The 10th International Conference on Future Networks and Communications (FNC 2015)

Combining Bond Graphs and Petri Nets Formalism for Modeling Hybrid Dynamic Systems

Mokhtar Bouhalouanea,*, Sekhri Larbi, Hafid Haffaf

Abstract

The hybrid nature of complex systems such Hybrid Dynamic Systems (HDS) makes the modeling, analysis and monitoring tasks very challenging. The complexity due to the hybrid behavior of HDS remains always the most significant reason. However, the graphical models such Bond graph and Petri Nets are extremely appreciable allowing the best description of the internal causalities in HDS. In this paper, we propose a unified formal based approach coupling Bond graphs and Petri Nets theories to model HDS. This approach gives an efficient solution to the transition problematic and serves as theoretical study for complexity reduction of hybrid models.

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Peer-review under responsibility of the Conference Program Chairs

Keywords: Hybrid Dynamic Systems, Bond Graph, Petri Net, Mode Transition System, Switch Transition System;

1. Introduction

Modeling and analysis of physical systems are the most crucial issues. The study of problems related to the control and monitoring of these systems is relatively new. In this work, we are interested by the changing modes and transition problematic in HDS: systems where continuous and discrete dynamics interact and influence each other behavior. For the continuous part; this result in abrupt changes that can occur in the dynamic either as switches in the continuous vector state and in the set of algebraic equations, or as jumps in the continuous state. Reversely, the continuous evolution affects the discrete part by generating discrete events that change the discrete state. However, it is often up to the discrete part to control and to handle the transitions and the changing modes of the continuous

* Corresponding author.
E-mail address: bouhalouane.mokhtar@hotmail.com (M. Bouhalouane).
states. In our study, such system states are often described by switched Bond graph. So, combining Bond graph and Petri Nets theories can allow an elegant solution to solve these problems.

We will also be based on the aim of extraction of some feared scenarios in computer-controlled systems. These scenarios characterize the sequences of actions leading to dangerous situations. Our approach based on both qualitative and quantities models can determines more precisely the exact conditions of the occurrence of the feared event, i.e. what has led the system to leave its normal operation and to evolve into the feared state.

HDS have dynamical evolutions, where the behavior of interest is determined by interacting explicitly and simultaneously continuous and discrete dynamics. HDS require for their description the use of continuous time model, discrete event model and the interface between them including at the same time, continuous state and discrete state variables.

2. Modeling hybrid dynamic systems

2.1. Bond Graph Modeling

Bond graph model was developed in 1961 by H. Paynter for continuous systems. It is a graph of structured bonds widely used for modeling, analyzing and simulation of physical systems. Bond graph is a multi-disciplinary graphical language that permits the representation of power transfers within the system. The classical Bond graph was enriched by new elements, namely switch elements to achieve the study of commutations and non linearity of HDS. Abrupt behavioral changes are not uncommon. Indeed, physical switching phenomena are designed as discrete events and are ubiquitous in engineering applications.

Characteristic of physical systems involving such phenomena is that they do not merely involve discontinuous changes in the value of one or more physical quantities. Two kinds of discontinuous changes may occur in physical system behavior:

- Changes that are relatives to physical components such as electrical switches, hydraulic valves, diodes, etc
- Changes that can occur in structure such as breaks in electrical wires and pipes, etc.

In our approach, switching modes are handled by switching mechanisms having two distinct states (On/Off) denoted Sw. In the first state, it acts as an ideal effort source with zero output; in the second state, it acts as an ideal flow source, also with zero output. These states are determined by a control structure depending on other quantities inside and/or outside the model.

2.2. Petri Nets Modeling

Petri Nets model is widely used to describe a wide range of systems that have a very powerful aspect of events. It was primarily intended for discrete event systems. However, a crucial problem was the explosion of the number of elements of their graphical form when they are used to describe complex systems. High-level Petri Nets were developed to overcome this problem by introducing higher-level concepts, such as the use of complex structured data as tokens, and using algebraic expressions to annotate net elements. The first most significant advance was the extension of the classical Petri Nets by new forms leading to hybrid Petri Nets.

2.3. Switching Phenomena Modeling

Works in real-time systems show that system behavior is a function of time and events dependency. These system models must incorporate dynamic views with different system configurations and states. In general, a system with n such components each with k behavior modes can assume k^n overall configurations or behavioral modes.

Initial attempt to accommodate abrupt switching in Bond graphs were made to introduce switching elements into the model in the form of modulated transformers and non linear resistive elements. The shortcoming of this method is that produce energy dissipation, therefore, could not model ideal switching elements. Broenink and Wijbrans have introduced the concept of switching bonds, which are used to connect or disconnect sub-models.
The switching bond is connected between two junctions. A control box, implemented as a finite state automaton, connected to the switching bond determines their On and Off states. The drawback of this method is in causing hanging junctions and changing boundary due to switching incorrectly handled.

Compositional modeling approach was proposed by Mosterman and Biswas enabling to compose dynamically model fragments\textsuperscript{12}. The approach translates the overall physical model to one Bond graph model that covers the energy flow relations within the system, and the signal flow model based on controlled junctions where the states are determined by a local control mechanism, implemented as a finite state automaton. Then, the modeling task involves the combination of two kinds of model: Bond graph for energy transfer modeling and Finite state automaton for controlled junctions modeling\textsuperscript{13}.

A controlled junction assumes one among two states On/Off where the input to its signal determines the On/Off states of the junctions. When the controlled junction is active it behaves like normal 0 or 1 junctions, but when it takes the state Off it forces the effort values or the flow values (according to its type) at all connected bonds to become 0. The algorithm for mode switching through mythical states is MMA (Mythic Mode Algorithm)\textsuperscript{14,15}. When multiple controlled junctions are handled in sequence, the effort and flow values at each of the intermediate modes are computed from the effort and flow values at the start of the switching process. And not the effort and flow values computed for the last mythical state, because the intermediate modes are not considered to be real behavior states that the system goes through.

Our study is based on Stromberg’s work\textsuperscript{2,16}. In this work discontinuous changes are handled by imposing zero-output flow (switch Off) and zero output effort (switch On). There is no energy dissipation and at each time we have the following relation verified:

\[ \text{Power} = \text{Effort} \times \text{Flow} = 0 \]  

Petri Nets formalism offers some advantages over finite automata, and it is also useful for HDS control. In general, the main objective of our approach can be summarized in the following steps:

- Modeling the system without constraints by Bond graph.
- Associating the switched Bond graph with Petri Nets and systematic composition of the overall computational model for numerical simulation.
- Reducing the complexity with drawing some specifications on its system behavior, that will help thereafter for a well defined control structure and monitoring that ensure the restrictions on the system behavior.

3. systematic composition procedure

We need to take some well determined steps in order to obtain the overall computational model. Thus, the intuitive presentation:

3.1. Towards hybridation

To obtain the representation of the causal Bond graph model related with Petri Net model we need a mathematical structure called \textit{MTS} (Mode Transition System) deduced from the related works in\textsuperscript{17,18}.

An \textit{MTS} is a three-tuple \(<M, T, Q>\) where:

\- \(M\): a non-empty set of modes
\- \(T\): a non-empty set of transitions
\- \(Q\): a hybrid model structure \((\mathcal{E}, X, U, \Pi, G, A)\)
  \- \(\mathcal{E}\): control mechanism, is effort from 0-junctions and flow values from 1-junction and its output is a control.
  \- \(X\): is a non-empty set of real-valued model variables
  \- \(U\): is a set of real valued independent variables
  \- \(\Pi\): is a set of computational continuous mode-models over variables in \(X\) such that only variables in \(U\) are independents.
  \- \(G\): a non-empty set of Boolean conditions
A: a set of initialization rules of the form
\[ Z = \Phi(x_1, x_2, \ldots, x_n) \] where \( Z \in X \cup U, X \subseteq X \) and \( \Phi: R^n \rightarrow R \)

A mode is a pair \((\ell, \lambda)\) where \( \ell \in \mathcal{E} \) and \( \lambda \in \mathcal{M} \). A transition is a three-tuple \(<e, g, a>\) where: \( e \in \mathcal{M}^* \mathcal{M} \), \( g \in \mathcal{G} \) and \( a \subseteq A \) such that for any two distinct elements \( a, a' \in A \), the left-hand sides are also distinct. When the system makes transition, if this transition has at least one initialization rule \( a \subset A \) then these rules are applied before the new mode is entered. For all the variables \( Z \) defined by differential equations in the new mode, but not initialized by any initialization rule in the transition, the initial values are equal to final values of the same variables in the previous mode. The discrete determinism is ensured if the following condition is verified:

\[ \forall t \forall m, m', m'' \in M \left[ (m' \neq m'') \rightarrow -(g_{m,m'}(z(t)) \land g_{m,m''}(z(t))) \right] \]  (2)

This can be represented by a Petri Net model, where each composite state in \( M \) is represented by a place and each transition condition into another mode is represented by a transition. So, \(<e, g, a>\) defines graphically the flow relation with weight (multiplicity) equal to 1 for each arc. Algebraically it defines the incidence matrix where the places and the transitions are:

\[ P = (\ell, \lambda), \quad e = P \times P, \quad T = g \]  (3)

So, in this definition, \(<e, g, a>\) can be restricted to \(<e, g>\) in order to facilitate the relationship with the Petri Net, because initialization rules concern only the continuous part, in other words they concern only the modes. This can be viewed that in each marked place there are a switched Bond graph characterized by the token. In other words it characterizes a set of differential algebraic equations. We will see later how to obtain pre-incidence and post-incidence matrix from the case study.

3.2. Composing the Computational Model

Petri Nets model is well adapted to the parallelism. A systematic composition is used to obtain the computational model intended for the numerical simulation. It involves the composition of the overall \( STS \) and the composition of the explicit computational \( MTS \). The first composition aims to build the discrete part and the second composition is to build dynamically continuous part by assembling the computational states. We replace each mode in \( STS \) by its Bond graph fragment; we obtain a switched Bond graph for each mode. Then, we use the causal propagation in order to introduce numerical order into the obtained model. Thus, we obtain a switched bond graph for each state where all variables are computationally organized as input/output variables.

In such case we have more than one control structure. Formally, the composition is parallel where obtained model captures the collective behavior of all the switches working independently. To this end, we first need the definition of the invariance condition.

\[ S = <M, T, Q> \] is an \( STS \); the invariance condition for a mode \( m \), is denoted \( g_{IM} \):

\[ g_{IM} = \land g_{m,m'} \text{ where:} m' \in M - \{m\} \]

\[ g_{m,m'} = \begin{cases} \text{False if } <(m, m'), g> \notin T \\ \text{g if } <(m, m'), g> \in T \end{cases} \]

3.3. Formal Definition of the Composition

\[ S_1 = <M_1, T_1, Q_1> \] and \( S_2 = <M_2, T_2, Q_2> \) are two control structures ; the parallel composition \( S_1 \parallel S_2 \) is a control structure \( S = <M, T, Q> \) defined as follows:

\[ Q = \langle \ell, Z, U, \Pi, G \rangle \]
\[ \mathcal{E} = \mathcal{E}_1 \times \mathcal{E}_2 \]
\[ \mathcal{Z} = \mathcal{Z}_1 \times \mathcal{Z}_2 \]
\[ \mathcal{U} = \mathcal{U}_1 \uplus \mathcal{U}_2 \]
\[ \Pi = \{ \lambda_{12} \mid \lambda_{12} = \lambda_1 \cup \lambda_2 \}, \lambda_1 \in \Pi_1 \text{ and } \lambda_2 \in \Pi_2 \} \]
\[ \mathcal{G} = \mathcal{G}_1 \uplus \mathcal{G}_2 \uplus \mathcal{G}_{12} \text{ with:} \]
\[ \mathcal{G}_1 = \{ g \mid g = g_1 \wedge g_{12}, g_1 \in \mathcal{G}_1, g_{12} \in \mathcal{G}_{12} \} \]
\[ \mathcal{G}_2 = \{ g \mid g = g_2 \wedge g_{12}, g_2 \in \mathcal{G}_2, g_{12} \in \mathcal{G}_{12} \} \]
\[ \mathcal{G}_{12} = \{ g_{12} \mid g_{12} = g_1 \wedge g_2, g_1 \in \mathcal{G}_1, g_2 \in \mathcal{G}_2 \} \]
\[ \mathcal{M} = \{ m_{12} \mid m_{12} = (\mathcal{E}_1 \uplus \mathcal{Z}_1, \mathcal{E}_1 \uplus \mathcal{Z}_2), (\mathcal{E}_2, \mathcal{Z}_2) \in \Pi_1 \text{ and } (\mathcal{E}_2, \mathcal{Z}_2) \in \Pi_2 \} \]
\[ \mathcal{T} = \mathcal{T}_1 \uplus \mathcal{T}_2 \uplus \mathcal{T}_{12} \text{ with:} \]
\[ \langle \mathcal{M} \rangle \times \langle \mathcal{T} \rangle \times \langle \mathcal{G} \rangle \times \langle \mathcal{M} \rangle \]
\[ \langle \mathcal{T} \rangle \times \langle \mathcal{G} \rangle \times \langle \mathcal{M} \rangle \times \langle \mathcal{T} \rangle \times \langle \mathcal{G} \rangle \times \langle \mathcal{M} \rangle \]

In reality, we need to observe the changing modes of the system as a whole. Also theoretically, in order to reduce the complexity of the composition model, we apply an interleaved composition which is similar to a parallel composition except for the addition of the assumption that only one switch can change mode at a time, even if many switches change mode simultaneously. Therefore, we avoid from the definition of the parallel composition: \( G_{12} \) from the set \( G \) and \( T_{12} \) from the set \( T \).

### 4. Complexity analysis

Consider the following pedagogical example of an electrical system with ideal diode (fig 1.(a)). Its bond graph representation is given in (fig 1.(b)).

![Electrical system with ideal diode](image)

![Bond graph representation](image)

From the Bond graph representation there are four states of the diode:
- \( m_{gf} \): blocking and the relay is \( \text{Off} \).
- \( m_{gf} \): blocking and the relay is \( \text{On} \).
- \( m_{ef} \): conducting and the relay is \( \text{On} \).
- \( m_{ee} \): conducting and the relay is \( \text{On} \).

By application of SCAP (Sequential Causal Assignment Procedure) on each state \( m_{ee} \), \( m_{ef} \), \( m_{gf} \), \( m_{gf} \) in order to introduce causality; we find that there is no conflict free causal assignment in all four states. All four states are both reachable and consistent. Applying interleaved composition, by the following reasoning we get the Petri Net of the global STS to our example as follows:

\[ \langle m_{gf}, m_{gd} \rangle, u_2 > t^+ \]
\[ \langle m_{g}, m_{gd} \rangle, e_1 > 0 \]
\[ \langle m_{ef}, m_{gf}, f_1 > 0 \]
\[ \langle m_{ef}, m_{gf}, f_1 > 0 \]
\[ \langle m_{ef}, m_{gf}, u_2 > t^+ \]
\[ \langle m_{ee}, m_{eg} \rangle, u_2 > t^+ \]

So, the transitions of Petri Net are:
\[ t_1 : g_1 = u_2 > t^+ \quad t_8 : g_1 = e_1 > 0 \]
\[ t_5 : g_2 = u_2 < t^- \quad t_9 : g_6 = f_1 < 0 \]
\[ t_6 : g_3 = e_1 > 0 \quad t_{10} : g_7 = u_2 < t^- \]
\[ t_7 : g_4 = f_1 < 0 \quad t_{11} : g_8 = u_2 > t^+ \]

We add a place \( p_0 \) relating to the system before functioning, in other words when the system is in rest. We add also four transitions \( t_0, t_1, t_2 \) and \( t_3 \) which are used to release a functioning state or mode.

Note that this place \( p_0 \) is initially marked and represent a state \( m_0 \) in reset. From what before, we obtain the pre-incidence and the post-incidence matrix as follows:

\[
\begin{bmatrix}
  t_0 & t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 & t_8 & t_9 & t_{10} & t_{11} \\
 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
\end{bmatrix}
\]

It is now clear how to calculate the incidence matrix: \( \text{Incidence} = \text{Post} - \text{Pre} \). Therefore, we can draw the Petri Net of the global STS of our system:

![Petri Net](image)

Since there are no conflict free causal assignment in all four states. What is very important is that the conditions of transition \( t_1 \) can be explained in function of source variables \( u, u_1 (e_1, f_1 \) can be explained by input variables). This procedure can easily be applied to other more complicated examples. The overall system of hybrid simulation may be composed of three modules:

- **MODELICA** or **SIMUL**: simulator of the continuous part.
- **TINA**: Time Net Analyzer, which determines the switching modes.
• **MMA:** algorithm of the mythical modes.

The simulation results with TINA tool applied to this example generates a Petri Net from a textual description edited in Netdraw (fig 3.(a)). From this file we can apply the reachability analysis to obtain the marking graph, the reachable states and the state classes. We can also obtain an extension file (.txt) by applying structural analysis in order to obtain the place invariants and transition invariants (fig 3.(b))

Assuming that $u > u_1 > 0$, we find that there are transient states in the system which are $\{m_{ff}, m_{EE}\}$. So, from the STS of the system, we note: whenever the relay is turned On, the diode will also immediately turn Off and whenever the relay is turned Off, the diode will immediately turn On.

The physical interpretation of this situation is that the transient states no correspond to any physical real state. As a matter of fact, the transient states appear in the model just by the systematic composition and we can accept them or not. Whether we accept them or not depends on the purpose of modeling. If the purpose is the preservation of structure we might be able to accept them, but if the purpose is to study the variable state by numerical simulation we might be able avoid them by considering the two switches as an implicit two port switch and replace it with an explicit two-port switch\textsuperscript{20}.

Consider another similar electrical system where its Bond-graph representation as follows:

![Bond-graph representation](image)

Note that the Petri Net model representing this system is the same model of the previous example. By application of SCAP procedure to the Bond graph model we obtain two states in which there is a causal conflict; in other words we obtain two inconsistent functioning modes. So, we can’t write the transition conditions in function of the input variables. But an important question arises: are these states transient or not? If they are transient then these states correspond to model defect with no interpretation in the real system else they correspond to destructive states which can really appear in the system. This kind of states should be avoided by an active control in the control structure of the Petri Net model serving to monitoring the system and ensuring restrictions on its behavior. The ordered set of events which lead to wards these states is referred to as a critical scenario. The method of extraction of feared scenarios should be made up of two steps: backward and forward reasoning in Petri Net. Taking into account the
failures, the temporal constraints and partially the continuous dynamic of the system makes it possible to respect the order of events appearance in the generated scenarios\textsuperscript{21,22}.

5. Conclusion

We have elaborated an hybridation studying switching phenomena where the dynamic behavior is described by a finite number of models with a set of rules among these models in order to conceive a control structure. We have taken well determined steps to obtain dynamically the overall computational model intended for simulation. To this end we have also presented a formal method to construct the discrete parts. Finally, we have discussed some ideas to deal with model complexity and non physical modes combining both structural detail and conceptual clarity. We have also discussed how to deal with critical scenarios, and hopes that this approach will be generalized for more complex systems.

References