MODELING OF HYBRID CONTROL SYSTEMS

András VÖRÖS
Advisor: Tamás BARTHA

I. Introduction

Petri nets are widely used to model and analyze discrete event dynamic systems (DEDS). However, modern complex control systems consist of both continuous and discrete parts. Usually there is a discrete digital control component defining a mode of separation, and some kind of continuous controllers. These control systems are the so-called hybrid systems, describing the behavior of them we have to combine both continuous and discrete approaches. In this paper I introduce two Petri net based approaches for the modeling of hybrid control systems.

II. Discrete and Continuous Petri Nets

Petri nets are graphical models for concurrent and asynchronous systems providing both structural and dynamical analysis possibilities like reachability and invariant behavior analysis.

A (marked) discrete Petri net is a 5-tuple $N = (P, T, w^-, w^+, M_0)$ represented graphically by a digraph, where $P = \{p_1, p_2,\dots, p_n\}$ is a finite set of places, $T = \{t_1, t_2,\dots, t_m\}$ is a finite set of transitions, $P \cap T = \emptyset$, w^-: $P \times T \rightarrow N$ is the input, w^+: $T \times P \rightarrow N$ is the output incidence function for each transition, represented by weighted arcs from places to transitions and from transitions to places, $w^*_{i,j}$ (or $w^*(p_i,t_j)$) is the signed weight of the arc between the i-th transition and the j-th place and $M_0 : P \rightarrow N$ is the initial marking, represented by tokens in the places. The |P|×|T| dimension matrix W incidence matrix is built from the values of $w^*_{i,j}$ respectively (formally: $W = \| w^*_{i,j} \|$). With the help of this matrix we can write up the fundamental state equation: $M = M_0 + W \cdot s$, where s is the firing vector corresponding to the transitions. Reachability: we call M reachable from M_0 if exists a sequence $\sigma = \langle s_1, s_2 \dots s_n \rangle$ of firings that $M_n = M_0 + W \cdot \sigma$ and $M \leq M_n$. The positive integer vector solution of this equation provides us a necessary condition for the reachability problem.

A timed Petri net has a timing function *time*: $T \rightarrow R^+$. This function associates a time duration to each transition, this duration is the time from the enabling of the transition to the firing.

Continuous Petri nets [1] can be gained by the *fluidification* of discrete nets. We say that a Petri net is continuous if the firings and the tokens in a place can not only be integer but real numbers. It is a relaxation of the discrete case, which can be gained by the split of each marking into k tokens, where k tends to infinity. In this case, the reachable markings become infinite.

A marked continuous Petri net is a 5-tuple $N = (P, T, w^-, w^+, M_0)$, P and T are similar to the discrete case, but $w^-: P \times T \rightarrow R^+$ is the input incidence function, $w^+: T \times P \rightarrow R^+$ is the output incidence function and M_0: $P \rightarrow R^+$ is the initial marking.

A transition t is enabled at M if for every p_i, where $w^-(p_i, t) > 0$, $M(p_i) > 0$.

The main advantage of the continuous relaxation of discrete nets is that it preserves the invariant behavior and *liveness* properties of the system in a significantly reduced state space.

Timing can be extended to the continuous case: we associate each transition a maximum firing speed, which is defined: if d_i is the timing associated to t_i in the discrete net, then $1/d_i$ will be the firing rate (flow through the transition) of the corresponding continuous net. This provides the modeling of processes, which are described by linear differential equations from any order.

Combining the approaches, we have to define the interface between the discrete and continuous nets. The discrete parts can affect the continuous parts through the continuous transitions as they can enable or disable them. If n discrete tokens are needed to enable a continuous transition, after the firing of it we have to take n token back to the discrete net. A continuous transition can't change the number of discrete tokens, by definition [1]. The continuous part can also affect the behavior of the discrete part, but in this case it is not necessary, that the amount of token remains the same at the continuous net. This affect means an arc from a continuous place to a discrete transition with a given weight; this will always fire when it is enabled as the priority of discrete transitions is higher than continuous transitions [1]. This arc will indicate weather the continuous part reached a predefined state, which is important for the discrete controller parts and continuous transitions can also compute the output signal to the controlled system. With the help of this approach we can model the influence of the discrete controller to the continuous controller and to the controlled processes efficiently.

Reachability analysis in hybrid Petri nets differs from discrete ones, as we have to consider continuous trajectories too. In the case of autonomous hybrid Petri nets - where continuous parts are considered as a relaxation of discrete nets - the reachability space is *relaxed* as we take into account only the fact weather a continuous place is marked or not, without the exact marking.

In timed hybrid Petri nets we examine the invariant behavior (IB) states of the system, where IB-state is such that the marking of the discrete part and the instantaneous speed vector of the continuous part remain constant as long as the system is in the same IB-state [1] . Each IB-state corresponds to a state in the corresponding hybrid automata, so after the determination of the set of IB-states we are also able to construct hybrid automata from the Petri net.

III. Object oriented approach

Hybrid models can be divided into smaller parts - objects – in order to make the analysis easier [2]. This approach defines a hierarchy in the system. As complex control systems consist of some smaller components, this decomposition method is a natural way to make the analysis easier. Each controller object is described by a Petri net, which uses continuous variables during the operation, and shares variables, places or transitions with the other objects. The creation of object oriented hybrid Petri net consists of a few steps: (1.) Definition of the control subsystems; (2.) Definition of the control Petri nets in each subsystem; (3.) Definition of the object variables; (4.) Definition of the interfaces between the objects, which contains shared variables and transitions; This few steps enable us to decompose the analysis into smaller steps and we can examine analysis questions locally. At first we examine the controller Petri nets of the object under examination with the help of linear logic. The smaller controller Petri nets are restricted to be safe. Linear logic let us make assumptions to the whole system from some components as linear logic makes the analysis process composable [2] .
The analysis contains the following steps: (1.) Formal specification of the property statement; (2.) Analysis of the first object: (2.1.) Analysis of the discrete dynamics with the help of linear logic; (2.2.) Analysis of the continuous dynamics for the necessary discrete states; (2.3.) Analysis of other object, which affects the analysis of the first object; The other objects affect only through the interface transitions and variables the behavior of the analyzed object; we have to examine other objects only when interface transition or variable is reached during the process. In each level of object hierarchy well defined interfaces help us to trace changes in the controller, and from the simple and small controller objects we are able to compose complex hybrid controller systems.

IV. Conclusion, further work

Combining the above introduced methods may provide an efficient analysis method for hybrid systems. The next step will be implementing analysis algorithms and examine the adaptability of them to our industrial systems and problems.

References

[1] René David, Hassane Alla: Discrete, Continuous and Hybrid Petri Nets, Springer 2005, ISBN: 3-540-22480-7

[2] E. Villani, P. E. Miyagi, R. Valette: Modelling and Analysis of Hybrid Supervisory Systems, Springer 2007