Minimization of Linear Constraints in Constant Slope Hybrid Dynamic Systems

Yamen El Touati

Department of Computer Sciences, Faculty of Computing and Information Technology,
Northern Border University, Kingdom of Saudi Arabia.

Summary
Dynamic hybrid systems (DHS) cover a wide variety of real time system as well as embedded systems where the behavior is captured by an interaction between discrete and continuous components. Typically, these systems are modelled by general Hybrid Automata (HA) which extend Finite state automata by differential equations and linear inequalities on the variables that model the system. The reachability problem is known undecidable for general Hybrid Automata. This means that there is no algorithm capable to sole the general form of the problem by a computer. However, many restrictions on general hybrid automata are proposed in literature where the reachability problem becomes decidable and covers in the same time a large category of systems. We focus on an interesting class of Rectangular HA (RHA) covering important aspects of real time systems. Besides, configurations and transitions of this modelling framework are defined by conjunction of linear constraints. And, the system of constraints related to system variables forms a convex polyhedron. In many cases, the constraints are computed by specific algorithms that may produce redundant constraints and where this redundancy is not easy to disclose. In this paper, we propose an approach to transform the initial linear constraints by an equivalent and minimized polyhedron.

Key words:
linear constraints, polyhedron, dynamic hybrid systems.

1. Introduction

Dynamic Hybrid Systems [1, 2, 3, 4] (DHS) are systems characterized by the interaction of both discrete and continuous components. A large variety of real-time and embedded systems and many computer automated systems as well as industrial and electrical systems are described by both continuous and discrete aspects. Tasks related to DHS, such as modeling, supervision and analysis, often pose complicated and challenging problems. Two types of communities are interested in DHS models: the discrete event systems (DES) community and the continuous systems community.
Within the community of continuous systems, DHS are modeled as systems that transition among various continuous models. This enables the researchers and engineers with continuous systems backgrounds to apply readily available techniques from the continuous systems literatures. Nevertheless, performing computations and analysis with such models can easily become a daunting task, especially for hybrid systems with a strong discrete component, which exhibit frequent switching between a multitude of different continuous models. Currently within the DES field, several different modeling frameworks are being used for modeling DHS. The most commonly used amongst them are timed and hybrid extensions of Petri nets [5,8,9,10,11] and automata [6,7].
As for the timed extensions, based on automata, such as timed automata [12,13], stop watch automata [14,15] and time transition systems [16,17], time constraints are added to states/configurations and event transitions. Global clocks are used to characterize the continuous system behavior. However, since automata do not naturally benefit from an intuitive graphical representation, the models they capture can easily become unmanageable, especially for complex DHS, requiring a large number of configurations, clocks, and/or clock resets.
As for the hybrid automata models, we can list hybrid automata [18, 19], linear hybrid automata [20, 21] and rectangular automata [22, 23] as the most commonly used ones. The behavior of the system in hybrid automata is captured by a multitude of variables that reflect the continuous states of the system and where the evolution is governed by differential equations. Switching between configurations is triggered by transitions with variable constraints called Guards. Remaining in one configuration is conditioned also by variable constraints called Invariant. The set of possible values of variables in a given configuration is called the reached space. In general, the set of reached space is given by a number of inequalities that forms a polyhedron [24, 25]. One of the problems to solve is to presents theses inequalities in a minimalistic way.
This paper is organized as follows. In the next section, we characterize a subclass of RHA formalism and its time transition system semantic. In section 3, we present and solve the minimization problem that we illustrate with some case study modelling. Our aim is to show that our approach is efficient to present reachable space of RHA in a minimalistic way.

Manuscript received October 5, 2018
Manuscript revised October 20, 2018

2. Rectangular Hybrid Automata subclass

We consider these notations. $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ is a finite set of real valued clocks (variables). $\dot{\mathcal{X}} = \{\dot{x}, x \in \mathcal{X}\}$ denotes the set of first derivative variables of $\mathcal{X}$. A variable x is considered piece-wise linear variable if $\dot{x} \in \mathbb{R}$. $\sim$ denotes an element of operator's set $\{<, \leq, =, \geq, >, \neq\}$. A rectangular inequality over $\mathcal{X}$, is an inequality of the form, $x \sim c$, where $c \in \mathbb{R}$, and $x \in \mathcal{X}$. A rectangular predicate over $\mathcal{X}$ is a conjunction of rectangular inequalities over $\mathcal{X}$. $Rect(\mathcal{X})$ denotes the set of rectangular predicates over $\mathcal{X}$. A polyhedral inequality over $\mathcal{X}$ is an inequality of the form $c_1 x_1 + \cdots + c_k x_k \sim c$, where $c, c_1, \ldots, c_k \in \mathbb{R}$, and $x_1, \ldots, x_k \in \mathcal{X}$. A polyhedral predicate over $\mathcal{X}$ is boolean combination of polyhedral inequalities over $\mathcal{X}$. $\Psi(\mathcal{X})$ is the set of polyhedral predicates over $\mathcal{X}$. $\mathbf{v} = (v_1, \ldots, v_n)$, denotes an element of $\mathbb{R}^n$, that captures clocks valuation, $v_i \in \mathbb{R}$, of every clock $x_i \in \mathcal{X}$. $v(x_i) = v_i$ corresponds to the value of x_i. We denote by region a subset of $\mathbb{R}^n$. For a region z and $x_i \in \mathcal{X}$, $z(x_i) = \{v_i | \mathbf{v} \in z\}$. $\psi(\mathbf{v})$ denotes the boolean function which equals **true** if the predicate ψ is satisfied by the input vector **v** and **false** if not. We denote by $[[\psi]]$, the region composed by the set of vectors $\mathbf{v} \in \mathbb{R}^n$, where the predicate ψ is **true** when we substitute each x_i by its corresponding v_i. $[[\psi]](x_i)$ denotes the interval of values captured by v_i, $\forall \mathbf{v} \in [[\psi]]$.

2.1 Constant Slope RHA

We start by defining Constant Slope Linear Hybrid Automata.

Definition 1. [26, 27, 28] A constant slope linear hybrid automaton (CSRHA) is a tuple $\mathcal{A} = (\mathcal{X}, \mathcal{Q}, \mathcal{T} \cup \{e_0\}, inv, dyn, guard, assign, l_0)$ where:

- $\mathcal{X}$, is a finite set of variables.
- $\mathcal{Q}$, is a finite set of locations.
- $\mathcal{T}$, is a finite set of transitions. A transition $e = (l, l') \in \mathcal{T}$, leads the system from the source location ,$l \in \mathcal{Q}$, to the end location, $l' \in \mathcal{Q}$. The entry transition of the initial state l_0 is denoted by e_0.
- $inv : \mathcal{Q} \to \Psi(\mathcal{X})$ is the location invariant, it associates a predicate to each location.
- $dyn : \mathcal{Q} \times \mathcal{X} \to \mathbb{R}$, is a function describing the evolution of variables. This evolution is usually of the form $l, \dot{x} = k$, $k \in \mathbb{R}$ or simply $\dot{x} = k$ in the location l. $\dot{\mathcal{X}}(l)$ denotes the evolution of all variables in the location l.
- $guard : \mathcal{T} \to \Psi(\mathcal{X})$ is the guard function. It associates a predicate, C_e to each transition, e. The guard, C_e should equals true to allow the execution of the transition e.
- $assign$, is the initialization function. It associates a relation ,$assign_e$ to each transition e defining the clocks to be reset.
- $l_0 \in \mathcal{Q}$, is the initial location.

■ The semantic of A constant slope rectangular hybrid automata (CSRHA) is given by the following definition.

2.2 Semantic of CSRHA

In this section, we focus on the semantic of a CSRHA in term of timed transition system, followed by the definition of run.

Definition 2. The semantic of a CSRHA $\mathcal{A} = (\mathcal{X}, \mathcal{Q}, \mathcal{T} \cup \{e_0\}, inv, dyn, guard, assign, l_0)$ is defined by a timed transition system $S_{\mathcal{A}} = (Q, q_0, \to)$ with

- $Q = \mathcal{Q} \times \mathbb{R}^n$ with $n = |\mathcal{X}|$.
- $q_0 = (l_0, init)$ is the initial state.
- $\to \in (Q \times (\mathcal{T} \cup \mathbb{R}_+) \times Q)$ is defined by :

∘ $(l, v) a_{\to} (l', v')$(**jump transition**) $if\ \exists\ e = (l, l') \in \mathcal{A}\ s.t. \begin{cases} a = e \\ guard(e)(v) = true \\ v' = assign_e(v) \\ inv(l')(v') = true \end{cases}$

∘ $(l, v)\varepsilon(t)_{\to} (l', v')$(**flow transition**) $if \begin{cases} l = l' \\ v' = v + t * \dot{\mathcal{X}}(l) \\ inv(l')(v') = true \end{cases}$

A run of CSRHA $\mathcal{A}$ is a path in $S_{\mathcal{A}}$ started from q_0. $[[\mathcal{A}]]$ denotes the set of all runs of $\mathcal{A}$. We note $(l, v)\varepsilon(t)_{\to}(l', v')a_{\to}(l'', v'')$ is equivalent to $(l, v) \to_a^{\varepsilon(t)} (l'', v'')$. A state (l_i, v_i) is considered as reachable, if $\exists (l_0, v_0) \to_{a_0}^{\varepsilon(t_0)} (l_1, v_1) \to_{a_1}^{\varepsilon(t_1)} (l_2, v_2) \to_{a_2}^{\varepsilon(t_2)} \ldots \to_{a_i}^{\varepsilon(t_i)} (l_i, v_i)$ where $(l_0, v_0) = q_0$.

A run $(l_0, v_0) \to_{a_0}^{\varepsilon(t_0)} (l_1, v_1) \to_{a_1}^{\varepsilon(t_1)} (l_2, v_2) \to_{a_2}^{\varepsilon(t_2)} \ldots \to_{a_i}^{\varepsilon(t_i)} (l_i, v_i) \ldots$ starting from $q_0 = (l_0, v_0)$ is a timed trace, denoted as $w = (a_0 = e_0, \delta_0) \to (a_1, \delta_1) \to (a_2, \delta_2) \to \cdots (a_2, \delta_2) \ldots$, where w is a sequence of pairs (a_i, δ_i), with $a_i \in \mathcal{T} \cup \{e_0\}$ a transition, and $\delta_{i+1} \in \mathcal{R}_+$ is the delay between the two successive events a_i and a_{i+1}, where : $\delta_0 = 0$, and $\forall i \geq 1$, $\delta_i = \varepsilon(t_i) - \varepsilon(t_{i-1})$.

3. Minimization of linear constraints

The state space in each location of an RHA is typically described by a polyhedron. In many cases, this space results of a computation made to fulfill a control, diagnose or general safety properties. Since these properties are often ensured by automated mechanism and algorithms, a hidden redundancy of inequalities is possible. A direct result is that the characterization of the reachable space becomes more difficult. It is more adequate to describe the polyhedron in a minimalistic way. Consider as example the following inequality system.

$$\begin{pmatrix} -1 & 1 \\ -1/4 & -1 \\ 3/2 & -1 \\ 1 & 1 \\ -3/4 & 1 \end{pmatrix} . \begin{pmatrix} x \\ y \end{pmatrix} \leq \begin{pmatrix} 0 \\ -3/4 \\ 4 \\ 6 \\ 2 \end{pmatrix}$$

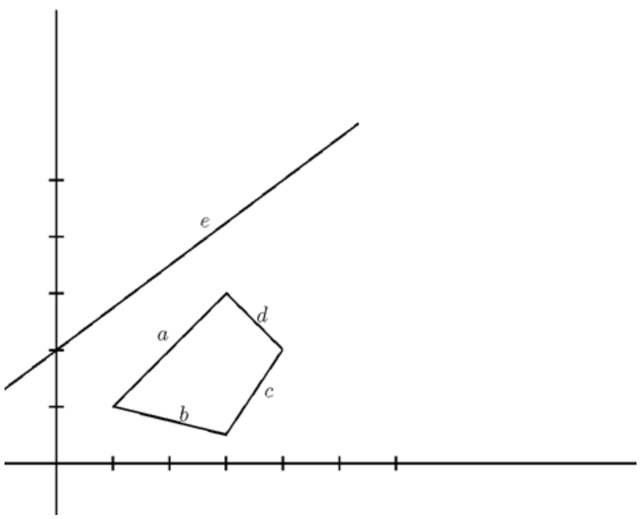

Fig. 1 Constraints representation

Figure 1 illustrates these constraints. The line segment **a** is the boundary of the first inequality: $-x + y \leq 0$. We note in the same way the other constraints. It is clear from the graphical representation of the polyhedron that the last inequality $(-3/4x + y \leq 2)$, whose boundary is represented by the line labeled **e**, has no influence on the representation of the polyhedron, and can be reduced by the system of inequalities. We try to check if the last inequality is useless/ redundant by a maximization problem. If we consider the following linear program:

$$f^* = maximize\left(-\frac{3}{4}\mathrm{x} + \mathrm{y}\right)$$
$$subject\ to \begin{pmatrix} -1 & 1 \\ -1/4 & -1 \\ 3/2 & -1 \\ 1 & 1 \end{pmatrix} . \begin{pmatrix} x \\ y \end{pmatrix} \leq \begin{pmatrix} 0 \\ -3/4 \\ 4 \\ 6 \end{pmatrix}$$
$$and - 3/4x + y \leq \mathbf{3}$$

Equivalent to

$$f^* = maximize\left(-\frac{3}{4}\mathrm{x} + \mathrm{y}\right)$$
$$subject\ to \begin{pmatrix} -1 & 1 \\ -1/4 & -1 \\ 3/2 & -1 \\ 1 & 1 \\ -3/4 & 1 \end{pmatrix} . \begin{pmatrix} x \\ y \end{pmatrix} \leq \begin{pmatrix} 0 \\ -3/4 \\ 4 \\ 6 \\ \mathbf{3} \end{pmatrix}$$

If the inequality $(-3/4x + y \leq 2)$ participates in the delimitation of the polyhedron space, then the maximization program will have a solution between 2 in the strict sense and 3. This comes from the fact that we relaxed the constant 2 of the constraint $(-3/4x + y \leq 2)$ by another strictly higher constant. In the case of this example, there will always be a solution of less than two.

In the following, we express a generalization that allows to check whether an inequality is redundant.

Let $A.X \leq b$ and $s^T x \leq t$ be the system with m + 1 inequalities of d variables $(x_1, x_2, \ldots, x_d)^T$. We want to test whether the system of the first m inequalities $A.X \leq b$ contains the last inequality $s^T x \leq t$. In the latter case the inequality $s^T x \leq t$ is considered redundant and can be removed from the system. A formulation of a linear program to verify this redundancy is as follows.

$$f^* = maximize\ s^T x$$
$$subject\ to\ A.X \leq b$$
$$and\ s^T x \leq \boldsymbol{t+1}$$

Theorem 1. Consider $X = (x_1, x_2, \ldots, x_d)^T$ and the following $m + 1$ inequality n system

$$A.X \leq b$$
$$and\ s^T x \leq t$$

The inequality $s^T x \leq t$ is considered non-redundant if for any non-negative constant c, the solution of the following maximization program

$$f^* = maximize\ s^T x$$
$$subject\ to$$
$$A.X \leq b$$
$$s^T x \leq t + \boldsymbol{c}$$

is bounded by $t < x \leq t + c$

If the solution is greater than t, this means that the inequality $s^T x \leq t$ is defining an effective edge of the polyhedron, so that adjusting the bound t will affect the form of the polyhedron. In another hand, if the solution is always less that or equal to t, this means that this edge is out of the polyhedron and it has no effect on the space definition. Thus, the inequality $s^T x \leq t$ is redondant iff the optimal value of f^* is less than or equal to t and then can be removed. We should repeat this linear program for all the other inequalities in order to check any possible redundancy.

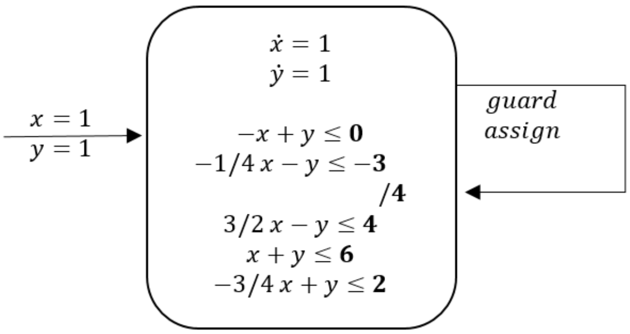

Fig. 2 CSRHA example

Figure 2 presents a simple CSRHA model where the location invariant in defined by a set of inequalities. In a quest of minimization we apply the maximization program mentioned in this section to each of the present inequalities. This results that the inequality $-3/4\,x + y \leq \mathbf{2}$ is redundant. The minimized version of this CSRHA is presented in Figure 3. Generally speaking, all computation actions on the minimized CSRHA will be less complex to perform.

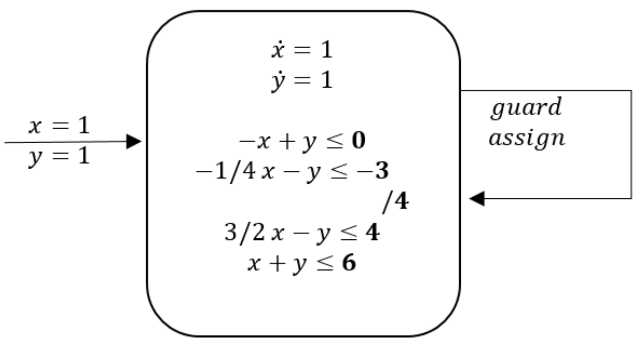

Fig. 3 Minimized CSRHA

4. Conclusion

The objective of this work is to reduce the number of constraints on a subclass of rectangular hybrid automata by some linear program technics. This will result in a more concise automaton where each constraint (on transition guards as well as location invariant) that remains is an effective constraint. Thus, the obtaining automata after applying our algorithm is minimal in term of constraints.

This technic is applicable on automata whose constraints are linear. Thus, the conjunction of these constraints results in a polyhedron. The CSRHA is a subclass of Hybrid automata that satisfy the linearity restriction and capable to describe a large category of real time systems as well as embedded systems.

In the future we aim to generalize this method to a more general format.

Acknowledgments

The authors wish to acknowledge the approval and the support of this research study by the grant N°7414-CIT-2017-1-8-F from the Deanship of the Scientific Research in Northern Border University, Arar, KSA.

References

[1] M. Kosmykov. Hybrid dynamics in large-scale logistics networks. PhD thesis, University of Bremen, 2011.
[2] J. Lygeros, K.H. Johansson, S.N. Simic, J. Zhang, and S.S. Sastry. Dynamical properties of hybrid automata. IEEE Transactions on Automatic Control, 48(1):2–17, 2003.
[3] R.G. Sanfelice. Robust Hybrid Control Systems. PhD thesis, University of California, Santa Barbara, 2007.
[4] A. van der Schaft and H. Schumacher. An introduction to hybrid dynamical systems Lecture Notes in Control and Information Sciences, volume 251. Springer-Verlag London Ltd., London, 2000.
[5] R.David and H. Alla. Petri nets for modeling of dynamic systems - a survey. Automatica, 30(1) 175–202, 1994.
[6] Hopcroft, John E., Rajeev Motwani, and Jeffrey D. Ullman. "Automata theory, languages, and computation." International Edition 24 (2006).
[7] D’Antoni, Loris, and Margus Veanes. "Forward bisimulations for nondeterministic symbolic finite automata." International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Springer, Berlin, Heidelberg, 2017.
[8] Merlin, Philip, and David Farber. "Recoverability of communication protocols--implications of a theoretical study." IEEE transactions on Communications 24.9 (1976): 1036-1043.
[9] Komenda, Jan, Sébastien Lahaye, and Jean-Louis Boimond. "Determinization of timed Petri nets behaviors." Discrete Event Dynamic Systems 26.3 (2016): 413-437.
[10] Berthomieu, Bernard, and Michel Diaz. "Modeling and verification of time dependent systems using time Petri nets." IEEE transactions on software engineering 17.3 (1991): 259-273.
[11] Boucheneb, Hanifa, et al. "Optimal Reachability in Cost Time Petri Nets." International Conference on Formal Modeling and Analysis of Timed Systems. Springer, Cham, 2017.
[12] Alur, Rajeev, and David L. Dill. "A theory of timed automata." Theoretical computer science 126.2 (1994): 183-235.
[13] Wang, Xinyu, et al. "Language Inclusion Checking of Timed Automata with Non-Zenoness." IEEE Transactions on Software Engineering (2017).
[14] Cassez, Franck, Peter Gjøl Jensen, and Kim Guldstrand Larsen. "Refinement of Trace Abstraction for Real-Time

Programs." International Workshop on Reachability Problems. Springer, Cham, 2017.

[15] Cassez, Franck, and Kim Larsen. "The impressive power of stopwatches." International Conference on Concurrency Theory. Springer, Berlin, Heidelberg, 2000.

[16] Cousot, Patrick. "Constructive design of a hierarchy of semantics of a transition system by abstract interpretation." Theoretical Computer Science 277.1-2 (2002): 47-103.

[17] Jing, Zhang, et al. "The real-time constraint semantics of events-driven state transition for embedded control systems." Control And Decision Conference (CCDC), 2017 29th Chinese. IEEE, 2017.

[18] Henzinger, Thomas A. "The theory of hybrid automata." Verification of Digital and Hybrid Systems. Springer Berlin Heidelberg, 2000. 265-292.

[19] Deng, Yi, et al. "Verification of hybrid automata diagnosability with measurement uncertainty." IEEE Transactions on Automatic Control 61.4 (2016): 982-993.

[20] Ren, Hao, et al. "Verification using counterexample fragment based specification relaxation: case of modular/concurrent linear hybrid automata." IET Cyber-Physical Systems: Theory & Applications 2.2 (2017): 65-74.

[21] Alur, Rajeev, and Erdinc Altug. "Linear Hybrid Automata." (2000).

[22] Henzinger, Thomas A., and Peter W. Kopke. "Discrete-time control for rectangular hybrid automata." Theoretical Computer Science 221.1 (1999): 369-392.

[23] Zhang, HaiBin, Cheng Zhao, and Rong Li. "A formal structure for symbolic reachability analysis of rectangular hybrid systems." Science China Technological Sciences 59.2 (2016): 347-356.

[24] Chutinan, Alongkrit, and Bruce H. Krogh. "Verification of polyhedral-invariant hybrid automata using polygonal flow pipe approximations." International workshop on hybrid systems: computation and control. Springer, Berlin, Heidelberg, 1999.

[25] Alur, Rajeev, et al. "Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems." Hybrid systems. Springer, Berlin, Heidelberg, 1993. 209-229.

[26] T. A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic model checking for real-time systems. Information and Computation, 111:394–406, 1994.

[27] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. Theoretical Computer Science, 138(1):3–34, 1995. (A preliminary version appeared in the Proceedings of the 11th International Conference on Analysis and Optimization of Systems: Discrete-Event Systems (ICAOS), Lecture Notes in Control and Information Sciences 199, Springer-Verlag, 1994, pp. 331-351.).

[28] Yamen El Touati, Saleh Altowaijri and Mohamed Ayari. Control of Industrial Systems to Avoid Failures: Application to Electrical System. International Journal of Advanced Computer Science and Applications(IJACSA), Vol. 9 No. 5, 2018. DOI : 10.14569/IJACSA.2018.090561

Yamen El Touati received his Engineering, M.S, and PhD degrees In Computer Science from the National School of Computer Science (ENSI), University of Manouba in 2003, 2005 and 2014, respectively. He is an assistant professor at ISAMM, University of Manouba, Tunisia and a permanent research member of the OASIS laboratory at the National School of Engineers of Tunis, University of Tunis El Manar, Tunis, Tunisia. Currently, he is on secondment as assistant professor in computer science at the Department of Computer Science, Faculty of Computing and Information Technology – Northern Border University (NBU) in the Kingdom of Saudi Arabia. His research interests include modelling, diagnosis and control supervision of dynamic hybrid systems and timed systems with various automata based and Petri-net based models. His research also, focuses on security and opacity issues for composed web services.