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

Mohammad Hejri\textsuperscript{a,*}, Hossein Mokhtari\textsuperscript{b}

\textsuperscript{a}Department of Electrical Engineering, Sahand University of Technology, P.O. Box: 51335-1996, Tabriz, Iran
\textsuperscript{b}Department of Electrical Engineering, Sharif University of Technology, P.O. Box: 11365-9363, Tehran, Iran

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 DHA and piecewise affine (PWA) systems. As an 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 which 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 well-posedness 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 a polynomial complexity in contrast to the existing numerical translation techniques via decomposed structure of the DHA with an exponential complexity. Examples are presented to confirm the effectiveness of the proposed translation techniques.

Keywords: Automaton-based discrete-time hybrid automaton, Piecewise affine (PWA) systems, Well-posedness, Complexity, Equivalency and 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\cite{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 automobiles\cite{2}, aircrafts\cite{3}, air traffic control systems\cite{4}, process control systems\cite{5}, communication networks\cite{6}, robotics\cite{7}, biology\cite{8}, circuits and electronics\cite{9}, networked control systems\cite{10} and power systems\cite{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\cite{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 to be very difficult. Another reason on the difficulty of mathematical treatment of 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 the 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 the 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 phenomena which are 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, there are some limitations with discrete-time models. Because of the continuous nature of the time and the concept of discretization, a level of approximation should be adopted during 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 within an optimization problem, the size of the resulting mathematical programming increases due to the introduction of 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 needed to use small discretization time, that in turn leads to large combinatorial problems of intractable size [17, 18]. Nevertheless, when a continuous-time hybrid system model comes to implementation, a part of the requirements will be realized in a software in the 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 discrete-time hybrid dynamical systems theory 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], have been presented. The existence of many equivalent modeling frameworks for a hybrid system confirms the importance 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 piecewise affine systems is made easily in this modeling framework [29, 30, 31], while the existence and uniqueness of the solutions can be investigated easier in linear complementarity systems [24, 32] and hybrid automata [33, 23]. 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], discrete-time hybrid automata (DHA) have been proposed as a general modeling framework to obtain hybrid models oriented to the solution of analysis and synthesis problems. The proposed DHA has a decomposed structure in the sense that whose continuous dynamics is described by affine difference equations, and the discrete dynamics are described by finite state machines. These dynamics are interfaced by mode selector and event generator 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 therefore is an starting point of modeling, simulating and composition of a wide classes 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, 38, 39, 40, 41]. While model predictive control techniques on the basis of online mixed integer optimization is more computationally tractable for MLD systems [42], nevertheless, PWA formulation is more suitable for other analysis and synthesis techniques of hybrid systems such as stability and stabilization [28, 43, 44, 45]. The analysis and synthesis of PWA systems have received much attention in the literature, and one can refer to [46, 47] and references therein. Depending on the structure of physical hybrid system, it is often a challenging task to directly obtain a corresponding 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 translated to the corresponding PWA.
There are two different approaches for translating a decomposed DHA to its equivalent PWA form. In the first method, DHA can be translated to its equivalent MLD representation, and then from MLD to its equivalent form by the proposed algorithms presented in [37, 40]. The other method is that the DHA is directly translated to its equivalent PWA form by 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 the increase of the dimension of the Euclidian space in which the hyperplanes of the event generator element of the decomposed DHA are defined. One reason for such a 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 in the translation techniques. However, in automaton-based DHA [23], these 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 to feasibility tests and applying mixed integer programming.

Although obtaining PWA representation from decomposed DHA and MLD systems has received much attention in the literature, however, up to our best knowledge, only few works [49, 50] have addressed the problem of equivalence 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 is 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, are not discussed. In [50], the relationship between an autonomous continuous-time linear hybrid automaton (LHA) and piecewise affine systems with disturbance inputs is examined. Modeling of the uncertainty associated with LHA transitions is performed via 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 of this paper:

Motivated by the issues mentioned above, the goal of this paper, is to solve the translation problem of an automaton-based DHA to its equivalent PWA form with an efficient and low-complexity manner. In the existing literature this problem is solved via traditional decomposed DHA [35] with an exponential complexity through numerical solving of cell enumeration problem and mixed integer programming techniques [38, 39, 41]. We show that this problem is easier solved via an automaton-based DHA with a 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 as automaton-based DHA is formally defined and represented. This modeling framework is established based on the continuous-time hybrid automata in the literature [51, 33, 52, 53, 54, 55] with some modifications in their modeling structures. The proposed continuous-time hybrid automaton models in [51, 33, 52, 53, 54], have not considered the inputs, outputs and controlled switching phenomena. The proposed model in this paper is a generalization of the HA model in [51, 33, 52, 53, 54], in which inputs, outputs and controlled switching phenomena are considered as well. In some aspects, the proposed model at 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 separated to the controlled and uncontrolled types, and the related notions are defined in more detail. The other difference is that in [55] the discrete inputs influence both switching between submodels and continuous dynamics, however, in our work discrete inputs affect only on the switching between modes and not on the continuous dynamics. As compared to our earlier work in [23], the proposed model in the present work is a bit modified and generalized in a way 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, i.e., the traditional backward evolution [35] and forward evolution [23] are defined for the proposed automaton-based DHA. Sufficient conditions are derived for the well-posedness of the proposed automaton-based DHA. The constructive conversion of the proposed automaton-based DHA to its equivalent PWA models and vise verse, based on two different types of state evolutions (backward and forward) are formally represented.

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 property of the DHA and PWA systems are investigated in this section. Section 5 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 to its equivalent FPWA and BPWA systems and vice verse are represented. The effectiveness of the proposed translation techniques are 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 $i^{th}$ element of $x$. Equalities for real vectors must be understood componentwise. We use $\cap$ and $\cup$ to denote "for all" and "there exists", respectively. For real numbers $x, y \in \mathbb{R}$, $x \neq y$ if and only if $\exists \delta \in [1, 2, ..., k]$ such that $x_i = y_i$. Given a set $A$, $P(A)$ is the power set of $A$, i.e., the set of all subsets of $A$. Let $\varphi$ 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 $\lor$ are used to denote the logical "AND", "OR", respectively.

2. Discrete-time models of hybrid systems

There are different approaches to develop 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.

Contrarily to the decomposed DHA, in automaton-based DHA, like to continuous-time hybrid automata, the continuous dynamic is a property of the state of the automaton, but 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 as mode selector (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 decomposed structure

As shown in Figure 1, the DHA proposed in [35] decomposes the hybrid system into two interacting subsystems: a continuous system with the continuous input $u_c(k) \in U_c \subseteq \mathbb{R}^p_c$, 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 a discrete system with discrete inputs $u_d(k) \in U_d \subseteq \{0, 1\}^{m \times n}$, 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 Finite State Machine (FSM) respectively. The interaction between these two subsystems are realized by two interfaces, called as Event Generator (EG) and Mode Selector (MS), respectively. The former maps continuous-valued signals into discrete-valued signals $\delta_c(k)$, 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, we define each of these components. A SAS is a collection of affine systems

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

$$
y_c(k) = C_{i(k)}x_c(k) + D_{i(k)}u_c(k) + g_{i(k)}
$$

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

\[
\begin{align*}
    x_d(k+1) &= f_D(x_d(k), u_d(k), \delta_d(k)) \\
    y_d(k) &= g_D(x_d(k), u_d(k), \delta_d(k))
\end{align*}
\] (2)

where \( f_D(\ldots) : X_d \times U_d \times D \to X_d \) and \( g_D(\ldots) : X_d \times U_d \times D \to Y_d \) are deterministic discrete functions and \( D \subseteq \{0, 1\}^n \). The EG generates binary event signal \( \delta_d(k) \) according to the fulfillment of the affine constraints or thresholds

\[
\delta_d(k) = f_M(x_c(k), u_c(k))
\] (3)

where \( f_M(\ldots) : X_c \times U_c \to D \) is a vector of descriptive functions of a linear hyperplane. The MS interface is described by

\[
i(k) = f_M(x_c(k), u_c(k), \delta_c(k))
\] (4)

where \( f_M(\ldots) : X_d \times U_d \times D \to I \) is a deterministic discrete function. A mode switch occurs at the time instant \( k \) if \( i(k-1) \neq i(k) \). In correspondence with a mode switch \( i(k) = j, i(k-1) = h \neq j, h, j \in D \), 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 \( x_c(k+1) = A_h x_c(k) + B_h u_c(k) + f_h \), 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 more than that of the decomposed DHA. Moreover, the graphical representation of the automaton-based DHA leads to an ease communication with the model for the solution of some of the problems such as the translation of the model to its equivalent PWA system. Nevertheless, for the decomposed DHA a tool called as 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] that provides a multi-modeling capability. Moreover, the decomposition of hybrid systems into a continuous and a discrete subsystems shows the hybrid nature of the system explicitly and makes it possible to use the methods available for continuous and discrete systems to the separate subsystems, although the obtained results for each isolated subsystem are not valid for the overall hybrid system [12]. In Subsection 3.2, we will come back to the decomposed DHA model where a complexity analysis is performed on the translation techniques from the decomposed DHA and automaton-based DHA to Piecewise Affine (PWA) systems.

2.2. The automaton-based realization of a DHA

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

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

\[
DHA = (X_c, X_d, Y_c, Y_d, U_c, U_d, f_A(\cdot, \cdot), f_J(\cdot, \cdot), f_y(\cdot, \cdot), f_{x_h}(\cdot, \cdot), Init, Inv(\cdot), E_c, E_w, G_c(\cdot), G_w(\cdot), R_{x_h}(\cdot, \cdot))
\] (5)

where

- \( X_c \subseteq \mathbb{R}^n \) is the set of admissible continuous states;
- \( X_d = \{x_d, x_{d2}, \ldots, x_{dn}\} \subseteq \{0, 1\}^{n_d} \) is a finite set of discrete states where \( n_d \in \mathbb{N} \) is the number of discrete states of the DHA;
- \( Y_c \subseteq \mathbb{R}^p \) is the set of admissible continuous outputs;
- \( Y_d \subseteq \{0, 1\}^m \) is the set of admissible discrete outputs;
- \( U_c \subseteq \mathbb{R}^m \) is the set of admissible continuous inputs;
- $U_d \subseteq \{0, 1\}^{n_x}$ 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;
- $f_s(\cdot, \cdot, \cdot) : X_c \times U_c \times X_d \to X_c$ is a function which determines the evolution of the continuous state $x_c(k)$ at each discrete state;
- $f_y(\cdot, \cdot, \cdot) : X_c \times U_c \times X_d \to X_c$ is a function which determines the evolution of the continuous output $y_c(k)$ at each discrete state;
- $f_d(\cdot, \cdot, \cdot) : X_d \times U_d \to Y_d$ is a function which specifies the discrete output $y_d(k)$ evolution corresponding to each discrete state;
- $Init \subseteq X_c \times X_d$ is the 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;
- $R_x(\cdot, \cdot, \cdot) : X_c \times U_c \times E \to X_c$ 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 Fig. 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:

$$x_c(k+1) = f_s(x_c(k), u_c(k), x_d) = A_c x_c(k) + B_c u_c(k) + h_c$$

$$y_c(k) = f_y(x_c(k), u_c(k), x_d) = C_c x_c(k) + D_c u_c(k) + g_c$$

$$y_d(k) = f_d(x_d, u_d(k)) = C_d x_d + D_d u_d(k) + g_d$$

where $i \in \{1, 2, \ldots, N_d\}$ and $A_c \in \mathbb{R}^{n_x \times n_x}$, $B_c \in \mathbb{R}^{n_x \times n_c}$, $h_c \in \mathbb{R}^{n_x}$, $C_c \in \mathbb{R}^{n_y \times n_x}$, $D_c \in \mathbb{R}^{n_y \times n_c}$, $R_c \in \mathbb{R}^{n_y \times n_y}$, $G_c \in \mathbb{R}^{n_y \times n_y}$, $D_d \in \mathbb{R}^{n_y \times n_d}$, $g_d \in \mathbb{R}^{n_y}$, $k \in \mathbb{Z}_{\geq 0}$ is the discrete-time indicator, $x_c(k) \in X_c$ is the continuous state vector, $u_c(k) \in U_c$ is the exogenous discrete input vector, $u_d(k) \in U_d$ is the exogenous discrete input vector, $y_d(k) \in Y_d$ is the discrete output vector, and $y_c(k) \in Y_c$ is the continuous output vector.

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

$$Inv(x_d) = \{(x_c(k), u_c(k), u_d(k)) \in X_c \times U_c \times U_d \mid H_i x_c(k) + J_i u_c(k) \leq K_i, \tilde{H}_i x_c(k) + \tilde{J}_i u_c(k) < \tilde{K}_i \land u_d(k) = u_d\}$$

where $H_i$, $J_i$, $K_i$, $\tilde{H}_i$, $\tilde{J}_i$, $\tilde{K}_i$ 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:

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

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

$$G_c(e_{ij}) = \{u_d(k) \in U_d \mid u_d(k) = u_{d_i}\}$$

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:

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

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

6
Remark 2. In Def. 1, in a DHA, each transition edge from discrete state \(x_d\) to \(x_d\) whose corresponding reset dynamic is equal to the assigned continuous dynamic at the incoming discrete state \(x_d\), as in Eq. (13), is called as a transition edge without a reset dynamic.

\[
A_{c_{ij}}x_c(k) + B_{c_{ij}}u_c(k) + h_{c_{ij}} = A_{c_{ij}}x_c(k) + B_{c_{ij}}u_c(k) + h_{c_{ij}}
\]  

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

Remark 3. In Figure 2, the self-loop edges without reset dynamics are not shown in the directed graph of the DHA. According to the fact that edge \(e_{ij}\) is a controlled or uncontrolled edge, the symbol \(G_e(e_{ij})/G_w(e_{ij})\) is used to denote the respective controlled or uncontrolled switching guard in the DHA graph. Accordingly, \(u_d(k) \in G_e(e_{ij})/(x_c(k), u_c(k)) \in G_w(e_{ij})\) 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 satisfy 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 preceding discussion. Obviously, since any edge in the DHA graph can only be controlled or uncontrolled type, from \(u_d(k) \in G_e(e_{ij})/(x_c(k), u_c(k)) \in G_w(e_{ij})\), we mean that only one of the conditions \(u_d(k) \in G_e(e_{ij})\) or \((x_c(k), u_c(k)) \in G_w(e_{ij})\) are 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 the Def. 4.

Definition 4. 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 \((K, x_c(\cdot) : K \rightarrow X_c, x_d(\cdot) : K \rightarrow X_d, u(\cdot) : K \rightarrow U_c, y(\cdot) : K \rightarrow Y_c)\) satisfying the following items:

- **Initialization:** Let us consider discrete state \(x_d\) of a DHA graph which 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 \(k = 0\), \((x_c(0), x_d) \in \text{Init}\) where \(i \in \{1, \ldots, N_d\}\).

- **Discrete state evolution:**
  1. A transition from \(x_d(k) = x_d\) to \(x_d(k + 1) = x_d\) where \((x_d, x_d) \in E_d \cup E_c\) may occur at the instant \(k + 1\) if and only if \((x_c(k), u_c(k), u_d(k)) \in \text{Inv}(x_d)\) and \((x_c(k), u_c(k)) \in G_w(e_{ij})\) \(u_d(k) \in G_e(e_{ij})\) (see Rem. 3). In other words, there is a choice between further staying in node \(i\) or a discrete transition.
  2. A transition from \(x_d(k) = x_d\) to \(x_d(k + 1) = x_d\) where \((x_d, x_d) \in E_c \cup E_w\) must occur at the instant \(k + 1\) if and only if \((x_c(k), u_c(k), u_d(k)) \notin \text{Inv}(x_d)\) and \((x_c(k), u_c(k)) \in G_w(e_{ij})/u_d(k) \in G_e(e_{ij})\) (see Rem. 3).

- **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, by which the continuous dynamics are selected. At each step time \(k\) and in the backward evolution, we use \(x_d(k)\) and \(x_d(k + 1)\), while in the forward evolution, \(x_d(k)\) and \(x_d(k + 1)\) are employed. A detailed discussion is presented by the following algorithms.

Definition 5. 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. \(G_e(e_{ij})/G_w(e_{ij})\). Given \((x_c(k), x_d) \in X_c \times X_d\) and \(u(k) = [u_c(k)]^T, u_d(k)]^T\) 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 \(x_c(-1) = x_c(0)\), compare \(x_d(k + 1) = x_d\) with \(x_d(k) = x_d\):

If \(x_d(k + 1) = x_d(k)\), \((x_c(k), u_c(k), u_d(k)) \in \text{Inv}(x_d)\) and \(u_d(k) \notin G_w(e_{ij})/(x_c(k), u_c(k)) \notin G_w(e_{ij})\), i.e., \(e_{ij}\) is inactivated, then compute \(x_c(k + 1)\) by the continuous dynamic associated to the node \(i\), namely:

\[
x_c(k + 1) = A_{c_{ij}}x_c(k) + B_{c_{ij}}u_c(k) + h_{c_{ij}}
\]
is defined as \( x(k + 1) = A_{v_c}x_c(k) + B_{v_c}u_c(k) + h_{v_c} \) through the reset dynamic associated to the edge \( e_{h_i} \), i.e.: 

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

\[
x_c(k + 1) = A_{v_c}x_c(k) + B_{v_c}u_c(k) + h_{v_c}
\]

Remark 6. If all edges incoming to the discrete state \( x_d \) are without a reset dynamic (see Rem. 2), then \( x_d = x_d \) and \( x_d \neq x_d \) in Def. 5 become superfluous and only \( x_d(k + 1) = x_d(k) \) decides for \( x_c(k + 1) \), namely, \( x_c(k + 1) = A_{v_c}x_c(k) + B_{v_c}u_c(k) + h_{v_c} \). This type of backward evolution has been used in some literature such as [56, 57, 58, 59].

Definition 7. In a forward evolution, the 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., \( G_c(e_{u_j}) \) or \( G_w(e_{u_j}) \). Note that \( x_d(k + 1) = x_d \) is available at instant \( k \) according to the occurrence of \( e_{u_j} = (x_{d_i}, x_{d_f}) \in E_c \cup E_w \) (See Def. 4, discrete state evolution item). The continuous state \( x_c(k + 1) \) is determined by the following algorithm:

Compare \( x_d(k) = x_d \) with \( x_d(k + 1) = x_d \):

If \( x_d(k) = x_d(k + 1) \) and \( u_d(k) \notin G_c(e_{h_i})/(x_c(k), u_c(k)) \in G_w(e_{h_i}) \), i.e., \( e_{h_i} \) is inactivated, then compute \( x_c(k + 1) \) by the continuous dynamic associated to the node \( i \), namely:

\[
x_c(k + 1) = A_{v_c}x_c(k) + B_{v_c}u_c(k) + h_{v_c}
\]

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

\[
x_c(k + 1) = A_{v_c}x_c(k) + B_{v_c}u_c(k) + h_{v_c}
\]

Remark 8. If all edges outgoing from the discrete state \( x_d \) are without a reset dynamic (based on Rem. 2) then \( x_d = x_d \) and \( x_d \neq x_d \) in Def. 7 become redundant and only \( x_d(k + 1) = x_d \) decides for \( x_c(k + 1) \), namely, \( x_c(k + 1) = A_{v_c}x_c(k) + B_{v_c}u_c(k) + h_{v_c} \). This type of forward evolution has been used in some 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 the Def. 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 the set of variables that the knowledge of them at some time together with the future inputs is sufficient to allow the determination of the system future behavior.

Definition 9. In the DHA with backward evolution, according to Def. 5, knowing \( x_d(k) \), \( x_c(k) \) and \( u(k) \) are 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 Def. 7, the state of the DHA with forward evolution, is defined as \( x(k) = [x_c(k)^T, x_d(k)^T]^T \).
Remark 10. 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 Rem. 6, 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. It is the case that 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 initial condition $(x_c(0), x_d(0)) \in \text{Init}$. $E_{DHA}$ is used to denote the union of $E_{DHA}(x_c(0), x_d(0))$ over all $(x_c(0), x_d(0)) \in \text{Init}$. The set Reach$_{DHA}$ in Eq. (14) is defined as the set of all reachable states and outputs with their corresponding inputs.

Reach$_{DHA} = \{(\tilde{x}_c(k), \tilde{x}_d(k), \tilde{u}_c(k), \tilde{u}_d(k), \tilde{y}_c(k), \tilde{y}_d(k)) \in X_c \times X_d \times U_c \times U_d \times Y_c \times Y_d : \exists k_0 \in \mathbb{Z}_{\geq 0} \text{ such that } \exists (K, x_c(), x_d(), u_c(), u_d(), y_c(), y_d()) \in E_{DHA} \text{ such that } (x_c(k_0), x_d(k_0), u_c(k_0), u_d(k_0), y_c(k_0), y_d(k_0)) = (\tilde{x}_c(k), \tilde{x}_d(k), \tilde{u}_c(k), \tilde{u}_d(k), \tilde{y}_c(k), \tilde{y}_d(k)) \}$  \hspace{1cm} (14)

Remark 11. 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 that situation, the switching between the discrete states is made one sampling step earlier. In other words, a switching occurs before the related guarded or switching boundary is actually crossed. An uncontrolled guard between nodes $i$ and $j$ of the DHA is crossed at instant $k$ if $(x_c(k), u_c(k)) \in \text{G}_u(e_{ij})$. The switching condition is a polyhedral partition in the continuous state space as:

$G_u(e_{ij}) = \{x_c^N(k) \in X_c | H_{ij} x_c^N(k) \leq K_{ij}, H_{ij} x_c^N(k) < \tilde{K}_{ij} \}$ \hspace{1cm} (15)

where $x_c^N(k) = x_c(k+1) = A_c x_c(k) + B_c u_c(k) + h_c$, is one step predicted value for the continuous state at instant $k$. If in these conditions, the continuous dynamic at node $i$ is independent of the continuous inputs, i.e., $x_c(k+1) = A_c x_c(k) + h_c$, 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:

$G_u(e_{ij}) = \{x_c^N(k) \in X_c | H_{ij} x_c^N(k) \leq K_{ij}, H_{ij} x_c^N(k) < \tilde{K}_{ij} \}$ \hspace{1cm} (16)

where $x_c^N(k) = x_c(k+N) = A_c^N x_c(k) + (\sum_{i=1}^{N-1} A_c^i) h_c$, is $N$ step predicted value for the continuous state at instant $k$ and is obtained by recursive implementation of $x_c(k+1) = A_c x_c(k) + h_c$.

Remark 12. Based on the preceding discussions, the combination of the predictive reset mentioned in Rem. 11 and the backward evolution in Def. 5 is called as backward predictive reset. On the other hand, the forward predictive reset is the combination of the predictive reset in Rem. 11 and the forward evolution in Def. 7. In cases in which the multi-step predictive reset is impossible in a DHA, then the forward predictive reset has the advantage of one-step prediction more with respect to the backward predictive reset, and therefore, 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 [62, 35, 23, 63, 64] for more details and numerical examples.

Def. 13 represents the concept of the well-posedness property of a hybrid system model which is of particular importance for its usability.

Definition 13. A DHA is well-posed on $X_c \times X_d$, $U_c \times U_d$, $Y_c \times Y_d$, if $\forall x(0) = [x_c(0)^T, x_d(0)^T]^T$, $(x_c(0), x_d(0)) \in \text{Init}$, 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 Def. 13, is for the forward evolution. In backward evolution, according to Def. 9, one can use the augmented vectors $x(0) = [x_i(0)^T, x_d(0)^T, x_d(-1)^T]^T$ and $x(k) = [x_i(k)^T, x_d(k)^T, x_d(k-1)^T]^T$ where $x_d(-1)^T = x_d(0)^T$ (see Def. 5).

Theorem 14 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:

$$CInv_i = \{ (x_i(k), u_i(k)) \in X_c \times U_c : (x_i(k), u_i(k), u_d(k)) \in Inv(x_d) \}$$

$$DInv_i = \{ u_d(k) \in U_d((x_i(k), u_i(k), u_d(k)) \in Inv(x_d)) \}$$

$$C_i = \{ (x_i(k), u_i(k), u_d(k)) \in X_c \times U_c \times U_d : u_d(k) \in \bigcup_j G_c(e_{ij}) \cap (x_i(k), u_i(k)) \in \bigcup_j G_u(e_{ij}), j \neq i \}$$  \quad (17)

$$\forall i, j, k \in \{1, \ldots, N_d\} \text{ such that } e_{ij} \in E_c \text{ and } e_{ik} \in E_{uc}. \text{ The sets } CInv_i \text{ and } DInv_i \text{ determine the continuous and discrete parts of } Inv(x_d), \text{ respectively. The set } C_i \text{ determines all the values of } (x_i(k), u_i(k), u_d(k)) \text{ for which at least one of the controlled and uncontrolled outgoing edges of the node } i \text{ to different discrete state successors are simultaneously enabled. Furthermore, the sets } CInv_i', DInv_i' \text{ are the complement sets of } CInv_i \text{ and } DInv_i, \text{ respectively. The set } Reach_{DHA}^{X,U,U_d} \text{ is defined as:}$$

$$Reach_{DHA}^{X,U,U_d} = \{(\tilde{x}_i(k), \tilde{u}_i(k), \tilde{u}_d(k)) \in X_c \times U_c \times U_d : \exists \tilde{x}_d(k) \in X_d, \exists \tilde{y}_d(k) \in Y_c, \exists \tilde{y}_d(k) \in Y_d$$

such that $(\tilde{x}_i(k), \tilde{x}_d(k), \tilde{u}_i(k), \tilde{u}_d(k), \tilde{y}_d(k)) \in Reach_{DHA}$

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

Theorem 14. The DHA of Def. 1 is well-posed if the following conditions are satisfied.

(a) $X_c = \mathbb{R}^{n_c}$, $Y_c = \mathbb{R}^{n_c}$.
(b) $C_{d} = 0_{p_{x,c}d}$, $D_{d} = 0_{p_{x,uc}d}$, $g_{d} \in Y_{d}$.
(c) $CInv_i = \bigcup_j G_{uc}(e_{ij}), \forall i, j \in \{1, \ldots, N_d\}, e_{ij} \in E_{uc}$
(d) $DInv_i = \bigcup_j G_{uc}(e_{ij}), \forall i, j \in \{1, \ldots, N_d\}, e_{ij} \in E_{c}$
(e) $G_{uc}(e_{ij}) \cap G_{uc}(e_{ik}) = \emptyset, \forall i, j, k \in \{1, \ldots, N_d\} \text{ such that } j \neq k \text{ and } e_{ij}, e_{ik} \in E_{uc}$
(f) $G_{uc}(e_{ij}) \cap G_{uc}(e_{ik}) = 0, \forall i, j, k \in \{1, \ldots, N_d\} \text{ such that } j \neq k \text{ and } e_{ij}, e_{ik} \in E_{uc}$
(g) $C_i = \emptyset, \forall i \in \{1, \ldots, N_d\}$

Proof. Condition (a) provides an invariant property for continuous state $x_i(k)$ and continuous output $y_c(k)$, and implies that over infinite time horizon they remain in their sets when they start from them. 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 $(x_i(0), x_d(0)) \in Init$ such that $x_d(0) = x_d$. The continuous state $x_i(k)$, continuous and discrete outputs, $y_c(k)$ and $y_d(k)$ of the DHA are extended uniquely by Eqs. (6)-(8) until at instant $k$, $(x_i(k), u_i(k), u_d(k)) \in Inv(x_d) \cap Reach_{DHA}^{X,U,U_d}$. This implies that $(x_i(k), u_i(k), u_d(k)) \notin Inv(x_d)$. Therefore, one can conclude that $(x_i(k), u_i(k), u_d(k)) \notin ClInv_i$ or $u_d(k) \notin DInv_i$. Equivalently, this implies that $(x_i(k), u_i(k)) \in ClInv_i'$ or $u_d(k) \in DInv_i'$. In case 1, if $(x_i(k), u_i(k)) \in ClInv_i'$, then according to (c), one can conclude that $(x_i(k), u_i(k)) \in \bigcup_j G_{uc}(e_{ij})$. This and (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 $x_d$ to $x_d$. Therefore, the continuous state trajectory of the system $x_i(k)$ 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)-(8). In case 2, if $u_d(k) \in DInv_i'$ then according to (d) one can conclude that $u_d(k) \in \bigcup_j G_{uc}(e_{ij})$. This and (f) imply only one of the outgoing controlled edges $e_{ij} \in E_c$ is enabled, and the discrete state is changed from $x_d$ to $x_d$ and $x_i(k)$ is evolved by the reset dynamic in Eq. (12) in the forward
evolution or by Eq. (6) in the backward evolution uniquely. Again, $y_i(k)$ and $y_d(k)$ change uniquely via Eqs. (7)-(8). Case 3 is when both $(x_i(k), u_i(k)) \in DInv_i^c$ and $u_d(k) \in DInv_d^c$ are simultaneously valid. But (g) prevents this case from appearing. This can be shown by a contradiction: Let $(x_i(k), u_i(k)) \in DInv_i^c$ and $u_d(k) \in DInv_d^c$. Then according to (a) and (b), one can conclude that $(x_i(k), u_i(k)) \in \bigcup_i G_{in}(e_{ij})$ and $u_d(k) \in \bigcup_j G_{en}(e_{ij})$. This means that there is a possibility of simultaneous controlled and uncontrolled transition from node $i$ to different discrete state successor. Equivalently, this implies that $\exists i \ni (x_i(k), u_i(k), u_d(k)) \in C_i$. 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 $x_i(k)$ 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 (6) associated to the node $i$ of the DHA graph in backward evolution. Besides, $y_i(k)$ and $y_d(k)$ change uniquely via Eqs. (7)-(8). Consequently, the DHA is well posed.

Note that the reverse statement does not hold, since in general, a well-posed DHA may be defined on non-satisfying conditions (a) – (g) of Theorem 14. Moreover, the 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 it has been proven that the DHA are equivalent to MLD systems [35, 23], one can conclude that the proposed well-posedness conditions in Theorem 14 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 are described by [35, 65, 39, 43, 36]:

$$
\begin{align*}
\dot{x}(k+1) &= \bar{A}_x x(k) + \bar{B}_x u(k) + \bar{f}_i \\
y(k) &= \bar{C}_x x(k) + \bar{D}_x u(k) + \bar{g}_i
\end{align*}
$$

(20)

where $x(k) = [x_i(k)^T, x_d(k)^T]^T$ is the state, $u(k) = [u_i(k)^T, u_d(k)^T]^T$ is the input and $y(k) = [y_i(k)^T, y_d(k)^T]^T$ is the output. The state, input and output are partitioned in the continuous components $x_i(k) \in X_i \subseteq \mathbb{R}^{n_i}, u_i(k) \in U_i \subseteq \mathbb{R}^{m_i}, y_i(k) \in Y_i \subseteq \mathbb{R}^{p_i}$ 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 $(\bar{A}_i, \bar{B}_i, \bar{f}_i, \bar{C}_i, \bar{D}_i, \bar{g}_i)$, $i = 1, \cdots, \tilde{s}$ is defined on a cell $\Omega_i \subset X_i \times X_d \times U_i \times U_d$, $i = 1, \cdots, \tilde{s}$ which is a (not necessarily closed) polyhedron on the state-input space defined by a system of inequalities as

$$
\Omega_i = \{(x(k), u(k)) | \bar{H}_{x,i} x(k) + \bar{H}_{u,i} u(k) \leq \bar{k}_i, \bar{H}_{y,i} x(k) + \bar{H}_{u,i} u(k) < \bar{k}_i\}
$$

(21)

where $\bar{H}_{x,i}, \bar{H}_{u,i}, \bar{H}_{y,i}$, $\bar{H}_{x,i}$ and $\bar{k}_i$ are matrices with suitable dimensions. Moreover, $\bar{A}_i, \bar{B}_i, \bar{f}_i, \bar{C}_i, \bar{D}_i, \bar{g}_i$ are real matrices of suitable dimensions that define the affine dynamics for all $i$. For PWA systems in Eq. (20), the well-posedness concept is defined as in Def. 13. A sufficient condition for the PWA system (20) to be well-posed is given by Lemma 15.

**Lemma 15.** The PWA system in (20) is well-posed on $\Omega \triangleq X_i \times X_d \times U_i \times U_d$ if the following conditions are satisfied:

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

$$
\begin{align*}
\bar{A}_i &= \begin{bmatrix} \bar{A}_{i1} & \bar{A}_{i2} \\
0_{n_1 \times n_2} & 0_{n_2 \times n_d} \end{bmatrix}, & \bar{B}_i &= \begin{bmatrix} \bar{B}_{i1} & \bar{B}_{i2} \\
0_{n_2 \times m_1} & 0_{n_2 \times m_d} \end{bmatrix}, & \bar{f}_i &= \begin{bmatrix} \bar{f}_{i1} \\
\bar{f}_{i2} \end{bmatrix}, \\
\bar{C}_i &= \begin{bmatrix} \bar{C}_{i1} & \bar{C}_{i2} \end{bmatrix}, & \bar{D}_i &= \begin{bmatrix} \bar{D}_{i1} & \bar{D}_{i2} \end{bmatrix}, & \bar{g}_i &= \begin{bmatrix} \bar{g}_{i1} \\
\bar{g}_{i2} \end{bmatrix},
\end{align*}
$$

(22)

where $\bar{f}_{i2} \in X_d, \bar{g}_{i2} \in Y_d, \bar{A}_{i1} \in \mathbb{R}^{n_1 \times n_2}, \bar{A}_{i2} \in \mathbb{R}^{n_2 \times n_d}, \bar{B}_{i1} \in \mathbb{R}^{n_1 \times m_1}, \bar{B}_{i2} \in \mathbb{R}^{n_2 \times m_d}, \bar{f}_{i1} \in \mathbb{R}^{n_1}, \bar{C}_{i1} \in \mathbb{R}^{p_1 \times n_2}, \bar{C}_{i2} \in \mathbb{R}^{p_2 \times n_d}, \bar{g}_{i1} \in \mathbb{R}^{p_1}, \bar{C}_{i1} \in \mathbb{R}^{p_1 \times n_2}, \bar{D}_{i1} \in \mathbb{R}^{p_1 \times m_1}, \bar{D}_{i2} \in \mathbb{R}^{p_2 \times m_d}.$

(b) $\forall i, X_i = \mathbb{R}^{n_i}, Y_i = \mathbb{R}^{p_i}$.

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

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

11
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 on their discrete sets \( X_d, Y_d \), respectively. Condition (b) provides the invariance property for continuous state \( x_c(k) \) and output \( y_c(k) \), namely, 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 \( (x(k), u(k)) \in \Omega \), the state and output trajectory \([x(k)^T, y(k)^T]^T\) of the PWA system is extended uniquely by the corresponding affine subsystem \((\tilde{A}_c, \tilde{B}_c, \tilde{f}_c, \tilde{C}_c, \tilde{D}_c, \tilde{g}_c)\) in (20). Condition (d) guarantees the continuous and discrete state evolutions over infinite time horizon and avoid existing some regions in the state-space that do not belong to any region \( \tilde{\Omega} \). In other words, it implies that the union of the sets \( \tilde{\Omega} \) span the whole state space, i.e., \( \Omega \triangleq \bigcup_{i=1}^{s} \tilde{\Omega}_i = \mathbb{X}_c \times \mathbb{X}_d \times \mathbb{U}_c \times \mathbb{U}_d \) where \( \mathbb{X}_c = \mathbb{R}^{n_c} \) according to condition (b). Therefore, by 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 14, regarding the well-posedness of the DHA, the given conditions in Lemma 15 for the well-posedness of the PWA systems are not necessary. In other words, a well-posed PWA system may be defined while the conditions (a)-(d) are not satisfied. It is noted that in the continuous-time domain there are some 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 to the elimination of the discrete states and their corresponding dynamics, continuous and discrete control inputs, controlled switching phenomena, continuous and discrete output signals.

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

**Definition 16.** A PWA system in the logic canonical form (PWA-LC system) is described by the state-space equations

\[
\begin{align*}
\begin{cases}
\dot{x}(k+1) &= A_i x(k) + B_i u(k) + f_i \\
y(k) &= C_i x(k) + D_i u(k) + g_i
\end{cases}
\text{for } (x(k), u(k)) \in \Omega_i
\end{align*}
\]

where \( i = 1, \ldots, s \),

\[
\Omega_i = \{( (x(k), u(k)) | H_{x_i} x_c(k) + H_{u_i} u_c(k) \leq k_i, \tilde{H}_{x_i} x_c(k) + \tilde{H}_{u_i} u_c(k) < \tilde{k}_i, u_c(k) = u_d(k), x_d(k) = x_d \}
\]

\[
A_i = \begin{bmatrix}
A_{c_i} & 0_{n_c \times n_d} \\
0_{n_d \times n_c} & A_{d_i}
\end{bmatrix}, \quad
B_i = \begin{bmatrix}
B_{c_i} & 0_{n_c \times n_m} \\
0_{n_m \times n_c} & B_{d_i}
\end{bmatrix}, \quad
f_i = \begin{bmatrix}
f_{c_i} \\
0_{n_d}
\end{bmatrix},
\]

\[
C_i = \begin{bmatrix}
C_{c_i} & 0_{n_c \times n_d} \\
0_{n_d \times n_c} & C_{d_i}
\end{bmatrix}, \quad
D_i = \begin{bmatrix}
D_{c_i} & 0_{n_c \times n_m} \\
0_{n_m \times n_c} & D_{d_i}
\end{bmatrix}, \quad
g_i = \begin{bmatrix}
g_{c_i} \\
0_{n_d}
\end{bmatrix},
\]

\( f_{d_i} \in \{0, 1\}^{n_d}, g_{d_i} \in \{0, 1\}^{n_d}, A_{c_i} \in \mathbb{R}^{n_c \times n_c}, B_{c_i} \in \mathbb{R}^{n_c \times n_m}, f_{c_i} \in \mathbb{R}^{n_d}, C_{c_i} \in \mathbb{R}^{n_c \times n_d}, D_{c_i} \in \mathbb{R}^{n_d \times n_m}, g_{c_i} \in \mathbb{R}^{n_d}. \)

Similar to Lemma 15, a set of sufficient conditions can be developed for well-posedness of PWA-LC systems via Lemma 17.

**Lemma 17.** The PWA-LC system in (23) is well-posed on \( \Omega \triangleq \mathbb{X}_c \times \mathbb{X}_d \times \mathbb{U}_c \times \mathbb{U}_d \) if the following conditions are satisfied:

(a) \( f_{d_i} \in \mathbb{X}_d, g_{d_i} \in \mathbb{Y}_d \).

(b) \( \mathbb{X}_c = \mathbb{R}^{n_c}, \mathbb{Y}_c = \mathbb{R}^{n_c} \).

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

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

**Proof.** The proof of this Lemma is similar to the proof of Lemma 15 and is omitted for the sake of brevity. \( \square \)
Compared to PWA systems in (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 \(x_i(k)\) and \(y_i(k)\) is not influenced by the discrete states \(x_d(k)\) and inputs \(u_d(k)\). In fact, the discrete input \(u_d(k)\) contributes only for the switching between different subsystems. The second feature is that when index \(i\) is fixed, according to condition (24), \(x_d(k)\) and \(u_d(k)\) are constant within a cell \(\Omega_i\), and \(x_d(k + 1)\) and \(y_d(k)\) are determined by only \(f_d\) and \(g_d\).

In the next section, we show how an automaton-based DHA can be translated to its equivalent PWA and PWA-LC forms. In PWA-LC systems the discrete variables influence the switching between sub-models but not on the continuous dynamics. This is an important feature of 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 piecewise affine systems

In this section, the relationship between the proposed discrete time hybrid automaton in Def. 1 with the class of PWA and PWA-LC systems presented in subsection 2.3 is discussed.

**Definition 18.** Let \(\Sigma_1\) and \(\Sigma_2\) 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 \(x_2(k) \in X_2\), \(k \in \mathbb{Z}_{\geq 0}\). The hybrid models \(\Sigma_1\) and \(\Sigma_2\) 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 \(k \in \mathbb{Z}_{\geq 0}\).

**Definition 19.** The PWA systems obtained from the translation of a DHA with the backward or forward evolution are called as backward or forward PWA (BPWA or FPWA) systems, respectively. The mathematical representation of BPWA and FPWA systems are similar to 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 Def. 1 is translated to its equivalent PWA system.

**Definition 20.** The PWA-LC systems obtained from the translation of a DHA with the backward or forward evolution are called as backward or forward PWA-LC (BPWA-LC or FPWA-LC) systems, respectively. The mathematical representation of BPWA-LC and FPWA-LC systems are similar to the general PWA-LC systems in Eq. (23).

Lemma 21 represents a constructive approach to convert a well-posed automaton-based DHA to its equivalent PWA and PWA-LC forms. It also plays a key role to evaluate the complexity of the proposed translation technique in Prop. 25 and subsection 3.2.

**Lemma 21.** A well-posed DHA in Def. 1 with defined backward and forward evolutions inDefs. 5 and 7, can be transformed to the equivalent well-posed BPWA and FPWA systems, and also to the BPWA-LC and FPWA-LC systems.

**Proof.** Let us consider discrete state \(x_d\) 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 Fig. 4.

For the brevity in the formulations, the terms in the BPWA and FPWA forms are only specified for the continuous state dynamics associated with the discrete state \(x_d\), the single in-coming and self-loop edges at this node which are depicted by the solid lines in Fig. 4. The overall BPWA and FPWA system 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 via the backward evolution in Def. 5. The BPWA form in Eq. (26) 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).

\[
\begin{bmatrix}
x_d^{new}(k)

x_d(k)

x_d(k-1)
\end{bmatrix}
\]  

(27)
Using Eq. (27), Eq. (26) can be rewritten as (28).
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 $x_d^{new}(k) = [x_d^T, x_d^T]^T \land Inv(x_d) \cup G_c(e_d)/G_w(e_d)$, all the regions are in the standard region representation of 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 $x_d^{new}(k) = [x_d^T, x_d^T]^T \land Inv(x_d)$ and $x_d^{new}(k) = [x_d^T, x_d^T]^T \land G_c(e_d)/G_w(e_d)$ 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).

Case 2: FPWA and FPWA-LC formulations: The FPWA representation is obtained via the forward evolution of the DHA (see Def. 7). The specified FPWA form in Eq. (30) corresponds to the part of the DHA graph that is depicted by the solid lines in Fig. 4.

$$
\begin{bmatrix}
    x(k + 1) \\
    y(k)
\end{bmatrix} =
\begin{bmatrix}
    x_d(k + 1) \\
    y_d(k)
\end{bmatrix} =
\begin{bmatrix}
    A_{c}, x_c(k) + B_{c}, u_c(k) + f_{c} \\
    C_{c}, x_c(k) + D_{c}, u_c(k) + g_{c} \\
    C_{d}, x_d(k) + D_{d}, u_d(k) + g_{d}
\end{bmatrix}
\begin{bmatrix}
    x_d(k) = x_d \land Inv(x_d) \\
    x_d(k) = x_d \land G_c(e_d)/G_w(e_d) \\
    x_d(k) = x_d \land G_c(e_d)/G_w(e_d)
\end{bmatrix}
$$

According to Eq. (30), one can see that the FPWA system is automatically in the form of standard PWA-LC system described by Eqs. (23)-(25). Since it is originated from the forward evolution of DHA, according to Def. 20 it is called as FPWA-LC form.

Remark 22. 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 fewer 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 try to compute their union as $\Omega_j = \bigcup_{i=1}^{n_j} \Omega_i$ or $\Omega_j = \bigcup_{j=1}^{n_i} \Omega_j$ where $\Omega_j$ is the $j^{th}$ region in the new compact representation, and $n_j$ 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 23. Based on Rem. 22, 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 as independent regions.

According to Rem. 22, in some special cases that the incoming edge $e_{ii} \in E$ in Fig. 4 is without reset dynamic
Furthermore, if the discrete state $x_d$ is without self-loop edge $e_d \in E$ (see Rem. 3), then the BPWA representation in

\[
\begin{bmatrix}
  x(k + 1) \\
  y(k)
\end{bmatrix} = 
\begin{bmatrix}
  x_d(k + 1) \\
  x_d^{\text{new}}(k + 1) \\
  y_d(k)
\end{bmatrix} = 
\begin{bmatrix}
  A_c x_c(k) + B_c u_c(k) + f_c \\
  C_c x_c(k) + D_c u_c(k) + g_c \\
  C_d x_d(k) + D_d u_d(k) + g_d
\end{bmatrix} \\
\begin{bmatrix}
  x_d^{\text{new}}(k) = [x_d^T, x_d^{\text{new}}]^T \land \text{Inv}(x_d) \\
  \cup (x_d^{\text{new}}(k) = [x_d^T, x_d^{\text{new}}]^T \land \text{Inv}(x_d) \cup G_i(e_i)/G_{\text{in}}(e_i))
\end{bmatrix}
\]

Furthermore, if the discrete state $x_d$ is without self-loop edge $e_d \in E$ (See Rem. 3), then the BPWA representation in
Eq. (31) can be simplified more as in Eq. (32).

\[
\begin{bmatrix}
  x(k+1) \\
  y(k)
\end{bmatrix} =
\begin{bmatrix}
  x_c(k+1) \\
  x_d(k+1) \\
  y_c(k) \\
  y_d(k)
\end{bmatrix} =
\begin{bmatrix}
  A_c x_c(k) + B_c u_c(k) + f_c \\
  C_c x_c(k) + D_c u_c(k) + g_c \\
  C_d x_d(k) + D_d u_d(k) + g_d
\end{bmatrix}
\]

\[
\begin{bmatrix}
  x_d^{new}(k) \\
  y_d^{new}(k)
\end{bmatrix} =
\begin{bmatrix}
  x_d^{new}(k) = [x_d^T, x_d^T]^T \land \text{Inv}(x_d) \\
  (x_d^{new})(k) = [x_d^T, x_d^T]^T \land \text{Inv}(x_d)
\end{bmatrix}
\]

\[
\begin{bmatrix}
  A_c x_c(k) + B_c u_c(k) + f_c \\
  C_c x_c(k) + D_c u_c(k) + g_c \\
  C_d x_d(k) + D_d u_d(k) + g_d
\end{bmatrix}
\]

\[
\begin{bmatrix}
  x_d^{new}(k) = [x_d^T, x_d^T]^T \land (G_i(e_i))/G_m(e_i) \\
  (x_d^{new})(k) = [x_d^T, x_d^T]^T \land G_i(e_i)/G_m(e_i)
\end{bmatrix}
\]

In contrast to the backward evolution case in Eq. (31), when the incoming edge \( e_{hi} \in E \) in Fig. 4 is without a reset dynamic (see Rem. 2), the number of regions in Eq. (30) cannot be reduced further. This is because as it is shown in Eq. (33), although some regions share the same continuous and discrete state dynamics, however, the continuous and discrete output dynamics are different in these regions.

\[
\begin{bmatrix}
  x(k+1) \\
  y(k)
\end{bmatrix} =
\begin{bmatrix}
  x_c(k+1) \\
  x_d(k+1) \\
  y_c(k) \\
  y_d(k)
\end{bmatrix} =
\begin{bmatrix}
  A_c x_c(k) + B_c u_c(k) + f_c \\
  C_c x_c(k) + D_c u_c(k) + g_c \\
  C_d x_d(k) + D_d u_d(k) + g_d
\end{bmatrix}
\]

\[
\begin{bmatrix}
  x_d(k) = x_d \land \text{Inv}(x_d) \\
  x_d(k) = x_d \land G_i(e_i)/G_m(e_i)
\end{bmatrix}
\]

\[
\begin{bmatrix}
  A_c x_c(k) + B_c u_c(k) + f_c \\
  C_c x_c(k) + D_c u_c(k) + g_c \\
  C_d x_d(k) + D_d u_d(k) + g_d
\end{bmatrix}
\]

\[
\begin{bmatrix}
  x_d(k) = x_d \land G_i(e_i)/G_m(e_i) \\
  x_d(k) = x_d \land (G_i(e_i))/G_m(e_i)
\end{bmatrix}
\]

\[
\begin{bmatrix}
  A_c x_c(k) + B_c u_c(k) + f_c \\
  C_c x_c(k) + D_c u_c(k) + g_c \\
  C_d x_d(k) + D_d u_d(k) + g_d
\end{bmatrix}
\]

\[
\begin{bmatrix}
  x_d(k) = x_d \land G_i(e_i)/G_m(e_i) \\
  x_d(k) = x_d \land (G_i(e_i))/G_m(e_i)
\end{bmatrix}
\]
Remark 24. In special cases that 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, then in the case that the incoming edge $e_{hi} \in E$ in Fig. 4 is without a reset dynamic (see Rem. 2), the FPWA form in Eq. (30) can be simplified further as in Eq. (34) (see Rem. 22).

As it can be seen from Eq. (26), when the edge $e_{ij}$ is activated, the discrete state of the system is changed from $x_{di}$ to $x_{dj}$. However, the continuous state of the system still evolves through continuous dynamic in discrete state $x_{dj}$ or the reset dynamic corresponding to the edge between $x_{di}$ and $x_{dj}$. 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 named a Backward PWA (BPWA) system which is the result of the backward evolution of the original DHA. Another approach is that the switching among continuous dynamics of the system is synchronized with the change of the system discrete state. As it can be seen from Eq. (30), when a switching event occurs from discrete state $x_{di}$ to discrete state $x_{dj}$ at the sampling step $k$, the continuous state of the system is also evolved with continuous dynamic associated with the reset dynamic of the transition edge from $x_{di}$ to $x_{dj}$, i.e., $R_{x_{dj}}$. 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 a Forward PWA (FPWA).

Furthermore, if the discrete state $x_{di}$ is without a self-loop edge $e_{ii} \in E$ (See Rem. 3), then the FPWA representation in Eq. (33) can be simplified more as in Eq. (35).

As it can be seen from Eq. (26), when the edge $e_{ij}$ is activated, the discrete state of the system is changed from $x_{di}$ to $x_{dj}$. However, the continuous state of the system still evolves through continuous dynamic in discrete state $x_{dj}$ or the reset dynamic corresponding to the edge between $x_{di}$ and $x_{dj}$. 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 named a Backward PWA (BPWA) system which is the result of the backward evolution of the original DHA. Another approach is that the switching among continuous dynamics of the system is synchronized with the change of the system discrete state. As it can be seen from Eq. (30), when a switching event occurs from discrete state $x_{di}$ to discrete state $x_{dj}$ at the sampling step $k$, the continuous state of the system is also evolved with continuous dynamic associated with the reset dynamic of the transition edge from $x_{di}$ to $x_{dj}$, i.e., $R_{x_{dj}}$. 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 a Forward PWA (FPWA).

Despite of one step delay compensation in the FPWA framework, comparing Eqs. (28)-(32) with Eqs. (30)-(35) shows that the number of regions in the BPWA form is always equal 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.
• \( n_{\text{inre}}(i) \) - the number of incoming edges \( e_{hi}, h \neq i \) at discrete state \( x_d \) with reset dynamics
• \( n_{\text{out}}(i) \) - the number of outgoing edges \( e_{ij}, i \neq j \) (with or without reset) at discrete state \( x_d \)
• \( n_{\text{in}}(i) \) - the number of incoming edges \( e_{hi}, h \neq i \) (with or without reset) at discrete state \( x_d \)
• \( N_{\text{self}} \) - the total number of self-loop edges \( e_d \) with reset dynamics
• \( N_{\text{inre}} \) - the total number of transition edges \( e_{ij}, i \neq j \) with reset dynamic
• \( N_d \) - the total number of discrete states in DHA.

\[ N_{\text{BPWA}} = N_d + N_{\text{self}} + 3N_t + \sum_{i=1}^{N_d} n_{\text{in}}(i) \]  

(36)

\[ N_{\text{FPWA}} = N_{\text{self}} + N_d + N_t \]  

(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 at discrete state \( x_d \), i.e., \( x_c(k+1) = A_c x_c(k) + B_c u_i(k) + f_c \) is \( 1 + n_{\text{out}}(i) \). Thus, the total number of such regions is \( \sum_{i=1}^{N_d} (1 + n_{\text{out}}(i)) \). 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 \( x_c(k+1) = A_c x_c(k) + B_c u_i(k) + f_c \) is \( N_{\text{self}} \). Also, the number of regions identified by an incoming edge \( e_{hi}, h \neq i \), with or without reset dynamics is equal to \( 2 + n_{\text{out}}(i) \). As a result, the total number of such regions is \( \sum_{i=1}^{N_d} (2 + n_{\text{out}}(i)) n_{\text{in}}(i) \). Therefore, the number of regions in a BPWA form of a DHA can be written as:

\[ N_{\text{BPWA}} = \sum_{i=1}^{N_d} (1 + n_{\text{out}}(i)) + \sum_{i=1}^{N_d} (2 + n_{\text{out}}(i)) n_{\text{in}}(i) + N_{\text{self}} \]  

(38)

Simplifying Eq. (38) yields:

\[ N_{\text{BPWA}} = N_d + N_{\text{self}} + 3N_t + \sum_{i=1}^{N_d} n_{\text{out}}(i) n_{\text{in}}(i) \]  

(39)

According to Eq. (30), one can see that each self-loop or incoming edge of the discrete state \( x_d \) 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., \( N_{\text{FPWA}} \), can be written as:

\[ N_{\text{FPWA}} = N_{\text{self}} + \sum_{i=1}^{N_d} 1 + \sum_{i=1}^{N_t} 1 \]  

(40)

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

\[ N_{\text{FPWA}} = N_{\text{self}} + N_d + N_t \]  

(41)

The regions whose numbers calculated in Prop. 25 are of standard type as represented in Eqs. (21) and (24). According to Rem. 22, 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 region numbers. The Props. 26 and 28 quantify the number of independent regions (see Def. 23) in these compact forms of BPWA and FPWA systems.

**Proposition 26.** The number of independent regions in the compact representation of the BPWA equivalent model of a DHA is equal to \( N_{\text{BPWA}} \), such that:

\[ N_{\text{BPWA}} = N_d + N_{\text{self}} + N_{\text{inre}} + N_t + \sum_{i=1}^{N_d} n_{\text{inre}}(i) \]  

(42)
Proof. According to Eqs. (28) and (31)-(32), one can see that the number of independent regions in compact form of a BPWA model characterized by the continuous dynamic at discrete-state $x_d$ in conjunction with the incoming edges without reset dynamics is $1 + n_{\text{out}}(i)$. The reset dynamic associated with each self-loop edge determine 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., $x_i(k + 1) = A_{ch_i}x_i(k) + B_{ch_i}u_i(k) + f_{ch_i}$ is equal to $1 + n_{\text{out}}(i)$. Thus, the total number of such independent regions is $N_{\text{inre}}^d = \sum_{i=1}^{N_d} (1 + n_{\text{out}}(i))n_{\text{inre}}(i)$. As a result, the total number of independent regions in the compact form of a BPWA model can be obtained as

$$N_{\text{Rpwa}} = \sum_{i=1}^{N_d} (1 + n_{\text{out}}(i)) + \sum_{i=1}^{N_d} (1 + n_{\text{out}}(i))n_{\text{inre}}(i) + N_{\text{self}}$$  \hspace{1cm} (43)

Simplifying Eq. (43) yields

$$N_{\text{Rpwa}} = N_d + N_{\text{self}} + N_{\text{inre}} + \sum_{i=1}^{N_d} n_{\text{out}}(i)n_{\text{inre}}(i)$$  \hspace{1cm} (44)

From Props. 25 and 26, one can conclude that the number of regions in BPWA is greater than that of the FPWA with a quantity $N_{\text{inre}} + \sum_{i=1}^{N_d} (1 + n_{\text{out}}(i))n_{\text{inre}}(i)$, and as a result, FPWA is a more compact representation.

Corollary 27. In the case that the physical plants do not have reset dynamics (based on Rem. 2), the number of regions in the BPWA and FPWA systems are identical and equal to $N_d + N_c$. At these conditions, according to Rem. 10, the augmentation of discrete state vector via $x_d(k-1)$ as it was also done in the case of the equivalent BPWA form in (27), is not required.

Reset dynamics or state jumps appear in various hybrid systems for instance 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].

Prop. 28 states when the conditions of Rem. 24 hold, the number of regions in the FPWA form can be further decreased than that of Eq. (37) in Prop. 25.

Proposition 28. Let the DHA of Def. 1 is 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

$$N_{\text{Rpwa}} = N_{\text{self}} + N_d + N_{\text{inre}}$$  \hspace{1cm} (45)

Proof. According to Rem. 24 and (34), one can see that each self-loop or incoming edge of the discrete state $x_d$ with reset dynamic specifies a single region in 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

$$N_{\text{Rpwa}} = N_{\text{self}} + \sum_{i=1}^{N_d} 1 + \sum_{i=1}^{N_d} n_{\text{inre}}(i)$$  \hspace{1cm} (46)

By simplifying (46), one can write

$$N_{\text{Rpwa}} = N_{\text{self}} + N_d + N_{\text{inre}}$$  \hspace{1cm} (47)

Corollary 29. If the conditions of Prop. 28 hold, and the DHA is without reset dynamics, then the number of regions in the FPWA form is reduced to $N_d$. 

22
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 Props. 25, 26 and 28 each continuous dynamic associated with each node of the equivalent DHA appears more than once in the BPWA form (the precise number is \(n_{\text{BPWA}}(i) + 1\)), 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 in the FPWA systems. After detection 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 30.** Any well-posed BPWA or FPWA system can be transformed to the equivalent well-posed DHA based on the Def. 1.

**Proof. Case 1. Translation of the BPWA to the DHA:** According to (28), one can see that the cells in which \(x_d^\text{new}(k) = x_d^\text{new}(k + 1)\) either determine the continuous dynamics associated with each discrete state of the DHA, i.e., \(x_i(k + 1) = A_i x_i(k) + B_i u_i(k) + f_i\), and corresponding invariant set \(\text{Inv}(x_d)\) or specify the reset dynamics \(x_i(k + 1) = A_{i,\text{new}} x_i(k) + B_{i,\text{new}} u_i(k) + f_{i,\text{new}}\) and corresponding guard conditions \(G_i(e_i)/G_{\text{new}}(e_i)\) in the self-loop edges \(e_i\). 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. During coding the discrete states of the DHA, since \(x_d^\text{new}(k) = [x_d(k)^T, x_d(k - 1)^T]^T\), the binary sub-vector \(x_d(k)\) can be used for coding.

The regions of (28) in which \(x_d^\text{new}(k) \neq x_d^\text{new}(k + 1)\) 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 \(x_i(k + 1) = A_i x_i(k) + B_i u_i(k) + f_i\), which have already been determined by the cells in which \(x_d^\text{new}(k) = x_d^\text{new}(k + 1)\), and therefore, does not give new information about continuous dynamics of the DHA. However, one can use the discrete state evolution of \(x_d^\text{new}(k) \neq x_d^\text{new}(k + 1)\) 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. But, in this case, there is no enough information to characterize the respective reset dynamic. This data comes from examining of other cells in the BPWA structure, i.e., cases (b) and (c).

With this information, the DHA net of Fig. 5, is further completed as shown in Fig. 6.

(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 \(x_i(k + 1) = A_i x_i(k) + B_i u_i(k) + f_{i,\text{new}}, h \neq i\). 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 type of cells for staying in the discrete state \(x_d(k)\) are not new since they have already been used in the cells where \(x_d^\text{new}(k) = x_d^\text{new}(k + 1)\) to specify the invariant and guard conditions of type \(\text{Inv}(x_d)\) and \(G_i(e_i)/G_{\text{new}}(e_i)\), respectively. Using the information in this part, the DHA net of Fig. 6 is completed more as illustrated in Fig. 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 type of cells are of type \(x_i(k + 1) = A_i x_i(k) + B_i u_i(k) + f_{i,\text{new}}, h \neq i\). 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 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 to the DHA:** According to Eq. (30), one can see 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., \(x_i(k + 1) = A_i x_i(k) + B_i u_i(k) + f_i\), and the corresponding invariant set \(\text{Inv}(x_d)\) or specify the reset dynamics \(x_i(k + 1) = A_{i,\text{new}} x_i(k) + B_{i,\text{new}} u_i(k) + f_{i,\text{new}}\) and corresponding guard conditions \(G_i(e_i)/G_{\text{new}}(e_i)\) in the self-loop edges \(e_i\). Using these 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 Fig. 8. The cells in which \(x_d(k) \neq x_d(k + 1)\) can be employed to characterize the transition edges between discrete states \(x_d(k) = x_d(k)\) and \(x_d(k + 1) = x_d(k)\) and the relative reset dynamics and switching guards as shown in Fig. 8.

By implementing the preceding translation technique for all regions of the FPWA model of 30, one can construct the equivalent DHA model. \(\square\)
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 31. In the DHA of Def. 1 the discrete predecessor operator $D_{pre}$ is defined for the discrete state $x_d \in X_d$ as the set of discrete states $x_{d_i}$ from which $x_d$ can be reached in one step:

$$D_{pre}(x_d) = \{x_{d_i} \mid (x_{d_i}, x_d) \in E_c \cup E_{uc}\}$$

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

$$D_{post}(x_d) = \{x_{d_j} \mid (x_{d_j}, x_d) \in E_c \cup E_{uc}\}$$

According to Rem. 6, if all edges in a DHA are without reset dynamics, one can conclude that considering $x_d(k-1)$ in $x_{d^\text{curr}}(k)$ is superfluous and the selection of continuous dynamics in the DHA is made by the $x_d(k)$ 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 Rem. 3) depicted in Fig. 4 is given by Eq. (50).

$$\begin{bmatrix}
  x(k+1) \\
  y(k)
\end{bmatrix} =
\begin{bmatrix}
  x_d(k) \\
  y_d(k)
\end{bmatrix} +
\begin{bmatrix}
  x_d(k) & x_d(k) & x_d(k) & x_d(k) & \vdots & \vdots \\
  y_d(k) & y_d(k) & y_d(k) & y_d(k) & \vdots & \vdots \\
\end{bmatrix} \cdot
\begin{bmatrix}
  A_c \cdot x_d(k) + B_c \cdot u_c(k) + f_c \\
  C_c \cdot x_d(k) + D_c \cdot u_c(k) + g_c \\
  C_d \cdot x_d(k) + D_d \cdot u_d(k) + g_d \\
  A_c \cdot x_d(k) + B_c \cdot u_c(k) + f_c \\
  C_c \cdot x_d(k) + D_c \cdot u_c(k) + g_c \\
  C_d \cdot x_d(k) + D_d \cdot u_d(k) + g_d \\
  \vdots \\
  \vdots \\
  \vdots \\
\end{bmatrix}
$$

where $\{x_{d_1}, \ldots, x_{d_n}, \ldots, x_{d_m}\} \in D_{post}(x_d)$.

For the same assumption, i.e., the DHA without reset dynamic, the FPWA form corresponding to a part of a DHA...
shown in Fig. 4 is given by Eq. (51).

\[
\begin{bmatrix}
  x_{c}(k+1) \\
  x_{d}(k+1) \\
  y_{c}(k) \\
  y_{d}(k)
\end{bmatrix} = \begin{bmatrix}
  A_{c}x_{c}(k) + B_{c}u_{c}(k) + f_{ci} \\
  C_{c}x_{c}(k) + D_{c}u_{c}(k) + g_{ci} \\
  C_{d}x_{d}(k) + D_{d}u_{d}(k) + g_{di} \\
  \vdots
\end{bmatrix}
\begin{bmatrix}
  x_{c}(k) \\
  x_{d}(k) \\
  y_{c}(k) \\
  y_{d}(k)
\end{bmatrix} = x_{d} \land \text{Inv}(x_{d})
\]

\[
x_{d}(k) = x_{d} \land \text{Inv}(x_{d})
\]

Lemma 32. Let in a well-posed DHA with forward evolution and without reset dynamics, the following conditions are 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 belong to Dpre\(x_{d}\) are identical with the continuous and discrete output dynamics at node \(x_{d}\) where \(i \in \{1, \ldots, N_{d}\}\).

(b) \(\text{Inv}(x_{d}) = G_{i}(e_{hi})\) or \(\text{Inv}(x_{d}) = G_{m}(e_{hi})\) where \(h, i \in \{1, \ldots, N_{d}\}, h \neq i\) and \(x_{d} \in \text{Dpre}(x_{d})\).

(c) \(\text{Inv}(x_{d}) \cap \text{Inv}(x_{d}) = 0\), 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 Fig. 4 in which the continuous dynamic \(A_{c}x_{c}(k) + B_{c}u_{c}(k) + f_{ci}\) appears can be represented as (51).
According to condition (a), the FPWA system in Eq. (51) can be rewritten as Eq. (52).

\[
\begin{bmatrix}
  x(k + 1) \\
y(k)
\end{bmatrix}
\begin{bmatrix}
x_0(k + 1) \\
x_d(k + 1) \\
y_c(k) \\
y_d(k)
\end{bmatrix}
= \begin{bmatrix}
A_c x_c(k) + B_c u_c(k) + f_c \\
C_c x_c(k) + D_c u_c(k) + g_c \\
C_d x_d(k) + D_d u_d(k) + g_d
\end{bmatrix}
\begin{bmatrix}
x_d(k) \\
y_d(k)
\end{bmatrix}
\begin{bmatrix}
x_d(k) = x_d \land \text{Inv}(x_d) \\
\cup(x_d(k) = x_d \land G_c(e_d) / G_m(e_d)) \\
\cup(x_d(k) = x_d \land G_c(e_{d_h}) / G_m(e_{d_h})) \\
\cup(x_d(k) = x_d \land G_c(e_{d_{gh}}) / G_m(e_{d_{gh}})) \\
\cup(x_d(k) = x_d \land \text{Inv}(x_d))
\end{bmatrix}
\]

Now using condition (b), (52) can be rewritten as

\[
\begin{bmatrix}
  x(k + 1) \\
y(k)
\end{bmatrix}
\begin{bmatrix}
x_0(k + 1) \\
x_d(k + 1) \\
y_c(k) \\
y_d(k)
\end{bmatrix}
= \begin{bmatrix}
A_c x_c(k) + B_c u_c(k) + f_c \\
C_c x_c(k) + D_c u_c(k) + g_c \\
C_d x_d(k) + D_d u_d(k) + g_d
\end{bmatrix}
\begin{bmatrix}
x_d(k) \\
y_d(k)
\end{bmatrix}
\begin{bmatrix}
x_d(k) = x_d \land \text{Inv}(x_d) \\
\cup(x_d(k) = x_d \land \text{Inv}(x_d)) \\
\cup(x_d(k) = x_d \land \text{Inv}(x_d)) \\
\cup(x_d(k) = x_d \land \text{Inv}(x_d))
\end{bmatrix}
\]

The FPWA form in Eq. (53), can be represented in a more compact form of Eq. (54).

\[
\begin{bmatrix}
  x(k + 1) \\
y(k)
\end{bmatrix}
\begin{bmatrix}
x_0(k + 1) \\
x_d(k + 1) \\
y_c(k) \\
y_d(k)
\end{bmatrix}
= \begin{bmatrix}
A_c x_c(k) + B_c u_c(k) + f_c \\
C_c x_c(k) + D_c u_c(k) + g_c \\
C_d x_d(k) + D_d u_d(k) + g_d
\end{bmatrix}
\begin{bmatrix}
x_d(k) \\
y_d(k)
\end{bmatrix}
\begin{bmatrix}
x_d(k) = x_d \lor x_d(k) \in D_{\text{pre}}(x_d(k)) \land \text{Inv}(x_d) \\
\cup(x_d(k) = x_d \lor x_d(k) \in D_{\text{pre}}(x_d(k)) \land \text{Inv}(x_d))
\end{bmatrix}
\]

26
Accordingly, the overall FPWA form of the DHA can be constructed as in Eq. (55).

\[
\begin{bmatrix}
  x(k + 1) \\
  y(k)
\end{bmatrix}
= 
\begin{bmatrix}
  x_i(k + 1) \\
  y_i(k) \\
  y_d(k)
\end{bmatrix}
\]

\[
= \begin{cases}
  A_{c_i}x_i(k) + B_{c_i}u_i(k) + f_{c_i} & (x_d(k) = x_d \lor x_d(k) \in Dpre(x_d)) \land Inv(x_d) \\
  C_{c_i}x_i(k) + D_{c_i}u_i(k) + g_{c_i} & \\
  C_{d_i}x_d(k) + D_{d_i}u_d(k) + g_{d_i} & (x_d(k) = x_d \lor x_d(k) \in Dpre(x_d)) \land Inv(x_d) \\
  \vdots & \vdots \\
  A_{c_{i_j}}x_i(k) + B_{c_{i_j}}u_i(k) + f_{c_{i_j}} & (x_d(k) = x_d \lor x_d(k) \in Dpre(x_d)) \land Inv(x_d) \\
  C_{c_{i_j}}x_i(k) + D_{c_{i_j}}u_i(k) + g_{c_{i_j}} & \\
  C_{d_{i_j}}x_d(k) + D_{d_{i_j}}u_d(k) + g_{d_{i_j}} & (x_d(k) = x_d \lor x_d(k) \in Dpre(x_d)) \land Inv(x_d)
\end{cases}
\]

(55)

Let at instant \( k \), \( Inv(x_d) \) is satisfied. It is shown that the satisfaction of \( Inv(x_d) \) implies the satisfaction of \( x_d(k) = x_d \) or \( x_d(k) \in Dpre(x_d) \), and therefore, the discrete state evolution is redundant and continuous dynamic associated to the cell \( \Omega_i \) is selected only based on the satisfaction of \( Inv(x_d) \).

This can be shown by a contradiction. Let at instant \( k \), \( Inv(x_d) \) is satisfied, but \( x_d(k) \neq x_d \) and \( x_d(k) \notin Dpre(x_d) \). Since the DHA, and as a result its equivalent FPWA form in Eq. (55), are well-posed, this implies that one of the cells, say \( \Omega_j, j \neq i \) is activated. Thus, at instant \( k \), both invariant sets \( Inv(x_d) \) and \( Inv(x_d) \) are satisfied. Namely, \( \exists(x_i(k), u_i(k), u_d(k)) \in Inv(x_d) \cap Inv(x_d) \). But, this contradicts condition (c). Therefore, discrete states in FPWA form of (55) are redundant and can be eliminated from the FPWA system as in Eq. (56):

\[
\begin{bmatrix}
  x_i(k + 1) \\
  y_i(k) \\
  y_d(k)
\end{bmatrix}
= \begin{cases}
  A_{c_i}x_i(k) + B_{c_i}u_i(k) + f_{c_i} & Inv(x_d) \\
  C_{c_i}x_i(k) + D_{c_i}u_i(k) + g_{c_i} & \\
  C_{d_i}x_d(k) + D_{d_i}u_d(k) + g_{d_i} & Inv(x_d) \\
  \vdots & \vdots \\
  A_{c_{i_j}}x_i(k) + B_{c_{i_j}}u_i(k) + f_{c_{i_j}} & Inv(x_d) \\
  C_{c_{i_j}}x_i(k) + D_{c_{i_j}}u_i(k) + g_{c_{i_j}} & \\
  C_{d_{i_j}}x_d(k) + D_{d_{i_j}}u_d(k) + g_{d_{i_j}} & Inv(x_d)
\end{cases}
\]

(56)

3.2. Complexity analysis

In Subsection 2.1, the traditional DHA with 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 $\tilde{\delta}_c$, $\tilde{x}_d$ and $\tilde{u}_d$, one can obtain the following PWA system:

$$x_c(k+1) = A_{f_d}(\tilde{x}_c, \tilde{u}_c, \tilde{\delta}_c) x_c(k) + B_{f_d}(\tilde{x}_c, \tilde{u}_c, \tilde{\delta}_c) + f_{f_d}(\tilde{x}_c, \tilde{u}_c, \tilde{\delta}_c)$$

(57a)

$$x_d(k+1) = f_D(\tilde{x}_d, \tilde{u}_d, \tilde{\delta}_d)$$

(57b)

$$y_c(k) = C_{f_d}(\tilde{x}_c, \tilde{u}_c, \tilde{\delta}_c) x_c(k) + D_{f_d}(\tilde{x}_c, \tilde{u}_c, \tilde{\delta}_c) + g_{f_d}(\tilde{x}_c, \tilde{u}_c, \tilde{\delta}_c)$$

(57c)

$$y_d(k) = g_D(\tilde{x}_d, \tilde{u}_d, \tilde{\delta}_d)$$

(57d)

$$i f \hspace{0.5cm} x_d(k) = \tilde{x}_d, \hspace{0.5cm} u_d(k) = \tilde{u}_d, \hspace{0.5cm} (x_c(k), u_c(k)) \in X_{\tilde{X}_c}$$

(57e)

where $\tilde{\delta}_c = f_D(x_c(k), u_c(k))$ holds for any point $(x_c(k), u_c(k)) \in X_{\tilde{X}_c} \subset X_c \times U_c$. 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 Def. 20, one can call (57) as the BPWA-LC form of the decomposed DHA as well.

In worst case there are $2^{m_c + m_r + m_u}$ possible combinations of discrete variables $(\tilde{x}_d, \tilde{u}_d, \tilde{\delta}_c)$, but not all of them are feasible because of system limitations. In [41], the authors present efficient algorithms that enumerate all feasible modes of discrete variable $\tilde{\delta}_c$ that run in $O(n \times p(n, d) \times M(R))$ times and $O(n, d)$ space, where $p(n, d)$ denotes the complexity of solving a Linear Program (LP) with $n$ constraints and $d = m_c + m_r$ 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 the 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 small number of discrete-states may result in a 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 the Prop. 25, the complexity of the proposed DHA-to-PWA translation techniques based on the automaton-based DHA are 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 the proposed translation techniques can be used 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, however, they represent the typical properties of more realistic hybrid systems. Furthermore, there are some technical subtleties behind these examples. In continuous-time domain the DC-DC converter model do not represent any reset dynamic or memory type switching phenomena. Therefore, in 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 discrete-time domain, to avoid the inductor current to take unrealistic negative values, we need to define the reset dynamic and, here is the point, the DHA representation with discrete state dynamics is required. The DC-DC converter DHA represent 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 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

Fig. 9 shows the DHA of a room temperature control system [37]. The aim of the control system is to regulate the room temperature $x_r(k)$ between a lower bound $m$ and an upper bound $M$. When the heater is OFF or $x_d(k) = 0$, the room temperature is decreased according to the dynamic $x_r(k+1) = A x_r(k)$, 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.

Since the DHA of Fig. 9 satisfies the given conditions in Theorem 14, it is well-posed. Equations (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 21 according to the DHA of the room temperature control system in Fig. 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, nevertheless, one can verify that the BPWA and FPWA forms in Eqs. (58) and (59) satisfy the given conditions in Lemma 15, and therefore, they are well-posed. As it can be seen, the number of regions in the piecewise affine form of the BPWA representation is four and in the FPWA form is two. This is aligned with the obtained general results in Prop. 25. Note that the DHA of room temperature control system is without output and reset dynamics and according to the corollary, the number of independent regions in FPWA is equal to the number of nodes \( N_d = 2 \) and according to corollary 27, the number of regions in the BPWA is the sum of number of nodes and transition edges \( N_d + N_f = 2 + 2 = 4 \).

\[
x(k + 1) = \begin{cases} 
\begin{bmatrix} \lambda x_c(k) \\ 0 \end{bmatrix} & \text{if } x_d(k) = 0 \land x_c(k) \geq m \\
\begin{bmatrix} \lambda x_c(k) \\ 1 \end{bmatrix} & \text{if } x_d(k) = 0 \land x_c(k) < m \\
\begin{bmatrix} \lambda x_c(k) + (1 - \lambda)u_c(k) \\ 1 \end{bmatrix} & \text{if } x_d(k) = 1 \land x_c(k) < M \\
\begin{bmatrix} \lambda x_c(k) + (1 - \lambda)u_c(k) \\ 0 \end{bmatrix} & \text{if } x_d(k) = 1 \land x_c(k) \geq M 
\end{cases}
\]

\[
x(k + 1) = \begin{cases} 
\begin{bmatrix} \lambda x_c(k) + (1 - \lambda)u_c(k) \\ 1 \end{bmatrix} & \text{if } x_d(k) = 1 \land x_c(k) < M \\
\begin{bmatrix} \lambda x_c(k) \\ 0 \end{bmatrix} & \text{if } x_d(k) = 0 \land x_c(k) < m \\
\begin{bmatrix} \lambda x_c(k) \\ 0 \end{bmatrix} & \text{if } x_d(k) = 0 \land x_c(k) \geq m \\
\begin{bmatrix} \lambda x_c(k) \\ 0 \end{bmatrix} & \text{if } x_d(k) = 1 \land x_c(k) \geq M 
\end{cases}
\]

### 4.2. DC-DC converter

A DC-DC step-down (buck) converter with parasitic elements is shown in Fig. 10 [23, 63]. The DHA corresponding to the buck converter with different modes of operation is shown in Fig. 11. In this figure, \( N \) in \( i_d(k + N) \) is set to 2 for backward evolution and 1 for forward evolution (see Remarks 11 and 12). As a result, \( i_d(k + 2) = [1 \ 0](A_{d_1}x_c(k) + A_{d_2}B_{d_1} + B_{d_2}) \) and \( i_d(k + 1) = [1 \ 0](A_{d_2}x_c(k) + B_{d_2}) \). 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 Fig. 11. This figure has three discrete states, i.e. \([0 \ 1]^T, [1 \ 0]^T \) and \([0 \ 0]^T \) that correspond to the modes 1, 2 and 3, respectively.

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

\[
x_c(k + 1) = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}(A_{d_1}x_c(k) + h_{c_1}) = A_{d_1}x_c(k) + h_{c_1}
\]

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 \( x_d^{\text{new}} = [x_{d_1}(k), x_{d_2}(k), x_{d_2}(k - 1)]^T \) is defined. Although according to Eq. (27), all bits of the discrete state vector \( x_d(k - 1) \) are used to define the new discrete state vector \( x_d^{\text{new}} = [x_{d_1}(k)]^T, [x_{d_2}(k-1)]^T \), however, in the converter DHA of Fig. 11, one can see that using only \( x_{d_2}(k - 1) \) 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 \([0 \ 0]^T, [1 \ 1]^T \). However, for node \([1 \ 0]^T \) which has both controlled
and uncontrolled outgoing edges, condition (g) of Theorem 14 must be imposed at this node. As a result, the invariant condition $i_L(k+N) > 0 \land u_d(k) = 0$, and the switching guards associated with the outgoing edges of the converter DHA in 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 $i_L(k+N) > 0 \lor u_d(k) = 0$. Using the proposed algorithm in Lemma 21, the equivalent BPWA and FPWA models of
the converter DHA in Fig. 11 can be written as in Eqs. (61) and (62), respectively.

\[ x(k + 1) = \]

\[
\begin{bmatrix}
A_{d_1}x_c(k) + B_{d_0} \\
0 \\
0 \\
0
\end{bmatrix}
\]

\[ x_{d_1}^{\text{new}}(k) = [0 \ 0 \ 0]^T \land u_d(k) = 0 \]

\[
\begin{bmatrix}
A_{d_1}x_c(k) + B_{d_0} \\
0 \\
1 \\
0
\end{bmatrix}
\]

\[ x_{d_1}^{\text{new}}(k) = [0 \ 0 \ 1]^T \land u_d(k) = 1 \]

\[
\begin{bmatrix}
A_{d_1}x_c(k) + B_{d_0} \\
0 \\
1 \\
0
\end{bmatrix}
\]

\[ x_{d_1}^{\text{new}}(k) = [0 \ 1 \ 0]^T \land u_d(k) = 1 \]

\[ \cup (x_{d_1}^{\text{new}}(k) = [0 \ 1 \ 1]^T \land u_d(k) = 1) \]

\[
\begin{bmatrix}
A_{d_1}x_c(k) + B_{d_0} \\
1 \\
0 \\
0
\end{bmatrix}
\]

\[ x_{d_1}^{\text{new}}(k) = [0 \ 1 \ 0]^T \land u_d(k) = 0 \]

\[
\begin{bmatrix}
A_{d_1}x_c(k) + B_{d_0} \\
0 \\
0 \\
1
\end{bmatrix}
\]

\[ x_{d_1}^{\text{new}}(k) = [1 \ 0 \ 0]^T \land u_d(k) = 0 \]

\[ \land (x_{d_1}^{\text{new}}(k) = [0 \ 1 \ 0]^T \land u_d(k) = 0 \land [1\ 0](A_{d_2}x_c(k) + A_{d_1}B_{d_2} + B_{d_1}) > 0) \]

\[
\begin{bmatrix}
A_{d_1}x_c(k) + B_{d_0} \\
0 \\
0 \\
1
\end{bmatrix}
\]

\[ x_{d_1}^{\text{new}}(k) = [1 \ 0 \ 1]^T \land u_d(k) = 0 \land [1\ 1](A_{d_2}x_c(k) + A_{d_1}B_{d_2} + B_{d_1}) > 0 \]

\[ \land (x_{d_1}^{\text{new}}(k) = [1 \ 0 \ 0]^T \land u_d(k) = 0 \land [1\ 0](A_{d_2}x_c(k) + A_{d_1}B_{d_2} + B_{d_1}) \leq 0) \]

\[
\begin{bmatrix}
A_{d_1}x_c(k) + B_{d_0} \\
0 \\
0 \\
1
\end{bmatrix}
\]

\[ x_{d_1}^{\text{new}}(k) = [1 \ 0 \ 1]^T \land u_d(k) = 1 \land [1\ 1](A_{d_2}x_c(k) + A_{d_1}B_{d_2} + B_{d_1}) > 0 \]

\[ \land (x_{d_1}^{\text{new}}(k) = [1 \ 0 \ 0]^T \land u_d(k) = 1 \land [1\ 0](A_{d_2}x_c(k) + A_{d_1}B_{d_2} + B_{d_1}) \leq 0) \]

\[
\begin{bmatrix}
A_{d_1}x_c(k) + B_{d_0} \\
0 \\
0 \\
0
\end{bmatrix}
\]

\[ x_{d_1}^{\text{new}}(k) = [0 \ 0 \ 1]^T \land u_d(k) = 0 \]

\[
\begin{bmatrix}
A_{d_1}x_c(k) + B_{d_0} \\
0 \\
0 \\
1
\end{bmatrix}
\]

\[ x_{d_1}^{\text{new}}(k) = [0 \ 0 \ 1]^T \land u_d(k) = 1 \]

31
\[ x(k + 1) = \begin{cases} 
A_{d_1}x(k) + B_{d_1} \\
0 \\
n_0 
\end{cases} \\
A_{d_1}x(k) + B_{d_1} \\
0 \\
1 
\begin{cases} 
(x_d(k) = 0) \land u_d(k) = 0 \\
(x_d(k) = 0) \land u_d(k) = 1 \\
(x_d(k) = 1) \land u_d(k) = 0 \\
(x_d(k) = 1) \land u_d(k) = 1 \\
(A_{d_1}x_c(k) + B_{d_1}) > 0 
\end{cases} \]

As it can be seen from Eqs. (61) and (62), the number of regions in the the BPWA form is nine and in the FPWA representation is four. According to Prop. 25, for the DHA of the DC-DC buck converter in Fig. 11, we have \( n_{\text{inter}}(1) = n_{\text{inter}}(2) = 0, n_{\text{inter}}(3) = 1, n_{\text{out}}(1) = n_{\text{out}}(3) = 1, n_{\text{out}}(2) = 2, N_{\text{self}} = 0, N_d = 3, N_t = 4, N_{\text{inter}} = 1 \). As a result, based on the given formulations in Props. 26 and 28, the number of independent regions in the equivalent BPWA and FPWA forms are 9 and 4 respectively, which are aligned with the obtained results in Eqs. (61) and (62).

5. Conclusion

In this paper, effective methods for the translation of an automaton-based DHA to its equivalent 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 is shown that changing the DHA model structure from the traditional decomposed construction to an automaton-based structure, reduces 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 DHA two types of evolutions, i.e., Backward and Forward evolutions are defined and associated with each type of evolution, two types of PWA systems, i.e., BPWA and FPWA systems are extracted. Some formulations are provided that quantify the exact number of regions in the BPWA and FPWA systems. It has been shown that the number of such regions in the BPWA systems is greater 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. Examples are presented to provide evidence for the merit of the proposed techniques. An importance 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


Biographies

Mohammad Hejri received his B.Sc. degree from Tabriz University in 2000 and the M.Sc. degree from Sharif University of Technology, Tehran, Iran in 2002 both in electrical engineering. He received his Ph.D. 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 with the Department of Electric Power and Energy Systems, School of Electrical Engineering, Royal Institute of Technology (KTH), Stockholm, Sweden. Since 2012, he has been an Assistant Professor with the Department of Electrical Engineering, Sahand University of Technology, Tabriz. His research interests include control theory with applications in power electronics, renewable energy and power systems.

Hossein Mokhtari received his B.Sc. degree in electrical engineering from Tehran University in 1989. From 1989-1992, he worked in the Consulting Division of Power Systems Dispatching Projects for the Electric Power Research Center Institute. Dr. Mokhtari obtained his M.Sc. degree in power electronics from the University of New Brunswick, Fredericton, NB, Canada, in 1994, and his Ph.D. 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.

Figure captions

Figure 1: DHA with decomposed structure
Figure 2: Automaton associated with a DHA
Figure 3: A node of a DHA with incoming and outgoing edges
Figure 4: Concerned part of a DHA net to extract BPWA and FPWA model
Figure 5: Concerned part of a DHA net to extract BPWA and FPWA model
Figure 6: Concerned part of a DHA net to extract BPWA and FPWA model
Figure 7: Concerned part of a DHA net to extract BPWA and FPWA model
Figure 8: Concerned part of a DHA net to extract BPWA and FPWA model
Figure 9: The DHA of a temperature control system
Figure 10: DC-DC Buck converter
Figure 11: DHA of a buck converter

Figures with their captions and numbers
Figure 1: DHA with decomposed structure

Figure 2: Automaton associated with a DHA

Figure 3: A node of a DHA with incoming and outgoing edges
Figure 4: Concerned part of a DHA net to extract BPWA and FPWA model

\[
\begin{align*}
    x_t(k+1) &= A_x x_t(k) + B_y u_t(k) + f_1 \\
    y_t(k) &= C_x x_t(k) + D_y u_t(k) + g_1 \\
    y_t(k) &= C_z x_t(k) + D_z u_z(k) + g_2
    \end{align*}
\]

Figure 5: Concerned part of a DHA net to extract BPWA and FPWA model

Figure 6: Concerned part of a DHA net to extract BPWA and FPWA model
Figure 7: Concerned part of a DHA net to extract BPWA and FPWA model

\[
x_{e}(k+1) = A_{e}x_{e}(k) + B_{e}u_{e}(k) + f_{e}
\]

Figure 8: Concerned part of a DHA net to extract BPWA and FPWA model

\[
x_{e}(k+1) = A_{e}x_{e}(k) + B_{e}u_{e}(k) + f_{e}
\]

Figure 9: The DHA of a temperature control system

\[
x_{e}(k+1) = \lambda x_{e}(k) + (1-\lambda)u_{e}(k)
\]

Figure 10: DC-DC Buck converter
Figure 11: DHA of a buck converter