Compared to PWA systems in Eq. (20), the PWA-LC systems have two additional properties. First, according to Eqs. (23) and (25), one can see that the dynamics of continuous-valued variables $x_c(k)$ and $y_c(k)$ are not influenced by the discrete states $x_d(k)$ and inputs $u_d(k)$. In fact, the discrete input $u_d(k)$ contributes only to the switching between different subsystems. The second feature is that when index i is fixed, according to Eq. (24), $x_d(k)$ and $u_d(k)$ are constant within a cell Ω_i, and $x_d(k+1)$ and $y_d(k)$ are determined by only f_{d_i} and g_{d_i}.

In the next section, we show how an automaton-based DHA can be translated into its equivalent PWA and PWA-LC forms. In the PWA-LC system, the discrete variables influence the switching between sub-models rather than the continuous dynamics. This is an important feature of the PWA-LC form because it provides a useful framework to investigate the properties of the continuous-valued signals for general PWA systems [43,68].

3. Automaton-based DHA and PWA systems

This section discusses the relationship of the proposed discrete time hybrid automaton in Definition 1 with the class of PWA and PWA-LC systems presented in Subsection 2.3.

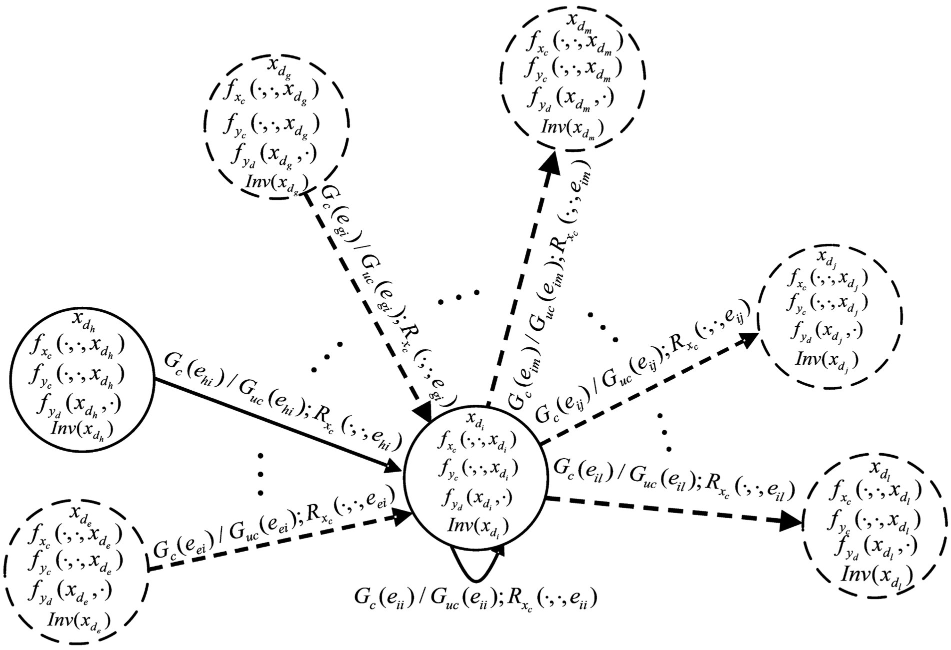

Figure 4. Concerned part of a Discrete-time Hybrid Automata (DHA) net to extract Backward PWA (BPWA) and Forward PWA (FPWA) models.

Definition 7. *Let Σ_1 and Σ_2 be two well-posed hybrid models by inputs $u_1(k) \in U_1$ and $u_2(k) \in U_2$, outputs $y_1(k) \in Y_1$ and $y_2(k) \in Y_2$, and states $x_1(k) \in X_1$ and $x_2(k) \in X_2$, $k \in Z_{\geq 0}$. The hybrid models Σ_1 and Σ_2 are equivalent on $\bar{X} = X_1 \cap X_2$, $\bar{U} = U_1 \cap U_2$ and $\bar{Y} = Y_1 \cap Y_2$ if for all initial conditions $x_1(0) = x_2(0) \in \bar{X}$ and for all inputs $u_1(k) = u_2(k) \in \bar{U}$, the state and output trajectories coincide, i.e., $x_1(k) = x_2(k)$ and $y_1(k) = y_2(k)$ for all discrete time steps $k \in Z_{\geq 0}$.*

Definition 8. *The PWA systems obtained from the translation of a DHA with the backward or forward evolution are called backward or forward PWA (BPWA or FPWA) systems, respectively. The mathematical representation of BPWA and FPWA systems is similar to that of the general PWA systems in Eq. (20).*

The names FPWA and BPWA reflect the evolution type (forward or backward) under which the DHA model of Definition 1 is translated into its equivalent PWA system.

Definition 9. *The PWA-LC systems obtained from the translation of a DHA with the backward or forward evolution are called backward or forward PWA-LC (BPWA-LC or FPWA-LC) systems, respectively. The mathematical representations of BPWA-LC and FPWA-LC systems are similar to those of the general PWA-LC systems in Eq. (23).*

Lemma 3 represents a constructive approach to convert a well-posed automaton-based DHA into its equivalent PWA and PWA-LC forms. It also plays a key role to evaluate the complexity of the proposed translation technique in Proposition 1 and Subsection 3.2.

Lemma 3. *A well-posed DHA in Definition 1 with defined backward and forward evolutions in Definitions 3 and 4 can be transformed into the equivalent well-posed BPWA and FPWA systems and also into the BPWA-LC and FPWA-LC systems.*

Proof. Let us consider discrete state x_{d_i} of a DHA graph that may be connected to other nodes of the overall system through in-coming and out going edges, as shown in Figure 4.

For the brevity of the formulations, the terms in the BPWA and FPWA forms are only specified for the continuous state dynamics associated with the discrete state x_{d_i}, the single incoming and self-loop edges at this node are depicted by the solid lines in Figure 4. The overall BPWA and FPWA systems can be obtained by the union of the similar terms for all discrete states and edges in the DHA.

Case 1: BPWA and BPWA-LC formulation. The BPWA form in Eq. (26) is obtained through the backward evolution in Definition 3. The BPWA form in Eq. (26) shown in Box I is not in the standard format of Eq. (20). However, it can be easily transformed to its standard form by the definition of a new augmented discrete state vector $x_d^{new}(k)$ as in Eq. (27).

$$x_d^{new}(k) = \begin{bmatrix} x_d(k) \\ x_d(k-1) \end{bmatrix}. \tag{27}$$

$$\begin{bmatrix} x(k+1) \\ y(k) \end{bmatrix} = \begin{bmatrix} x_c(k+1) \\ x_d(k+1) \\ y_c(k) \\ y_d(k) \end{bmatrix} =$$

$$= \begin{cases} \vdots & \vdots \\ \begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ x_{d_i} \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d(k)=x_{d_i} \wedge x_d(k-1)=x_{d_i} \wedge Inv(x_{d_i}) \\ \begin{bmatrix} A_{c_{ii}}x_c(k)+B_{c_{ii}}u_c(k)+f_{c_{ii}} \\ x_{d_i} \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d(k)=x_{d_i} \wedge x_d(k-1)=x_{d_i} \wedge G_c(e_{ii})/G_{uc}(e_{ii}) \\ \begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ x_{d_l} \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d(k)=x_{d_i} \wedge x_d(k-1)=x_{d_i} \wedge G_c(e_{il})/G_{uc}(e_{il}) \\ \vdots & \vdots \\ \begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ x_{d_m} \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d(k)=x_{d_i} \wedge x_d(k-1)=x_{d_i} \wedge G_c(e_{im})/G_{uc}(e_{im}) \\ \begin{bmatrix} A_{c_{hi}}x_c(k)+B_{c_{hi}}u_c(k)+f_{c_{hi}} \\ x_{d_i} \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d(k)=x_{d_i} \wedge x_d(k-1)=x_{d_h} \wedge Inv(x_{d_i}) \cup G_c(e_{ii})/G_{uc}(e_{ii}) \\ \begin{bmatrix} A_{c_{hi}}x_c(k)+B_{c_{hi}}u_c(k)+f_{c_{hi}} \\ x_{d_l} \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d(k)=x_{d_i} \wedge x_d(k-1)=x_{d_h} \wedge G_c(e_{il})/G_{uc}(e_{il}) \\ \vdots & \vdots \\ \begin{bmatrix} A_{c_{hi}}x_c(k)+B_{c_{hi}}u_c(k)+f_{c_{hi}} \\ x_{d_m} \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d(k)=x_{d_i} \wedge x_d(k-1)=x_{d_h} \wedge G_c(e_{im})/G_{uc}(e_{im}) \\ \vdots & \vdots \end{cases} \tag{26}$$

Box I

Using Eq. (27), Eq. (26) can be rewritten as Eq. (28), shown in Box II. According to Eq. (28), all the continuous and discrete dynamics associated with each cell can be realized in the standard continuous and discrete dynamics of the PWA form in Eq. (20) or PWA-LC form described in Eqs. (23) and (25). Moreover, except one of the regions described by $x_d^{new}(k) = [x_{d_i}^T \ x_{d_h}^T]^T \wedge Inv(x_{d_i}) \cup G_c(e_{ii})/G_{uc}(e_{ii})$, all the regions are in the standard region representation of the PWA form in Eq. (21) and PWA-LC systems in Eq. (24). However, this region can be divided into two standard PWA or PWA-LC regions as $x_d^{new}(k) = [x_{d_i}^T \ x_{d_h}^T]^T \wedge Inv(x_{d_i})$

$$\begin{bmatrix} x(k+1) \\ y(k) \end{bmatrix} = \begin{bmatrix} x_c(k+1) \\ x_d^{new}(k+1) \\ y_c(k) \\ y_d(k) \end{bmatrix} = \tag{28}$$

$$\begin{cases} \vdots & \vdots \\ \begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ [x_{d_i}^T\ x_{d_i}^T]^T \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d^{new}(k)=[x_{d_i}^T\ x_{d_i}^T]^T \wedge Inv(x_{d_i}) \\ \begin{bmatrix} A_{c_{ii}}x_c(k)+B_{c_{ii}}u_c(k)+f_{c_{ii}} \\ [x_{d_i}^T\ x_{d_i}^T]^T \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d^{new}(k)=[x_{d_i}^T\ x_{d_i}^T]^T \wedge G_c(e_{ii})/G_{uc}(e_{ii}) \\ \begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ [x_{d_l}^T\ x_{d_i}^T]^T \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d^{new}(k)=[x_{d_i}^T\ x_{d_i}^T]^T \wedge G_c(e_{il})/G_{uc}(e_{il}) \\ \vdots & \vdots \\ \begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ [x_{d_m}^T\ x_{d_i}^T]^T \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d^{new}(k)=[x_{d_i}^T\ x_{d_i}^T]^T \wedge G_c(e_{im})/G_{uc}(e_{im}) \\ \begin{bmatrix} A_{c_{hi}}x_c(k)+B_{c_{hi}}u_c(k)+f_{c_{hi}} \\ [x_{d_i}^T\ x_{d_i}^T]^T \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d^{new}(k)=[x_{d_i}^T\ x_{d_h}^T]^T \wedge Inv(x_{d_i}) \cup G_c(e_{ii})/G_{uc}(e_{ii}) \\ \begin{bmatrix} A_{c_{hi}}x_c(k)+B_{c_{hi}}u_c(k)+f_{c_{hi}} \\ [x_{d_l}^T\ x_{d_i}^T]^T \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d^{new}(k)=[x_{d_i}^T\ x_{d_h}^T]^T \wedge G_c(e_{il})/G_{uc}(e_{il}) \\ \vdots & \vdots \\ \begin{bmatrix} A_{c_{hi}}x_c(k)+B_{c_{hi}}u_c(k)+f_{c_{hi}} \\ [x_{d_m}^T\ x_{d_i}^T]^T \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d^{new}(k)=[x_{d_i}^T\ x_{d_h}^T]^T \wedge G_c(e_{im})/G_{uc}(e_{im}) \\ \vdots & \vdots \end{cases}$$

Box II

and $x_d^{new}(k)=[x_{d_i}^T\ x_{d_h}^T]^T \wedge G_c(e_{ii})/G_{uc}(e_{ii})$ that share the same continuous and discrete dynamics. As a result, the BPWA or BPWA-LC form of a DHA with the backward evolution can be obtained from Eq. (28) as in Eq. (29) shown in Box III.

Case 2: FPWA and FPWA-LC formulations: The FPWA representation is obtained from the forward evolution of the DHA (see Definition 4). The specified FPWA form in Eq. (30), shown in Box IV, corresponds to the part of the DHA graph which is depicted by the solid lines in Figure 4. According to Eq. (30), one can see that the FPWA system is automatically in the form of standard PWA-LC system described

$$
\begin{bmatrix} x(k+1) \\ y(k) \end{bmatrix} = \begin{bmatrix} x_c(k+1) \\ x_d^{new}(k+1) \\ y_c(k) \\ y_d(k) \end{bmatrix} =
\begin{cases}
\vdots & \vdots \\
\begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ [x_{d_i}^T \; x_{d_i}^T]^T \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d^{new}(k)=[x_{d_i}^T \; x_{d_i}^T]^T \wedge Inv(x_{d_i}) \\
\begin{bmatrix} A_{c_{ii}}x_c(k)+B_{c_{ii}}u_c(k)+f_{c_{ii}} \\ [x_{d_i}^T \; x_{d_i}^T]^T \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d^{new}(k)=[x_{d_i}^T \; x_{d_i}^T]^T \wedge G_c(e_{ii})/G_{uc}(e_{ii}) \\
\begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ [x_{d_l}^T \; x_{d_i}^T]^T \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d^{new}(k)=[x_{d_i}^T \; x_{d_i}^T]^T \wedge G_c(e_{il})/G_{uc}(e_{il}) \\
\vdots & \vdots \\
\begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ [x_{d_m}^T \; x_{d_i}^T]^T \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d^{new}(k)=[x_{d_i}^T \; x_{d_i}^T]^T \wedge G_c(e_{im})/G_{uc}(e_{im}) \\
\begin{bmatrix} A_{c_{hi}}x_c(k)+B_{c_{hi}}u_c(k)+f_{c_{hi}} \\ [x_{d_i}^T \; x_{d_i}^T]^T \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d^{new}(k)=[x_{d_i}^T \; x_{d_h}^T]^T \wedge Inv(x_{d_i}) \\
\begin{bmatrix} A_{c_{hi}}x_c(k)+B_{c_{hi}}u_c(k)+f_{c_{hi}} \\ [x_{d_i}^T \; x_{d_i}^T]^T \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d^{new}(k)=[x_{d_i}^T \; x_{d_h}^T]^T \wedge G_c(e_{ii})/G_{uc}(e_{ii}) \\
\begin{bmatrix} A_{c_{hi}}x_c(k)+B_{c_{hi}}u_c(k)+f_{c_{hi}} \\ [x_{d_l}^T \; x_{d_i}^T]^T \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d^{new}(k)=[x_{d_i}^T \; x_{d_h}^T]^T \wedge G_c(e_{il})/G_{uc}(e_{il}) \\
\vdots & \vdots \\
\begin{bmatrix} A_{c_{hi}}x_c(k)+B_{c_{hi}}u_c(k)+f_{c_{hi}} \\ [x_{d_m}^T \; x_{d_i}^T]^T \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d^{new}(k)=[x_{d_i}^T \; x_{d_h}^T]^T \wedge G_c(e_{im})/G_{uc}(e_{im}) \\
\vdots & \vdots
\end{cases}
\tag{29}
$$

Box III

$$\begin{bmatrix} x(k+1) \\ y(k) \end{bmatrix} = \begin{bmatrix} x_c(k+1) \\ x_d(k+1) \\ y_c(k) \\ y_d(k) \end{bmatrix} = \begin{cases} \vdots & \vdots \\ \begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ x_{d_i} \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d(k)=x_{d_i} \wedge Inv(x_{d_i}) \\ \begin{bmatrix} A_{c_{ii}}x_c(k)+B_{c_{ii}}u_c(k)+f_{c_{ii}} \\ x_{d_i} \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d(k)=x_{d_i} \wedge G_c(e_{ii})/G_{uc}(e_{ii}) \\ \begin{bmatrix} A_{c_{hi}}x_c(k)+B_{c_{hi}}u_c(k)+f_{c_{hi}} \\ x_{d_i} \\ C_{c_h}x_c(k)+D_{c_h}u_c(k)+g_{c_h} \\ C_{d_h}x_d(k)+D_{d_h}u_d(k)+g_{d_h} \end{bmatrix} & x_d(k)=x_{d_h} \wedge G_c(e_{hi})/G_{uc}(e_{hi}) \\ \vdots & \vdots \end{cases} \tag{30}$$

Box IV

by Eqs. (23)–(25). Since it is originated from the forward evolution of DHA, it is called FPWA-LC form according to Definition 9.

Remark 8. *In some cases, e.g., for the simulation purposes and not the analysis studies, it is required to use a more compact representation of the PWA systems in Eq. (20) with a smaller number of regions as far as possible. In such conditions, in a post processing operation, to reduce the number of polyhedral regions in the standard PWA or PWA-LC forms, we check all regions whose affine subsystems are the same and attempt to compute their union as* $\Omega_{cj} = \bigcup_{k=1}^{n_i} \tilde{\Omega}_k$ *or* $\Omega_{cj} = \bigcup_{k=1}^{n_i} \Omega_k$ *where* Ω_{cj} *is the* j^{th} *region in the new compact representation, and* n_i *is the number of regions in the standard PWA or PWA-LC forms that share the same affine subsystems. Since, in general, the union of the convex polyhedra is not a convex polyhedron and as a result, the new regions in the compact representation of these systems may not be represented as their standard forms in Eq. (21) [69], they cannot be called PWA forms. However, in this paper, for the sake of brevity and according to the fact that in a reverse process these compact regions can always be decomposed to the same standard regions in Eq. (21) or Eq. (24) sharing the same affine subsystems, we still call them PWA systems.*

Definition 10. Based on Remark 8, the regions of the compact forms of PWA or PWA-LC systems obtained from the union of the regions of the standard forms of these systems and do not share their affine subsystems with other regions are called independent regions.

According to Remark 8, in some special cases where the incoming edge $e_{hi} \in E$ in Figure 4 is without reset dynamic (see Remark 1), the BPWA form in Eq. (28) can be further simplified as in Eq. (31), shown in Box V (see Remark 8). Furthermore, if the discrete state x_{d_i} is without self-loop edge $e_{ii} \in E$ (see Remark 2), then the BPWA representation in Eq. (31) can be simplified to a greater degree as in Eq. (32) shown in Box VI. In contrast to the backward evolution case in Eq. (31), when the incoming edge $e_{hi} \in E$ in Figure 4 is without a reset dynamic (see Remark 1), the number of regions in Eq. (30) cannot be reduced further. This is because as it is presented in Eq. (33), shown in Box VII, although some regions share the same continuous and discrete state dynamics, the continuous and discrete output dynamics vary in these regions.

Remark 9. *In special cases where the proposed DHA is without continuous and discrete outputs given by Eqs. (7) and (8) or they are identical in all nodes of the DHA, in the case that the incoming edge* $e_{hi} \in E$ *in Figure 4 is without a reset dynamic (see Remark 1), the FPWA form in Eq. (30) can be simplified further as in Eq. (34) shown in Box VIII (see Remark 8).*

Furthermore, if the discrete state x_{d_i} is without a self-loop edge $e_{ii} \in E$ (see Remark 2), then the FPWA representation in Eq. (33) can be simplified more as in Eq. (35) shown in Box IX. As can be

$$\begin{bmatrix} x(k+1) \\ y(k) \end{bmatrix} = \begin{bmatrix} x_c(k+1) \\ x_d^{new}(k+1) \\ y_c(k) \\ y_d(k) \end{bmatrix} =$$

$$\begin{cases} \vdots & \vdots \\ \begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ [x_{d_i}^T \ x_{d_i}^T]^T \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & \begin{matrix} x_d^{new}(k)=[x_{d_i}^T \ x_{d_i}^T]^T \wedge Inv(x_{d_i}) \\ \cup(x_d^{new}(k)=[x_{d_i}^T \ x_{d_h}^T]^T \wedge [Inv(x_{d_i}) \cup G_c(e_{ii})/G_{uc}(e_{ii})]) \end{matrix} \\ \begin{bmatrix} A_{c_{ii}}x_c(k)+B_{c_{ii}}u_c(k)+f_{c_{ii}} \\ [x_{d_i}^T \ x_{d_i}^T]^T \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d^{new}(k)=[x_{d_i}^T \ x_{d_i}^T]^T \wedge G_c(e_{ii})/G_{uc}(e_{ii}) \\ \begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ [x_{d_l}^T \ x_{d_i}^T]^T \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & \begin{matrix} x_d^{new}(k)=[x_{d_i}^T \ x_{d_i}^T]^T \wedge G_c(e_{il})/G_{uc}(e_{il}) \\ \cup(x_d^{new}(k)=[x_{d_i}^T \ x_{d_h}^T]^T \wedge G_c(e_{il})/G_{uc}(e_{il})) \end{matrix} \\ \vdots & \vdots \\ \begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ [x_{d_m}^T \ x_{d_i}^T]^T \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & \begin{matrix} x_d^{new}(k)=[x_{d_i}^T \ x_{d_i}^T]^T \wedge G_c(e_{im})/G_{uc}(e_{im}) \\ \cup(x_d^{new}(k)=[x_{d_i}^T \ x_{d_h}^T]^T \wedge G_c(e_{im})/G_{uc}(e_{im})) \end{matrix} \\ \vdots & \vdots \end{cases} \tag{31}$$

Box V

seen from Eq. (26), when the edge e_{ij} is activated, the discrete state of the system is changed from x_{d_i} to x_{d_j}. However, the continuous state of the system still evolves through continuous dynamic in the discrete state x_{d_i} or the reset dynamic corresponding to the edge between x_{d_h} and x_{d_i}. In this modeling method, one step sampling delay occurs between the change of the discrete state and corresponding continuous dynamic during switching events. This type of PWA system is called Backward PWA (BPWA) system which is the result of the backward evolution of the original DHA. Another approach is that switching among continuous dynamics of the system is synchronized with the change of the system discrete state. As can be seen from Eq. (30), when a switching event occurs from discrete state x_{d_h} to discrete state x_{d_i} at the sampling step k, the continuous state of the system also evolves with continuous dynamic associated with the reset dynamic of the transition edge from x_{d_h} to x_{d_i}, i.e., $R_{x_{c_{hi}}}$. With this approach, the change of the discrete state is synchronized with the change of the corresponding continuous dynamics during a switching event. This type of modeling, which is the result of forward evolution of the DHA, yields an FPWA.

Despite one step delay compensation in the FPWA framework, comparison Eqs. (28)–(32) with Eqs. (30)–(35) shows that the number of regions in the BPWA form is always equal to or greater than that of the FPWA form. The following proposition is introduced on the number of regions in FPWA and BPWA forms of a given DHA. The following parameters are defined:

- $n_{inre}(i)$ is the number of incoming edges $e_{hi}, h \neq i$ in the discrete state x_{d_i} with reset dynamics;
- $n_{out}(i)$ is the number of out-going edges $e_{ij}, i \neq j$ (with or without reset) in the discrete state x_{d_i};
- $n_{in}(i)$ is the number of incoming edges $e_{hi}, h \neq i$ (with or without reset) in the discrete state x_{d_i};

$$
\begin{bmatrix} x(k+1) \\ y(k) \end{bmatrix} = \begin{bmatrix} x_c(k+1) \\ x_d^{new}(k+1) \\ y_c(k) \\ y_d(k) \end{bmatrix} =
$$

$$
\begin{cases}
\vdots & \vdots \\
\begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ [x_{d_i}^T \ x_{d_i}^T]^T \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & \begin{array}{c} x_d^{new}(k)=[x_{d_i}^T \ x_{d_i}^T]^T \wedge Inv(x_{d_i}) \\ \cup(x_d^{new}(k)=[x_{d_i}^T \ x_{d_h}^T]^T \wedge Inv(x_{d_i})) \end{array} \\
\begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ [x_{d_l}^T \ x_{d_i}^T]^T \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & \begin{array}{c} x_d^{new}(k)=[x_{d_i}^T \ x_{d_i}^T]^T \wedge G_c(e_{il})/G_{uc}(e_{il}) \\ \cup(x_d^{new}(k)=[x_{d_i}^T \ x_{d_h}^T]^T \wedge G_c(e_{il})/G_{uc}(e_{il})) \end{array} \\
\vdots & \vdots \\
\begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ [x_{d_m}^T \ x_{d_i}^T]^T \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & \begin{array}{c} x_d^{new}(k)=[x_{d_i}^T \ x_{d_i}^T]^T \wedge (G_c(e_{im})/G_{uc}(e_{im}) \\ \cup(x_d^{new}(k)=[x_{d_i}^T \ x_{d_h}^T]^T \wedge G_c(e_{im})/G_{uc}(e_{im})) \end{array} \\
\vdots & \vdots
\end{cases}
\tag{32}
$$

Box VI

$$
\begin{bmatrix} x(k+1) \\ y(k) \end{bmatrix} = \begin{bmatrix} x_c(k+1) \\ x_d(k+1) \\ y_c(k) \\ y_d(k) \end{bmatrix} = \begin{cases}
\vdots & \vdots \\
\begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ x_{d_i} \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d(k)=x_{d_i} \wedge Inv(x_{d_i}) \\
\begin{bmatrix} A_{c_{ii}}x_c(k)+B_{c_{ii}}u_c(k)+f_{c_{ii}} \\ x_{d_i} \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d(k)=x_{d_i} \wedge G_c(e_{ii})/G_{uc}(e_{ii}) \\
\begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ x_{d_i} \\ C_{c_h}x_c(k)+D_{c_h}u_c(k)+g_{c_h} \\ C_{d_h}x_d(k)+D_{d_h}u_d(k)+g_{d_h} \end{bmatrix} & x_d(k)=x_{d_h} \wedge G_c(e_{hi})/G_{uc}(e_{hi}) \\
\vdots & \vdots
\end{cases}
\tag{33}
$$

Box VII

- N_{self} is the total number of self-loop edges e_{ii} with reset dynamics;
- N_{inre} is the total number of transition edges $e_{ij}, i \neq j$ with reset dynamic;
- N_d is the total number of discrete states in DHA.
- N_t is the total number of transition edges $e_{ij}, i \neq j$ (with or without reset) in the DHA.

Proposition 1. *The number of regions in BPWA and*

$$\begin{bmatrix} x(k+1) \\ y(k) \end{bmatrix} = \begin{bmatrix} x_c(k+1) \\ x_d(k+1) \\ y_c(k) \\ y_d(k) \end{bmatrix} = \begin{cases} \vdots & \vdots \\ \begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ x_{d_i} \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & \begin{array}{c} x_d(k)=x_{d_i} \wedge Inv(x_{d_i}) \\ \cup(x_d(k)=x_{d_h} \wedge G_c(e_{hi})/G_{uc}(e_{hi})) \end{array} \\ \begin{bmatrix} A_{c_{ii}}x_c(k)+B_{c_{ii}}u_c(k)+f_{c_{ii}} \\ x_{d_i} \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d(k)=x_{d_i} \wedge G_c(e_{ii})/G_{uc}(e_{ii}) \\ \vdots & \vdots \end{cases} \tag{34}$$

Box VIII

$$\begin{bmatrix} x(k+1) \\ y(k) \end{bmatrix} = \begin{bmatrix} x_c(k+1) \\ x_d(k+1) \\ y_c(k) \\ y_d(k) \end{bmatrix} = \begin{cases} \vdots & \vdots \\ \begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ x_{d_i} \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d(k)=x_{d_i} \wedge Inv(x_{d_i}) \\ \begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ x_{d_i} \\ C_{c_h}x_c(k)+D_{c_h}u_c(k)+g_{c_h} \\ C_{d_h}x_d(k)+D_{d_h}u_d(k)+g_{d_h} \end{bmatrix} & x_d(k)=x_{d_h} \wedge G_c(e_{hi})/G_{uc}(e_{hi}) \\ \vdots & \vdots \end{cases} \tag{35}$$

Box IX

FPWA of a DHA is equal to $N_{R_{BPWA}}$ and $N_{R_{FPWA}}$ such that:

$$N_{R_{BPWA}} = N_d + N_{self} + 3N_t + \sum_{i=1}^{N_d} n_{out}(i)n_{in}(i), \tag{36}$$

$$N_{R_{FPWA}} = N_{self} + N_d + N_t. \tag{37}$$

Proof. According to Eq. (29), one can see that the number of regions in the BPWA form of a DHA characterized by the continuous dynamic in the discrete state x_{d_i}, i.e., $x_c(k+1) = A_{c_i}x_c(k) + B_{c_i}u_c(k) + f_{c_i}$ is $1 + n_{out}(i)$. Thus, the total number of such regions is $\sum_{i=1}^{N_d}(1 + n_{out}(i))$. The reset dynamic associated with each self-loop creates only one region. Therefore, the total number of regions characterized by all self-loops with reset dynamics of type $x_c(k+1) = A_{c_{ii}}x_c(k) + B_{c_{ii}}u_c(k) + f_{c_{ii}}$ is N_{self}. Also, the number of the regions identified by an incoming edge $e_{hi}, h \neq i$, with or without reset dynamics is equal to $2 + n_{out}(i)$. As a result, the total number of such regions is $\sum_{i=1}^{N_d}(2 + n_{out}(i))n_{in}(i)$. Therefore, the number of regions in a BPWA form of a DHA can be written as:

$$N_{R_{BPWA}} = \sum_{i=1}^{N_d}(1 + n_{out}(i)) + \sum_{i=1}^{N_d}(2 + n_{out}(i))n_{in}(i) + N_{self}. \tag{38}$$

Simplification of Eq. (38) yields:

$$N_{R_{BPWA}} = N_d + N_{self} + 3N_t + \sum_{i=1}^{N_d} n_{out}(i)n_{in}(i). \tag{39}$$

According to Eq. (30), one can see that each self-loop or incoming edge of the discrete state x_{d_i} with or without reset dynamic specifies a single region in the FPWA model. Furthermore, the continuous dynamic associated with each node of the DHA creates only one region. Therefore, the number of regions in the FPWA form of a DHA, i.e., $N_{R_{FPWA}}$, can be written as:

$$N_{R_{FPWA}} = N_{self} + \sum_{i=1}^{N_d} 1 + \sum_{i=1}^{N_t} 1. \tag{40}$$

Finally, Eq. (40) can be simplified as:

$$N_{R_{FPWA}} = N_{self} + N_d + N_t. \tag{41}$$

The regions whose numbers calculated in Proposition 1 are of standard type as represented in Eqs. (21) and (24). According to Remark 8, in a post-processing task, one can merge all the regions sharing the same affine subsystems and achieve more compact forms of BPWA and FPWA systems with fewer regions. Propositions 2 and 3 quantify the number of *independent* regions (see Definition 10) in these compact forms of BPWA and FPWA systems.

Proposition 2. *The number of independent regions in the compact representation of the BPWA equivalent model of a DHA is equal to* $N_{R_{BPWA}}$ *such that:*

$$N_{R_{BPWA}} = N_d + N_{self} + N_{inre} + N_t + \sum_{i=1}^{N_d} n_{out}(i) n_{inre}(i). \tag{42}$$

Proof. According to Eqs. (28), (31) and (32), one can see that the number of independent regions in the compact form of a BPWA model characterized by the continuous dynamic in the discrete state x_{d_i} in conjunction with the incoming edges without reset dynamics is $1 + n_{out}(i)$. The reset dynamic associated with each self-loop edge determines only one independent region. The number of independent regions identified by the reset dynamic associated with an incoming edge e_{hi}, $h \neq i$, i.e., $x_c(k+1) = A_{c_{hi}} x_c(k) + B_{c_{hi}} u_c(k) + f_{c_{hi}}$ is equal to $1 + n_{out}(i)$. Thus, the total number of such independent regions is $\sum_{i=1}^{N_d}(1 + n_{out}(i)) n_{inre}(i)$. As a result, the total number of independent regions in the compact form of a BPWA model can be obtained as:

$$N_{R_{BPWA}} = \sum_{i=1}^{N_d}(1 + n_{out}(i)) + \sum_{i=1}^{N_d}(1 + n_{out}(i)) n_{inre}(i) + N_{self}. \tag{43}$$

Simplification of Eq. (43) yields:

$$N_{R_{BPWA}} = N_d + N_{self} + N_{inre} + N_t + \sum_{i=1}^{N_d} n_{out}(i) n_{inre}(i). \tag{44}$$

From Propositions 1 and 2, one can conclude that the number of regions in BPWA is greater than that of the FPWA with a quantity $N_{inre} + \sum_{i=1}^{N_d}(1 + n_{out}(i)) n_{inre}(i)$, and as a result, FPWA is a more compact representation.

Corollary 1. *In the case that the physical plants do not have reset dynamics (based on Remark 1), the number of regions in the BPWA and FPWA systems is identical and equal to* $N_d + N_t$. *In these conditions, according to Remark 5, the augmentation of discrete state vector via* $x_d(k-1)$ *similar to what was done in the case of the equivalent BPWA form in Eq. (27) is not required.*

Reset dynamics or state jumps appear in various hybrid systems, e.g., the mechanical systems containing elastic collisions. Usually, the analysis of a system without jumps (reset dynamics) is easier than that of with jumps. On the other hand, there are also many practical examples of hybrid systems that do not present state jumps in transition from one mode to another. An example of such hybrid systems is the gear shift control system of a car or mechanical systems with inelastic collisions [70,71].

According to Proposition 3, when the conditions of Remark 9 hold, the number of regions in the FPWA form can be further reduced compared to that of Eq. (37) in Proposition 1.

Proposition 3. *Let the DHA of Definition 1 be without continuous and discrete outputs given by Eqs. (7) and (8), or these dynamics are identical in all discrete states of the DHA then, the number of regions of the FPWA model is determined by:*

$$N_{R_{FPWA}} = N_{self} + N_d + N_{inre}. \tag{45}$$

Proof. According to Remark 9 and Eq. (34), it can be observed that each self-loop or incoming edge of the discrete state x_{d_i} with reset dynamic specifies a single region in the FPWA model. Moreover, the continuous dynamic associated with each node of the DHA creates only one region. As a result, the number of regions in the FPWA model can be computed as:

$$N_{R_{FPWA}} = N_{self} + \sum_{i=1}^{N_d} 1 + \sum_{i=1}^{N_d} n_{inre}(i). \tag{46}$$

Simplifying Eq. (46), we have:

$$N_{R_{FPWA}} = N_{self} + N_d + N_{inre}. \tag{47}$$

Corollary 2. *If the conditions of Proposition 3 hold and the DHA is without reset dynamics, then the number of regions in the FPWA form is reduced to* N_d.

For a given PWA system in the standard form of Eq. (20), one can detect whether it is evolved by a backward evolution or forward evolution. According to Propositions 1–3, each continuous dynamic associated with each node of the equivalent DHA appears more than once in the BPWA form (the precise number is $n_{out}(i)+1$), while in the FPWA systems, it generates only one region and as a result, it appears only once in the FPWA form. In other words, in the BPWA system, one can find regions with the same continuous dynamics, but with different discrete state successors, while this is not the case for the FPWA systems. Upon detecting that the given PWA system is either in the BPWA or FPWA form, one can use the given algorithms in Lemma 30 to obtain the equivalent DHA.

Lemma 3. *Any well-posed BPWA or FPWA system can be transformed to the equivalent well-posed DHA based on t Definition 1.*

Proof**. Case 1. Translation of the BPWA into the DHA:** According to Eq. (28), it can be observed that the cells in which $x_d^{new}(k) = x_d^{new}(k+1)$ either determine the continuous dynamics associated with each discrete state of the DHA, i.e., $x_c(k+1) = A_{c_i}x_c(k) + B_{c_i}u_c(k) + f_{c_i}$ and corresponding invariant set $Inv(x_{d_i})$, or specify the reset dynamics $x_c(k+1) = A_{c_{ii}}x_c(k) + B_{c_{ii}}u_c(k) + f_{c_{ii}}$ and corresponding guard conditions $G_c(e_{ii})/G_{uc}(e_{ii})$ in the self-loop edges e_{ii}. Thus, according to Figure 5, one can construct the discrete states of the DHA with the respective continuous dynamics, invariant conditions, and possible self-loop edges with the corresponding reset dynamics and guard conditions. In the process of coding the discrete states of the DHA, since $x_d^{new}(k) = [x_d(k)^T, x_d(k-1)^T]^T$, the binary sub-vector $x_d(k)$ can be used for coding.

Regions (28) in which $x_d^{new}(k) \neq x_d^{new}(k+1)$ can be categorized in three groups:

(a) $x_d(k) = x_d(k-1)$ and $x_d(k) \neq x_d(k+1)$. The continuous state dynamic in such cells is of type $x_c(k+1) = A_{c_i}x_c(k) + B_{c_i}u_c(k) + f_{c_i}$ which has already been determined by the cells in which $x_d^{new}(k) = x_d^{new}(k+1)$ and therefore, does not give new information about continuous dynamics of the DHA. However, one can use the discrete state evolution of $x_d^{new}(k) \neq x_d^{new}(k+1)$ and the related switching condition to construct the transition edge between nodes $x_d(k)$ and $x_d(k+1)$ with the corresponding switching guards. However, in this case, there is not enough information to characterize the respective reset dynamic. This data comes from examining other cells in the BPWA structure, i.e., cases (b) and (c). With this information, the DHA net of Figure 5 is further completed, as shown in Figure 6.

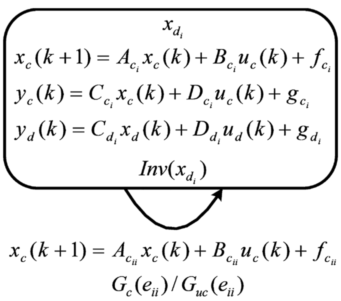

Figure 5. Concerned part of a Discrete-time Hybrid Automata (DHA) net to extract Backward PWA (BPWA) and Forward PWA (FPWA) models.

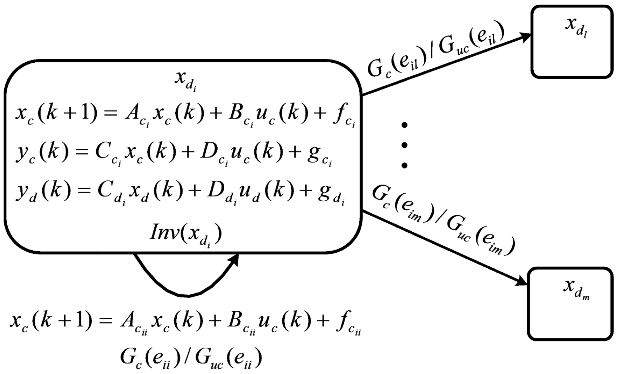

Figure 6. Concerned part of a Discrete-time Hybrid Automata (DHA) net to extract Backward PWA (BPWA) and Forward PWA (FPWA) models.

(b) $x_d(k) \neq x_d(k-1)$ and $x_d(k) = x_d(k+1)$. The continuous state dynamics in these cells are of type $x_c(k+1) = A_{c_{hi}}x_c(k) + B_{c_{hi}}u_c(k) + f_{c_{hi}}, h \neq i$. In either case, they specify the possible reset dynamics associated with the transition edge between discrete states $x_d(k-1)$ and $x_d(k)$. The respective switching condition can be determined by the available information in cases (a) or (c). The available conditions in these types of cells remaining in the discrete state $x_d(k)$ are not new since they have already been used in the cells where $x_d^{new}(k) = x_d^{new}(k+1)$ to specify the invariant and guard conditions of type $Inv(x_{d_i})$ and $G_c(e_{ii})/G_{uc}(e_{ii})$, respectively. Based on the information in this part, the DHA net of Figure 6 is completed to a greater degree, as illustrated in Figure 7.

(c) $x_d(k) \neq x_d(k-1)$ and $x_d(k) \neq x_d(k+1)$. Similar to case (b), the continuous dynamics in these types of cells are of type $x_c(k+1) = A_{c_{hi}}x_c(k) + B_{c_{hi}}u_c(k) + f_{c_{hi}}, h \neq i$. They can be used to determine the possible reset dynamics in the transition edges between $x_d(k-1)$ and $x_d(k)$. Further, the available

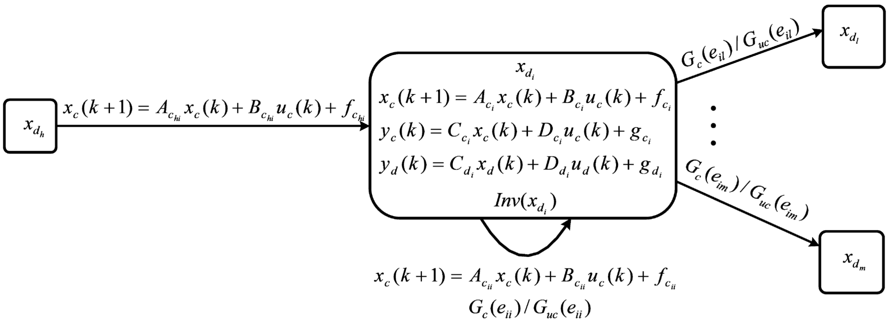

Figure 7. Concerned part of a Discrete-time Hybrid Automata (DHA) net to extract Backward PWA (BPWA) and Forward PWA (FPWA) models.

information in the corresponding cell polyhedron can be used to determine the guard conditions associated with the transition edges from discrete states $x_d(k)$ to discrete $x_d(k+1)$. All the information in this part has been used to construct the DHA in the previous cases.

With this approach by spanning all the regions of the BPWA model in Eq. (28), one can construct the equivalent DHA model.

Case 2. Translation of the FPWA into the DHA: According to Eq. (30), it can be seen that the cells in which $x_d(k)=x_d(k+1)$ either determine the continuous dynamics associated to each discrete state of the DHA, i.e., $x_c(k+1)=A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i}$ and the corresponding invariant set $Inv(x_{d_i})$, or specify the reset dynamics $x_c(k+1)=A_{c_{ii}}x_c(k)+B_{c_{ii}}u_c(k)+f_{c_{ii}}$ and corresponding guard conditions $G_c(e_{ii})/G_{uc}(e_{ii})$ in the self-loop edges e_{ii}. Based on these pieces of information, the DHA nodes with the associated continuous dynamics, invariant sets and possible self-loop edges with the corresponding reset dynamics and guard conditions can be constructed as shown in Figure 8. The cells in which $x_d(k)\neq x_d(k+1)$ can be employed to characterize the transition edges between discrete states $x_d(k)=x_{d_h}$ and $x_d(k+1)=x_{d_i}$ and the relative reset dynamics and switching guards, as shown in Figure 8.

By implementing the preceding translation technique for all regions of the FPWA model of (30), one can construct the equivalent DHA model.

3.1. BPWA and FPWA in some special cases

In this subsection, more simplified versions of the DHA and its corresponding BPWA or FPWA representation are investigated.

Definition 11. *In the DHA of Definition 1 the discrete predecessor operator Dpre is defined for the discrete state $x_{d_i}\in X_d$ as the set of discrete states x_{d_h} from which x_{d_i} can be reached in one step:*

$$Dpre(x_{d_i})=\{x_{d_h}\mid(x_{d_h},x_{d_i})\in E_c\cup E_{uc}\}. \tag{48}$$

Furthermore, discrete posterior operator Dpost is defined for the discrete state x_{d_i} as a set of discrete states x_{d_j} that can be reached from x_{d_i} in one step:

$$Dpost(x_{d_i})=\{x_{d_j}\mid(x_{d_i},x_{d_j})\in E_c\cup E_{uc}\}. \tag{49}$$

According to Remark 3, if all edges in a DHA are without reset dynamics, one can conclude that considering $x_d(k-1)$ in $x_d^{new}(k)$ is superfluous and the selection of continuous dynamics in the DHA is made by $x_d(k)$ only. In this case, the BPWA form corresponding to a part of the DHA graph with incoming and outgoing edges, but without self-loop edges (see Remark 2) depicted in Figure 4 is given by Eq. (50), as shown in Box X, where $\{x_{d_l},\ldots,x_{d_j},\ldots,x_{d_m}\}\in Dpost(x_{d_i})$.

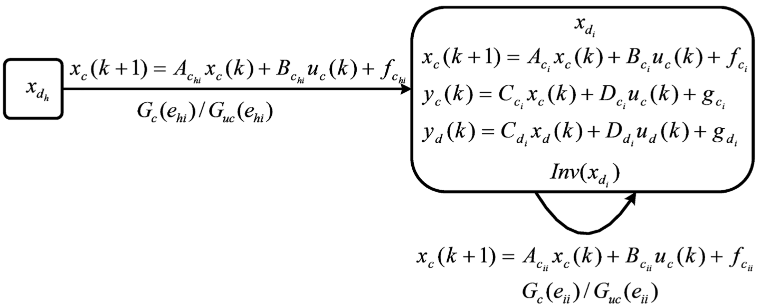

Figure 8. Concerned part of a Discrete-time Hybrid Automata (DHA) net to extract Backward PWA (BPWA) and Forward PWA (FPWA) models.

$$
\begin{bmatrix} x(k+1) \\ y(k) \end{bmatrix} = \begin{bmatrix} x_c(k+1) \\ x_d(k+1) \\ y_c(k) \\ y_d(k) \end{bmatrix} = \begin{cases}
\vdots & \vdots \\
\begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ x_{d_i} \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d(k)=x_{d_i} \wedge Inv(x_{d_i}) \\
\begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ x_{d_l} \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d(k)=x_{d_i} \wedge G_c(e_{il})/G_{uc}(e_{il}) \\
\vdots & \vdots \\
\begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ x_{d_j} \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d(k)=x_{d_i} \wedge G_c(e_{ij})/G_{uc}(e_{ij}) \\
\vdots & \vdots \\
\begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ x_{d_m} \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d(k)=x_{d_i} \wedge G_c(e_{im})/G_{uc}(e_{im}) \\
\vdots & \vdots
\end{cases} \tag{50}
$$

Box X

For the same assumption, i.e., the DHA without reset dynamic, the FPWA form corresponding to a part of the DHA shown in Figure 4 is given by Eq. (51), as shown in Box XI, where $\{x_{d_e},\ldots,x_{d_h},\ldots,x_{d_g}\} \in Dpre(x_{d_i})$. There are too many works in the literature that deal with the analysis and synthesis of only a subclass of the general PWA systems in Eq. (20). In most cases, the discrete states of the general PWA system in Eq. (20) are ignored [28,46,47,72–74]. Therefore, the following issue arises: 'under what conditions and evolution types (forward or backward) a DHA can be reduced to these special and simplified classes of PWA system without discrete states'.

Lemma 4. *In a well-posed DHA with forward evolution and without reset dynamics, let the following conditions be satisfied; then, discrete states evolution will be redundant and can be removed from the equivalent FPWA system:*

(a) *The continuous and discrete output dynamics in all discrete states belonging to $Dpre(x_{d_i})$ are identical with the continuous and discrete output dynamics at node x_{d_i} where $i \in \{1,\ldots,N_d\}$;*

(b) *$Inv(x_{d_i}) = G_c(e_{hi})$ or $Inv(x_{d_i}) = G_{uc}(e_{hi})$ where $h,i \in \{1,\ldots,N_d\}, h \neq i$ and $x_{d_h} \in Dpre(x_{d_i})$;*

(c) *$Inv(x_{d_i}) \cap Inv(x_{d_j}) = \emptyset$, where $i,j \in \{1,\ldots,N_d\}$, $i \neq j$.*

Proof. Since the DHA is well-posed and without a reset dynamic, the FPWA form corresponding to the part of the DHA shown in Figure 4 in which the continuous dynamic $A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i}$ appears can be represented as Eq. (51). According to Condition (a), the FPWA system in Eq. (51) can be rewritten as Eq. (52) is shown in Box XII. Now, based on Condition (b), Eq. (52) can be rewritten as in Eq. (53), shown in Box XIII. The FPWA form in Eq. (53) can be represented in a more compact form of Eq. (54) shown in Box XIV. Accordingly, the overall FPWA form of the DHA can be constructed as in Eq. (55) shown in Box XV. Let $Inv(x_{d_i})$ be satisfied at instant k. It is shown that the satisfaction of $Inv(x_{d_i})$ implies the satisfaction of $x_d(k) = x_{d_i}$ or $x_d(k) \in Dpre(x_{d_i})$; therefore, the discrete state evolution is redundant and continuous dynamic associated to the cell Ω_i is selected only based on the satisfaction of $Inv(x_{d_i})$.

This can be shown by a contradiction. Let

$$\begin{bmatrix} x(k+1) \\ y(k) \end{bmatrix} = \begin{bmatrix} x_c(k+1) \\ x_d(k+1) \\ y_c(k) \\ y_d(k) \end{bmatrix} = \begin{cases} \vdots & \vdots \\ \begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ x_{d_i} \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & x_d(k)=x_{d_i} \wedge Inv(x_{d_i}) \\ \begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ x_{d_i} \\ C_{c_e}x_c(k)+D_{c_e}u_c(k)+g_{c_e} \\ C_{d_e}x_d(k)+D_{d_e}u_d(k)+g_{d_e} \end{bmatrix} & x_d(k)=x_{d_e} \wedge G_c(e_{ei})/G_{uc}(e_{ei}) \\ \vdots & \vdots \\ \begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ x_{d_i} \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_h}x_d(k)+D_{d_h}u_d(k)+g_{d_h} \end{bmatrix} & x_d(k)=x_{d_h} \wedge G_c(e_{hi})/G_{uc}(e_{hi}) \\ \vdots & \vdots \\ \begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ x_{d_i} \\ C_{c_g}x_c(k)+D_{c_g}u_c(k)+g_{c_g} \\ C_{d_g}x_d(k)+D_{d_g}u_d(k)+g_{d_g} \end{bmatrix} & x_d(k)=x_{d_g} \wedge G_c(e_{gi})/G_{uc}(e_{gi}) \\ \vdots & \vdots \end{cases} \tag{51}$$

Box XI

$$\begin{bmatrix} x(k+1) \\ y(k) \end{bmatrix} = \begin{bmatrix} x_c(k+1) \\ x_d(k+1) \\ y_c(k) \\ y_d(k) \end{bmatrix} = \begin{cases} \vdots & \vdots \\ \begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ x_{d_i} \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & \begin{array}{c} x_d(k)=x_{d_i} \wedge Inv(x_{d_i}) \\ \cup(x_d(k)=x_{d_e} \wedge G_c(e_{ei})/G_{uc}(e_{ei})) \\ \vdots \\ \cup(x_d(k)=x_{d_h} \wedge G_c(e_{hi})/G_{uc}(e_{hi})) \\ \vdots \\ \cup(x_d(k)=x_{d_g} \wedge G_c(e_{gi})/G_{uc}(e_{gi})) \end{array} \\ \vdots & \vdots \end{cases} \tag{52}$$

Box XII

$Inv(x_{d_i})$ be satisfied at instant k; $x_d(k) \neq x_{d_i}$ and $x_d(k) \notin Dpre(x_{d_i})$. Since the DHA and its equivalent FPWA form in Eq. (55) are well-posed, one of the cells, say $\Omega_j, j \neq i$ is activated. Thus, at instant k, both invariant sets $Inv(x_{d_i})$ and $Inv(x_{d_j})$ are satisfied; $\exists(x_c(k), u_c(k), u_d(k)) \in Inv(x_{d_i}) \cap Inv(x_{d_j})$. However, this contradicts condition (c). Therefore, discrete states in the FPWA form of Eq. (55) are redundant and can be eliminated from the FPWA system as in Eq. (56) as shown in Box XVI.

$$\begin{bmatrix} x(k+1) \\ y(k) \end{bmatrix} = \begin{bmatrix} x_c(k+1) \\ x_d(k+1) \\ y_c(k) \\ y_d(k) \end{bmatrix} = \begin{cases} \vdots & \vdots \\ \begin{bmatrix} A_{c_i}x_c(k) + B_{c_i}u_c(k) + f_{c_i} \\ x_{d_i} \\ C_{c_i}x_c(k) + D_{c_i}u_c(k) + g_{c_i} \\ C_{d_i}x_d(k) + D_{d_i}u_d(k) + g_{d_i} \end{bmatrix} & \begin{array}{c} x_d(k) = x_{d_i} \wedge Inv(x_{d_i}) \\ \cup(x_d(k) = x_{d_e} \wedge Inv(x_{d_i})) \\ \vdots \\ \cup(x_d(k) = x_{d_h} \wedge Inv(x_{d_i})) \\ \vdots \\ \cup(x_d(k) = x_{d_g} \wedge Inv(x_{d_i})) \end{array} \\ \vdots & \vdots \end{cases} \tag{53}$$

Box XIII

$$\begin{bmatrix} x(k+1) \\ y(k) \end{bmatrix} = \begin{bmatrix} x_c(k+1) \\ x_d(k+1) \\ y_c(k) \\ y_d(k) \end{bmatrix} =$$

$$= \begin{cases} \vdots & \vdots \\ \begin{bmatrix} A_{c_i}x_c(k) + B_{c_i}u_c(k) + f_{c_i} \\ x_{d_i} \\ C_{c_i}x_c(k) + D_{c_i}u_c(k) + g_{c_i} \\ C_{d_i}x_d(k) + D_{d_i}u_d(k) + g_{d_i} \end{bmatrix} & (x_d(k) = x_{d_i} \vee x_d(k) \in Dpre(x_{d_i})) \wedge Inv(x_{d_i}) \\ \vdots & \vdots \end{cases} \tag{54}$$

Box XIV

$$\begin{bmatrix} x(k+1) \\ y(k) \end{bmatrix} = \begin{bmatrix} x_c(k+1) \\ x_d(k+1) \\ y_c(k) \\ y_d(k) \end{bmatrix} =$$

$$= \begin{cases} \begin{bmatrix} A_{c_1}x_c(k) + B_{c_1}u_c(k) + f_{c_1} \\ x_{d_1} \\ C_{c_l}x_c(k) + D_{c_l}u_c(k) + g_{c_l} \\ C_{d_l}x_d(k) + D_{d_l}u_d(k) + g_{d_l} \end{bmatrix} & (x_d(k) = x_{d_1} \vee x_d(k) \in Dpre(x_{d_1})) \wedge Inv(x_{d_1}) \\ \vdots & \vdots \\ \begin{bmatrix} A_{c_i}x_c(k) + B_{c_i}u_c(k) + f_{c_i} \\ x_{d_i} \\ C_{c_i}x_c(k) + D_{c_i}u_c(k) + g_{c_i} \\ C_{d_i}x_d(k) + D_{d_i}u_d(k) + g_{d_i} \end{bmatrix} & (x_d(k) = x_{d_i} \vee x_d(k) \in Dpre(x_{d_i})) \wedge Inv(x_{d_i}) \\ \vdots & \vdots \\ \begin{bmatrix} A_{c_{N_d}}x_c(k) + B_{c_{N_d}}u_c(k) + f_{c_{N_d}} \\ x_{d_{N_d}} \\ C_{c_{N_d}}x_c(k) + D_{c_{N_d}}u_c(k) + g_{c_{N_d}} \\ C_{d_{N_d}}x_d(k) + D_{d_{N_d}}u_d(k) + g_{d_{N_d}} \end{bmatrix} & (x_d(k) = x_{d_{N_d}} \vee x_d(k) \in Dpre(x_{d_{N_d}})) \wedge Inv(x_{d_{N_d}}) \end{cases} \tag{55}$$

Box XV

$$\begin{bmatrix} x_c(k+1) \\ y_c(k) \\ y_d(k) \end{bmatrix} = \begin{cases} \begin{bmatrix} A_{c_1}x_c(k)+B_{c_1}u_c(k)+f_{c_1} \\ C_{c_l}x_c(k)+D_{c_l}u_c(k)+g_{c_l} \\ C_{d_l}x_d(k)+D_{d_l}u_d(k)+g_{d_l} \end{bmatrix} & Inv(x_{d_1}) \\ \vdots & \vdots \\ \begin{bmatrix} A_{c_i}x_c(k)+B_{c_i}u_c(k)+f_{c_i} \\ C_{c_i}x_c(k)+D_{c_i}u_c(k)+g_{c_i} \\ C_{d_i}x_d(k)+D_{d_i}u_d(k)+g_{d_i} \end{bmatrix} & Inv(x_{d_i}) \\ \vdots & \vdots \\ \begin{bmatrix} A_{c_{N_d}}x_c(k)+B_{c_{N_d}}u_c(k)+f_{c_{N_d}} \\ C_{c_{N_d}}x_c(k)+D_{c_{N_d}}u_c(k)+g_{c_{N_d}} \\ C_{d_{N_d}}x_d(k)+D_{d_{N_d}}u_d(k)+g_{d_{N_d}} \end{bmatrix} & Inv(x_{d_{N_d}}) \end{cases} \tag{56}$$

Box XVI

3.2. Complexity analysis

In Subsection 2.1, the traditional DHA with a decomposed structure was introduced. Here, a summary of the equivalence between decomposed DHA and PWA systems proven in [41] is given. By considering fixed discrete variables $\bar{\delta}_e$, $\bar{x}_d$, and $\bar{u}_d$, one can obtain the following PWA system:

$$x_c(k+1) = A_{f_M(\bar{x}_d,\bar{u}_d,\bar{\delta}_e)}x_c(k) + B_{f_M(\bar{x}_d,\bar{u}_d,\bar{\delta}_e)} + f_{f_M(\bar{x}_d,\bar{u}_d,\bar{\delta}_e)}, \tag{57a}$$

$$x_d(k+1) = f_D(\bar{x}_d,\bar{u}_d,\bar{\delta}_e), \tag{57b}$$

$$y_c(k) = C_{f_M(\bar{x}_d,\bar{u}_d,\bar{\delta}_e)}x_c(k) + D_{f_M(\bar{x}_d,\bar{u}_d,\bar{\delta}_e)} + g_{f_M(\bar{x}_d,\bar{u}_d,\ d\bar{e}lta_e)}, \tag{57c}$$

$$y_d(k) = g_D(\bar{x}_d,\bar{u}_d,\bar{\delta}_e), \tag{57d}$$

$$\text{if } x_d(k)=\bar{x}_d, \quad u_d(k)=\bar{u}_d, \quad (x_c(k),u_c(k))\in\mathcal{X}_{\bar{\delta}_e}, \tag{27e}$$

where $\bar{\delta}_e = f_H(x_c(k),u_c(k))$ holds for any point $(x_c(k),u_c(k)) \in \mathcal{X}_{\bar{\delta}_e} \subset X_c \times U_c$. The PWA form in Eq. (57) is in the standard form of PWA-LC representation in Eq. (23). According to our discussion in Subsection 2.1, since the state evolution in the traditional DHA is of backward type, based on Definition 9, one can call Eq. (57) as the BPWA-LC form of the decomposed DHA as well.

In the worst case, there are $2^{(n_d+m_d+n_e)}$ possible combinations of discrete variables $(\bar{x}_d,\bar{u}_d,\bar{\delta}_e)$, but not all of them are feasible because of system limitations. In [41], the authors presented efficient algorithms that enumerate all feasible modes of discrete variable $\bar{\delta}_e$ that run in $O(n\ \ lp(n,d)\#M(R))$ times and $O(nd)$ space, where $lp(n,d)$ denotes the complexity of solving a Linear Program (LP) with n constraints and $d = m_c + n_c$ variables, and $\#M(R) \leq O(n^d)$ denotes the number of cells in the corresponding cell enumeration problem. These notions have several important results. First, the translation complexity from the decomposed DHA to the corresponding PWA form is of exponential type. Second, even with a small number of distinct hyperplanes, n, in the d-dimensional Euclidean space $\mathbb{R}^d$, the complexity of cell enumeration problem grows exponentially as the dimension of continuous state-input space d becomes large. This issue highlights the fact that hybrid systems with a small number of discrete-states may result in high computational burden when they are translated from the decomposed DHA formulation to the equivalent PWA forms. Finally, the translation from the decomposed DHA to PWA form needs numerical techniques based on the mixed integer programming that are computationally expensive. In contrast, according to Proposition 1, the complexity of the proposed DHA-to-PWA translation techniques based on the automaton-based DHA is of polynomial type independent of the continuous state-input and discrete input space dimensions. Moreover, in the automaton-based DHA, the translation is made analytically rather than numerically, without any need to solve cell-enumeration problems or any other mixed integer programming algorithms.

4. Examples

In this section, two examples are presented to show how to use the proposed translation techniques to extract the BPWA and FPWA forms of a given DHA. The proposed examples in our work are presented to prove the concept by hand calculations. Although they may seem toy examples, they represent the typical properties of more realistic hybrid systems. Furthermore, there are some technical subtleties behind these examples. In the continuous-time domain, the DC-DC converter model does not represent any reset dynamic or memory type switching phenomena. Therefore, in the continuous-time domain, to describe the dynamical behavior of the converter, we do not need to use hybrid automaton with memory effects; rather, the converter model is described via a PWA representation without discrete state dynamics [63]. When the converter dynamic is described in the discrete-time domain, to prevent the inductor current from taking unrealistic negative values, we need to define the reset dynamic. Here is the point: the DHA representation with discrete state dynamics is required. The DC-DC converter DHA represents both controlled and uncontrolled switching effects with reset dynamics. In contrast, the room temperature control example has intrinsically memory type dynamics in both continuous- and discrete-time domains. According to the preceding discussion, these examples are complementary to each other and complex enough to lead the nontrivial translation task by hand calculations.

4.1. Temperature control system

Figure 9 shows the DHA of a room temperature control system [37]. The aim of the control system is to regulate the room temperature $x_c(k)$ between a lower bound m and an upper bound M. When the heater is OFF or $x_d(k) = 0$, the room temperature is reduced according to the dynamic $x_c(k+1) = \lambda x_c(k)$, and when the heater is ON or $x_d(k) = 1$, the temperature increases based on the dynamic $x_c(k+1) = \lambda x_c(k) + (1-\lambda)u_c(k)$, where $u_c(k)$ is a manipulated input proportional to the heater power and $0 < \lambda < 1$ is a constant.

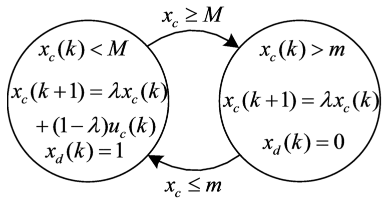

Figure 9. The Discrete-time Hybrid Automata (DHA) of a temperature control system.

Since the DHA of Figure 9 satisfies the given conditions in Theorem 1, it is well-posed. Eqs. (58) and (59) represent the equivalent BPWA and FPWA forms of the DHA model of the described temperature control system, respectively. These forms are obtained by the schemes introduced in Lemma 3 according to the DHA of the room temperature control system in Figure 9. Although the equivalency relation between the DHA and the resulting FPWA and BPWA systems automatically transfers the well-posedness property of the DHA to these systems, one can verify that the BPWA and FPWA forms in Eqs. (58) and (59), as shown in Box XVII, satisfy the given conditions in Lemma 1 and therefore, they are well-posed. As can be seen, the number of regions in the PWA form of the BPWA representation is four and in the FPWA form is two. This is in agreement with the obtained general results in Proposition 1. Note that the DHA of room temperature control system is without output and reset dynamics and according to Corollary 2, the number of independent regions in FPWA is equal to the number of nodes $N_d = 2$ and according to Corollary 1, the number

$$x(k+1) = \begin{cases} \begin{bmatrix} \lambda x_c(k) \\ 0 \end{bmatrix} & x_d(k) = 0 \wedge x_c(k) \geq m \\[2ex] \begin{bmatrix} \lambda x_c(k) \\ 1 \end{bmatrix} & x_d(k) = 0 \wedge x_c(k) < m \\[2ex] \begin{bmatrix} \lambda x_c(k) + (1-\lambda)u_c(k) \\ 1 \end{bmatrix} & x_d(k) = 1 \wedge x_c(k) < M \\[2ex] \begin{bmatrix} \lambda x_c(k) + (1-\lambda)u_c(k) \\ 0 \end{bmatrix} & x_d(k) = 1 \wedge x_c(k) \geq M \end{cases} \tag{58}$$

$$x(k+1) = \begin{cases} \begin{bmatrix} \lambda x_c(k) + (1-\lambda)u_c(k) \\ 1 \end{bmatrix} & \begin{array}{l} (x_d(k) = 1 \wedge x_c(k) < M) \\ \cup (x_d(k) = 0 \wedge x_c(k) < m) \end{array} \\[3ex] \begin{bmatrix} \lambda x_c(k) \\ 0 \end{bmatrix} & \begin{array}{l} (x_d(k) = 0 \wedge x_c(k) \geq m) \\ \cup (x_d(k) = 1 \wedge x_c(k) \geq M) \end{array} \end{cases} \tag{59}$$

Box XVII

of regions in the BPWA is the sum of the number of nodes and transition edges $N_d + N_t = 2 + 2 = 4$.

4.2. DC-DC converter

A DC-DC step-down (buck) converter with parasitic elements is shown in Figure 10 [23,63].

The DHA corresponding to the buck converter with different modes of operation is shown in Figure 11. In this figure, N in $i_L(k+N)$ is set to 2 for backward evolution and 1 for forward evolution (see Remarks 6 and 7). As a result, $i_L(k+2) = [1\ 0](A_{d_2}^2 x_c(k) + A_{d_2}B_{d_2} + B_{d_2})$ and $i_L(k+1) = [1\ 0](A_{d_2}x_c(k) + B_{d_2})$. The vector $x_d(k) = [x_{d_1}(k), x_{d_2}(k)]^T$ is defined as a discrete state vector in the DHA of the buck converter in Figure 11. This figure has three discrete states namely $[0\ 1]^T$, $[1\ 0]^T$, and $[0\ 0]^T$ that correspond for the Modes 1, 2, and 3, respectively.

A reset dynamic in Eq. (60) is considered to the edge between the discrete modes 2 and 3, which are shown by binary vectors $[1\ 0]^T$ and $[0\ 0]^T$, respectively.

$$x_c(k+1) = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} (A_{c_3}x_c(k) + h_{c_3}) = A_{c_4}x_c(k) + h_{c_4}. \tag{60}$$

Using the reset dynamic, one can update the unrealistic negative value of the inductor current to zero in one sampling step. For the backward evolution of the converter DHA, a new discrete state vector $x_d^{new} = [x_{d_1}(k), x_{d_2}(k), x_{d_1}(k-1)]^T$ is defined. Although, all bits of the discrete state vector $x_d(k-1)$ are used according to Eq. (27) to define the new discrete state vector $x_d^{new} = [x_d(k)^T, x_d(k-1)^T]^T$, in the converter DHA of Figure 11, one can see that using only $x_{d_1}(k-1)$ is enough to specify the reset dynamic in Eq. (60) between nodes 2 and 3. The well-posedness conditions given in Theorem 14 are satisfied in discrete states $[0\ 0]^T$, $[0\ 1]^T$. However, for the node $[1\ 0]^T$ that has both controlled and uncontrolled outgoing edges, condition (g) of Theorem 1 must be imposed at this node. As a result, the invariant condition $i_L(k+N) > 0 \wedge u_d(k) = 0$ and the switching guards associated with the outgoing edges of the converter DHA in the discrete state $[1\ 0]^T$, namely $u_d(k) = 1$ and $i_L(k+N) \leq 0$, are always combined by the well-posedness condition of $i_L(k+N) > 0\ \vee\ u_d(k) = 0$. Using the proposed algorithm in Lemma 3, the equivalent BPWA and FPWA models of the converter DHA in Figure 11 can be written as in Eqs. (61) and (62) shown in Boxes XVIII and XIX, respectively. As can be seen from Eqs. (61) and (62), the number of regions in the BPWA form is nine and in the FPWA representation is four. According to Proposition 1, for the DHA of the DC-DC buck converter in Figure 11, we have $n_{inre}(1) = n_{inre}(2) = 0, n_{inre}(3) = 1, n_{out}(1) = n_{out}(3) = 1, n_{out}(2) = 2, N_{self} = 0, N_d = 3, N_t = 4$, and $N_{intre} = 1$. As a result, based on the given formulations in Propositions 2 and 3, the number of independent regions in the equivalent BPWA and FPWA forms is 9 and 4, respectively, which are aligned with the obtained results in Eqs. (61) and (62).

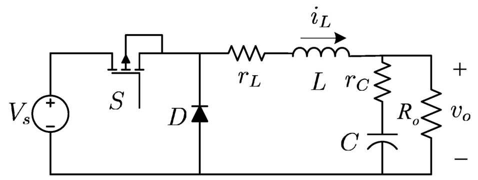

Figure 10. DC-DC buck converter.

5. Conclusion

In this paper, effective methods for the translation of an automaton-based Discrete-time Hybrid Automata (DHA) to its equivalent Piecewise affine (PWA) systems are presented. In contrast to the existing methods based on the decomposed structure of the DHA, the proposed procedure does not need any complex cell enumeration and numerical feasibility test algorithms. Hence, it can be easily employed by hand calculations and applied to the translation of complex and large-scale DHA models. It was found that changing the DHA model structure from the traditional decomposed construction to an automaton-based structure reduced the order of time complexity of the resulting translation algorithms from exponential type in the case of the decomposed DHA to the polynomial type in the case of automaton-based DHA. For the automaton-based

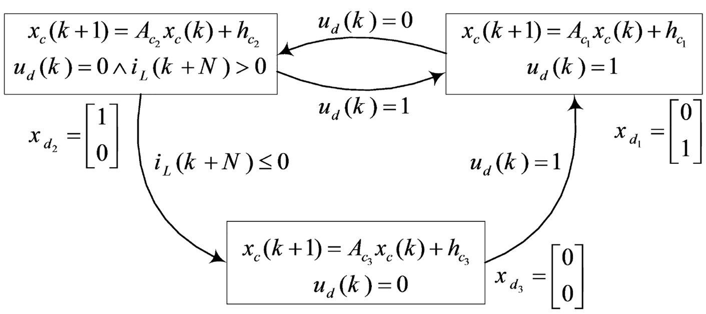

Figure 11. Discrete-time Hybrid Automata (DHA) of a buck converter.

$$x(k+1)=$$

$$=\begin{cases}
\begin{bmatrix} A_{d_3}x_c(k)+B_{d_3} \\ 0 \\ 0 \\ 0 \end{bmatrix} & x_d^{new}(k)=[0\ 0\ 0]^T \wedge u_d(k)=0 \\
\begin{bmatrix} A_{d_3}x_c(k)+B_{d_3} \\ 0 \\ 1 \\ 0 \end{bmatrix} & x_d^{new}(k)=[0\ 0\ 0]^T \wedge u_d(k)=1 \\
\begin{bmatrix} A_{d_1}x_c(k)+B_{d_1} \\ 0 \\ 1 \\ 0 \end{bmatrix} & \begin{array}{l} x_d^{new}(k)=[0\ 1\ 0]^T \wedge u_d(k)=1 \\ \cup(x_d^{new}(k)=[0\ 1\ 1]^T \wedge u_d(k)=1) \end{array} \\
\begin{bmatrix} A_{d_1}x_c(k)+B_{d_1} \\ 1 \\ 0 \\ 0 \end{bmatrix} & x_d^{new}(k)=[0\ 1\ 0]^T \wedge u_d(k)=0 \\
\begin{bmatrix} A_{d_2}x_c(k)+B_{d_2} \\ 1 \\ 0 \\ 1 \end{bmatrix} & \begin{array}{l} x_d^{new}(k)=[1\ 0\ 1]^T \wedge u_d(k)=0 \wedge [1\ 0](A_{d_2}^2x_c(k)+A_{d_2}B_{d_2}+B_{d_2})>0 \\ \cup(x_d^{new}(k)=[1\ 0\ 0]^T \wedge u_d(k)=0 \wedge [1\ 0](A_{d_2}^2x_c(k)+A_{d_2}B_{d_2}+B_{d_2})>0) \end{array} \\
\begin{bmatrix} A_{d_2}x_c(k)+B_{d_2} \\ 0 \\ 0 \\ 1 \end{bmatrix} & \begin{array}{l} x_d^{new}(k)=[1\ 0\ 1]^T \wedge u_d(k)=0 \wedge [1\ 0](A_{d_2}^2x_c(k)+A_{d_2}B_{d_2}+B_{d_2})\leq 0 \\ \cup(x_d^{new}(k)=[1\ 0\ 0]^T \wedge u_d(k)=0 \wedge [1\ 0](A_{d_2}^2x_c(k)+A_{d_2}B_{d_2}+B_{d_2})\leq 0) \end{array} \\
\begin{bmatrix} A_{d_2}x_c(k)+B_{d_2} \\ 0 \\ 1 \\ 1 \end{bmatrix} & \begin{array}{l} x_d^{new}(k)=[1\ 0\ 0]^T \wedge u_d(k)=1 \wedge [1\ 0](A_{d_2}^2x_c(k)+A_{d_2}B_{d_2}+B_{d_2})>0 \\ \cup(x_d^{new}(k)=[1\ 0\ 1]^T \wedge u_d(k)=1 \wedge [1\ 0](A_{d_2}^2x_c(k)+A_{d_2}B_{d_2}+B_{d_2})>0) \end{array} \\
\begin{bmatrix} A_{d_4}x_c(k)+B_{d_4} \\ 0 \\ 0 \\ 0 \end{bmatrix} & x_d^{new}(k)=[0\ 0\ 1]^T \wedge u_d(k)=0 \\
\begin{bmatrix} A_{d_4}x_c(k)+B_{d_4} \\ 0 \\ 1 \\ 0 \end{bmatrix} & x_d^{new}(k)=[0\ 0\ 1]^T \wedge u_d(k)=1
\end{cases} \tag{61}$$

Box XVIII

DHA, two types of evolutions, i.e., backward and forward evolutions, were defined and associated with each type of evolution two types of PWA systems, i.e., BPWA and FPWA systems, were extracted. Some formulations were provided that could quantify the exact number of regions in the BPWA and FPWA systems. It was shown that the number of such regions in the BPWA systems was larger than that of the FPWA forms of an order of at least N_t, where N_t is the number of transition edges in the DHA graph.

$$x(k+1) = \\ = \begin{cases} \begin{bmatrix} A_{d_3}x_c(k)+B_{d_3} \\ 0 \\ 0 \end{bmatrix} & x_d(k)=[0\ 0]^T \wedge u_d(k)=0 \\ \begin{bmatrix} A_{d_1}x_c(k)+B_{d_1} \\ 0 \\ 1 \end{bmatrix} & \begin{array}{l}(x_d(k)=[0\ 0]^T \wedge u_d(k)=1) \\ \cup(x_d(k)=[0\ 1]^T \wedge u_d(k)=1) \\ \cup(x_d(k)=[1\ 0]^T \wedge u_d(k)=1 \wedge [1\ 0](A_{d_2}x_c(k)+B_{d_2})>0)\end{array} \\ \begin{bmatrix} A_{d_2}x_c(k)+B_{d_2} \\ 1 \\ 0 \end{bmatrix} & \begin{array}{l}(x_d(k)=[0\ 1]^T \wedge u_d(k)=0) \\ \cup(x_d(k)=[1\ 0]^T \wedge u_d(k)=0 \wedge [1\ 0](A_{d_2}x_c(k)+B_{d_2})>0)\end{array} \\ \begin{bmatrix} A_{d_4}x_c(k)+B_{d_4} \\ 0 \\ 0 \end{bmatrix} & x_d(k)=[1\ 0]^T \wedge u_d(k)=0 \wedge [1\ 0](A_{d_2}x_c(k)+B_{d_2})\leq 0 \end{cases} \tag{62}$$

Box XIX

Examples were presented to provide evidence for the merit of the proposed techniques. The significance of this research topic is to transfer the analysis and synthesis techniques, e.g., the controller synthesis and stability analysis, from one class of hybrid systems to another.

Acknowledgement

Mohammad Hejri would like to thank Professor Alessandro Giua of the University of Cagliari for his valuable comments and discussions.

References

1. Cassandras, C.G. and Lafortune, S. *Introduction to Discrete Event Systems*, 2nd Edition, Springer Berlin Heidelberg (2008).
2. Pcolka, M., Zacekova, E., Celikovsky, S., and Sebek, M. "Toward a smart car: hybrid nonlinear predictive controller with adaptive horizon", *IEEE Trans. Control Syst. Technol.*, **26**(6), pp. 1970–1981 (2017).
3. Tantawy, A., Koutsoukos, X., and Biswas, G. "Aircraft power generators: hybrid modeling and simulation for fault detection", *IEEE Trans. Aerosp. Electron. Syst.*, **48**(1), pp. 552–571 (2012).
4. Soler, M., Kamgarpour, M., Lloret, J., and Lygeros, J. "A hybrid optimal control approach to fuel-efficient aircraft conflict avoidance", *IEEE Trans. Intell. Transp. Syst.*, **17**(7), pp. 1826–1838 (2016).
5. Manon, P., Valentin-Roubinet, C., and Gilles, G. "Optimal control of hybrid dynamical systems: application in process engineering", *Control Eng. Pract.*, **10**(2), pp. 133–149 (2002).
6. Lee, J., Bohacek, S., Hespanha, J.P., and Obraczka, K. "Modeling communication networks with hybrid systems", *IEEE/ACM Trans. Netw.*, **15**(3), pp. 630–643 (2007).
7. Ding, J., d Gillula, J.H., Huang, H., Vitus, M.P., Zhang, W., and Tomlin, C.J. "Hybrid systems in robotics", *IEEE Robot. Autom. Mag.*, **18**(3), pp. 33–43 (2011).
8. Bortolussi, L. and Policriti, A. "Hybrid systems and biology" In *Formal Methods for Computational Systems Biology: 8th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, Advanced Lectures*, Springer Berlin Heidelberg, pp. 424–448 (2008).
9. Theunisse, T.A.F., Chai, J., Sanfelice, R.G., and Heemels, W.P.M.H. "Robust global stabilization of the DC-DC boost converter via hybrid control", *IEEE Trans. Circuits Syst. I*, **62**(4), pp. 1052–1061 (2015).
10. Moarref, M. and Rodrigues, L. "Piecewise affine networked control systems" *IEEE Trans. Control Netw. Syst.*, **3**(2), pp. 173–181 (2016).
11. Fourlas, G.K., Kyriakopoulos, K.J., and Vournas, C.D. "Hybrid systems modeling for power systems", *IEEE Circuits Syst. Mag.*, **4**(3), pp. 16–23 (2004).
12. Kowalewski, S. "Introduction to the analysis and verification of hybrid systems", In *Modelling, Analysis, and Design of Hybrid Systems*, S. Engell, G. Frehse and E. Schnieder, Eds., pp. 153–171, Springer Berlin Heidelberg (2002).
13. Belta, C., Yordanov, B., and Gol, E.A. *Formal Methods for Discrete-Time Dynamical Systems*, J. Kacprzyk, Ed., Springer International Publishing (2017).
14. Borrelli, F., Baotić, M., Bemporad, A., and Morari, M. "Dynamic programming for constrained optimal

control of discrete-time linear hybrid systems", *Automatica*, **41**(10), pp. 1709–1721 (2005).

15. Karer, G. and Škrjanc, I. "Introduction to predictive control of complex systems", In *Predictive Approaches to Control of Complex Systems*, Springer Berlin Heidelberg, pp. 147–156 (2013).

16. Johansson, K.H., Egerstedt, M., Lygeros, J., and Sastry, S. "On the regularization of zeno hybrid automata", *Syst. Control Lett.*, **38**, pp. 141–150 (1999).

17. Floudas, C.A. and Lin, X. "Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review", *Comput. Chem. Eng.*, **28**(11), pp. 2109–2129 (2004).

18. Imura, J.-ichi "Optimal control of sampled-data piecewise affine systems", *Automatica*, **40**(4), pp. 661–669 (2004).

19. Stauner, T. "Discrete-time refinement of hybrid automata", *Hybrid Systems: Computation and Control*, C.J. Tomlin and M.R. Greenstreet, Eds., pp. 407–420, Springer Berlin Heidelberg (2002).

20. Zaytoon, J. "Hybrid dynamic systems: overview and discussion on verification methods", *Informatics in Control, Automation and Robotics II*, J. Filipe, J.-L. Ferrier, J. A. Cetto and M. Carvalho, Eds., pp. 17–26, Springer Netherlands (2007).

21. Heemels, W.P.H.M., De Schutter, B., and Bemporad, A. "Equivalence of hybrid dynamical models", *Automatica*, **37**(7), pp. 1085–1091 (2001).

22. Bemporad, A. and Morari, M. "Control of systems integrating logic, dynamic and constraints", *Automatica*, **35**(3), pp. 407–427 (1999).

23. Hejri, M., Giua, A., and Mokhtari, H. "On the complexity and dynamical properties of mixed logical dynamical systems via an automaton-based realization of discrete-time hybrid automaton", *Int. J. of Robust Nonlin.*, **28**(16), pp. 4713–4746 (2018).

24. Heemels, W.P.M.H., Schumacher, J.M., and Weiland, S. "Linear complementarity systems", *SIAM J. Appl. Math.*, **60**(4), pp. 1234–1269 (2000).

25. De Schutter, B. and DeMoor, B. "The extended linear complementarity problem and the modeling and analysis of hybrid systems", *Lecture Notes in Computer Science*, P. Antsaklis, W. Kohn, M. Lemmon, A. Nerode, and S. Sastry, Eds., pp. 70–85 (1999).

26. De Schutter, B. and van den Boom, T. "Model predictive control for max-plus-linear systems", *American Control Conference*, pp. 4046–4050 (2000).

27. Sontag, E.D. "Nonlinear regulation: the piecewise linear approach", *IEEE Trans. Autom. Control*, **26**(2), pp. 346–358 (1981).

28. Ferrari-Trecate, G., Cuzzola, F.A., Mignone, D., and Morari M. "Analysis of discrete-time piecewise affine and hybrid systems", *Automatica*, **38**(12), pp. 2139–2146 (2002).

29. Johansson, M. and Rantzer, A. "Computation of piecewise quadratic Lyapunov functions for hybrid systems", *IEEE Trans. Autom. Control*, **43**(4), pp. 555–559 (1998).

30. Cuzzola, F.A. and Morari, M. "A generalized approach for analysis and control of discrete-time piecewise affine and hybrid systems", M.D. Di Benedetto and A. Sangiovanni-Vincentelli, Eds., *Hybrid Systems: Computation and Control: 4th International Workshop, HSCC 2001*, Rome, Italy, Springer Berlin Heidelberg, pp. 189–203 (2001).

31. Hajiahmadi, M., De Schutter, B., and Hellendoorn, H. "Design of stabilizing switching laws for mixed switched affine systems", *IEEE Trans. Autom. Control*, **61**(6), pp. 1676–1681 (2016).

32. Van der Schaft, A.J. and Schumacher, J.M. "Complementarity modeling of hybrid systems", *IEEE Trans. Autom. Control*, **43**(4), pp. 483–490 (1998).

33. Lygeros, J., Johansson, K.H., Simic, S.N., Zhang, J., and Sastry, S.S. "Dynamical properties of hybrid automata", *IEEE Trans. Autom. Control*, **48**(1), pp. 2–17 (2003).

34. Camacho, E.F., Ramirez, D.R., Limon, D., Monuz de la Pena, D., and Alamo, T. "Model predictive control techniques for hybrid systems", *Annu. Rev. Control*, **34**, pp. 21–31 (2010).

35. Torrisi, F.D. and Bemporad, A. "HYSDEL-a tool for generating computational hybrid models for analysis and synthesis problems", *IEEE Trans. Control Syst. Technol.*, **12**(2), pp. 235–249 (2004).

36. Borrelli, F., Bemprad, A., and Morari, M., *Predictive Control for Linear and Hybrid Systems*, Cambridge University Press (2017).

37. Bemporad, A., Ferrari-Trecate, G., and Morari, M. "Observability and controllability of piecewise affine and hybrid systems", *IEEE Trans. Autom. Control*, **45**(10), pp. 1864–1876 (2000).

38. Geyer, T., Torrisi, F.D., and Morari, M. "Efficient mode enumeration of compositional hybrid systems", *Hybrid Systems: Computation and Control*, O. Maler and A. Pnueli, Eds., Springer Berlin Heidelberg, pp. 216–232 (2003).

39. Potocnik, B., Music, G., and Zupancic B. "A new technique for translating discrete hybrid automata into piecewise affine systems", *Math. Comp. Model. Dyn.*, **10**(1), pp. 41–57 (2004).

40. Bemporad, A. "Efficient conversion of mixed logical dynamical systems into an equivalent piecewise affine form", *IEEE Trans. Autom. Control*, **49**(5), pp. 832–838 (2004).

41. Geyer, T., Torrisi, F.D., and Morari, M. "Efficient mode enumeration of compositional hybrid systems", *Int. J. Control*, **83**(2), pp. 313–329 (2010).

42. Groot, N., De Schutter, B., and Hellendoorn, H. "Integrated model predictive traffic and emission control using a piecewise-affine approach", *IEEE Trans. Intell. Transp. Syst.*, **14**(2), pp. 587–598 (2013).

43. Ferrari-Trecate, G., Cuzzola, F.A., and Morari M. "Lagrange stability and performance analysis of discrete-time piecewise affine systems with logic states", *Int. J. Control*, **76**(16), pp. 1585–1598 (2003).

44. Ferrari-Trecate, G., Cuzzola, F.A., and Morari, M. "An LMI approach for H-infinity analysis and control of discrete-time piecewise affine systems", *Int. J. Control*, **75**(16–17), pp. 1293–1301 (2002).

45. Mignone, D., Ferrari-Trecate, G., and Morari, M. "Stability and stabilization of piecewise affine and hybrid systems: an LMI approach", *Proceedings of the 39th IEEE Conference on Decision and Control*, **1**, pp. 504–509 (2000).

46. Johansson, M., *Piecewise Linear Control Systems: A Computational Approach*, Springer (2003).

47. Xu, J. and Xie, L., *Control and Estimation of Piecewise Affine Systems*, Woodhead publishing (2013).

48. Christophersen, F., *Optimal Control of Constrained Piecewise Affine Systems*, M. Thoma and M. Morari, Eds., Springer Berlin Heidelberg (2007).

49. Sontag, E.D. "Interconnected automata and linear systems: a theoretical framework in discrete-time", *Proceedings of the DIMACS/SYCON Workshop on Hybrid Systems III: Verification and Control*, Springer-Verlag New York, pp. 436–448 (1996).

50. Cairano, S. and Bemporad, A. "Equivalent piecewise affine models of linear hybrid automata", *IEEE Trans. Autom. Control*, **55**(2), pp. 498–502 (2010).

51. Henzinger, T.A. "The theory of hybrid automata", *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science (LICS '96)*, New Brunswick, pp. 278–292 (1996).

52. Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T.A., Ho, P.-H., Nicollin, X., Olivero, A., Sifakis, J., and Yovine, S. "The algorithmic analysis of hybrid systems", *Theor. Comput. Sci.*, **138**(1), pp. 3–34 (1995).

53. Alur, R., Courcoubetis, C., Henzinger, T.A., and Ho, P.-H. "Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems", *Hybrid Systems*, pp. 209–229, Springer Berlin Heidelberg (1993).

54. Nicollin, X., Olivero, A., Sifakis, J., and Yovine, S. "An approach to the description and analysis of hybrid systems", In *Hybrid Systems*, R.L. Grossman, A. Nerode, A. P. Ravn and H. Rischel, Eds., pp. 149–178, Springer Berlin Heidelberg (1993).

55. Lygeros, J., Godbole, D.N., and Sastry, S.S. "Verified hybrid controllers for automated vehicles", *IEEE Trans. Autom. Control*, **43**(4), pp. 522–539 (1998).

56. Stursberg, O., Panek, S., Till, J., and Engell, S. "Generation of optimal control policies for systems with switched hybrid dynamics", *Modelling, Analysis, and Design of Hybrid Systems*, S. Engell, G. Frehse and E. Schnieder, Eds., pp. 337–352, Springer Berlin Heidelberg (2002).

57. Stursberg, O. and Engell, S. "Optimal control of switched continuous systems using mixed-integer programmings", *IFAC Proceedings Volumes*, **35**(1), pp. 433–438 (2002).

58. Stursberg, O. and Panek, S. "Control of switched hybrid systems based on disjunctive formulations", *Hybrid Systems: Computation and Control*, C.J. Tomlin and M.R. Greenstreet, Eds., Springer Berlin Heidelberg, pp. 421–435 (2002).

59. Pang, Y. and Spathopoulos, M.P. "Time-optimal control for discrete-time hybrid automata", *Int. J. Control*, **78**(11), pp. 847–863 (2005).

60. Zoncu M., Balluchi, A., Sangiovanni-Vicentelli, A.L., and Bicchi, A. "On the stabilization of linear discrete-time hybrid automata", *42nd IEEE International Conference on Decision and Control*, pp. 1147–1152 (2003).

61. Seatzu, C., Gromov, D., Raisch, J., Corona, D., and Giua, A. "Optimal control of discrete-time hybrid automata under safety and liveness constraints", *Nonlinear Anal.-Theor.*, **65**(6), pp. 1188–1210 (2006).

62. Hejri, M. "Hybrid modeling and control of power electrinic converters", Ph.D Thesis, Sharif University of Technology Cotutorship with University of Cagliari, Iran and Italy (2010).

63. Hejri, M. and Giua, A. "Hybrid modeling and control of switching DC-DC converters via MLD systems", *IEEE 7th International Conference on Automation Science and Engineering*, Trieste, Italy (2011).

64. Hejri, M. and Mokhtari, H. "Hybrid modeling and control of a DC-DC boost converter via Extended Mixed Logical Dynamical systems (EMLDs)", *Power Electronics, Drive Systems and Technologies Conference (PEDSTC)*, Tehran, Iran, pp. 373–378 (2014).

65. Bemporad, A. "An efficient technique for translating mixed logical dynamical systems into piecewise affine systems", *41th IEEE Conf. on Decision and Control*, pp. 1970–1975 (2002).

66. Xia, X. "Well posedness of piecewise-linear systems with multiple modes and multiple criteria", *IEEE Trans. Autom. Control*, **47**(10), pp. 1716–1720 (2002).

67. Sahan, G. and Eldem, V. "Well posedness conditions for bimodal piecewise affine systems", *Syst. Control Lett.*, **83**, pp. 9–18 (2015).

68. Ferrari-Trecate, G., Cuzzola, F.A., and Morari, M. "Analysis of discrete-time PWA systems with logic states", *Hybrid Systems: Computation and Control*, C.J. Tomlin, and M.R. Greenstreet, Eds., pp. 194–208, Springer Berlin Heidelberg (2002).

69. Bemporad, A. and Fukuda, K., and Torrisi, F.D. "Convexity recognition of the union of polyhedra", *Computational Geometry*, **18**(3), pp. 141–154 (2001).

70. Branicky, M.S., Borkar, V.S., and Mitter, S.K. "A unified framework for hybrid control: model and optimal control theory", *IEEE Trans. Autom. Control*, **43**(1), pp. 31–45 (1998).

71. Imura, J.-ichi and Van der Schaft, A. "Characterization of well-posedness of piecewise-linear systems", *IEEE Trans. Autom. Control*, **45**(9), pp. 1600–1619 (2000).

72. Lazar M., Heemels, W.P.M.H., Weiland, S., and Bemporad, A. "Stabilizing model predictive control of hybrid systems", *IEEE Trans. Autom. Control*, **51**(11), pp. 1813–1818 (2006).

73. Lin, H. and Antsaklis, P.J. "Stability and stabilizability of switched linear systems: a survay of recent results", *IEEE Trans. Autom. Control*, **54**(2), pp. 308–322 (2009).

74. Esfahani P.S. and Pieper J.K. "H_∞ model predictive control for constrained discrete-time piecewise affine systems", *Int. J. of Robust Nonlin.*, **28**(6), pp. 1973–1995 (2017).

Biographies

Mohammad Hejri received his BSc degree from Tabriz University in 2000 and the MSc degree from Sharif University of Technology, Tehran, Iran in 2002 both in Electrical Engineering. He received his PhD degree in Electrical Engineering from Sharif University of Technology, Tehran and the University of Cagliari, Cagliari, Italy, in 2010 as a co-tutorship program. He has been with several industries and research centers such as Iran Tractor Foundry Company, Azerbaijan Regional Electric Company, Tabriz Oil and Refining Company, and Iran's Niroo (Energy) Research Institute (NRI). From 2010 to 2012, he was a Postdoctoral Research Associate at the Department of Electric Power and Energy Systems, School of Electrical Engineering, Royal Institute of Technology (KTH), Stockholm, Sweden. Since 2012, he joined the Department of Electrical Engineering, Sahand University of Technology, Tabriz, where he is currently an Associate Professor. His research interests include control theory with applications in power electronics, renewable energy, and power systems.

Hossein Mokhtari received his BSc degree in Electrical Engineering from Tehran University in 1989. From 1989 to 1992, he worked in the Consulting Division of Power Systems Dispatching Projects for the Electric Power Research Center Institute. Dr. Mokhtari obtained his MSc degree in Power Electronics from the University of New Brunswick, Fredericton, NB, Canada in 1994 and his PhD degree in Power Electronics/Power Quality from the University of Toronto, Toronto, ON, Canada in 1999. Since 2000, he has been with the Department of Electrical Engineering, Sharif University of Technology, Tehran, where he is currently a Professor. He is also a senior consultant to several utilities and industries.