An Efficient Transformation from Max-Plus-Linear Systems to Piecewise Affine Systems

Dieky Adzkiya
Department of Mathematics
Institut Teknologi Sepuluh Nopember
Surabaya 60111, Indonesia
dieky@matematika.its.ac.id

Alessandro Abate
Department of Computer Science
University of Oxford
Oxford OX1 3QD, United Kingdom
aabate@cs.ox.ac.uk

***Abstract*—Max-Plus-Linear (MPL) systems are a modeling framework for synchronization without concurrency. In this paper, we propose a procedure to construct Piecewise Affine (PWA) systems from MPL systems, such that the regions in the PWA systems are a partition of the state space and without using any refinement process. On the contrary, in the literature the standard procedure uses a refinement process. According to our numerical benchmark, the procedure without refinement outperforms the one with refinement. The obtained PWA models are key for the analysis of the dynamics of the received MPL systems.**

***Index Terms*—Max-Plus-Linear systems, Piecewise Affine systems, Difference-Bound Matrices**

I. Introduction

Max-Plus-Linear (MPL) systems are a subclass of discrete-event systems to model synchronization phenomena without concurrency [1], [2]. These systems have been applied in many areas, such as computational systems [1], railway systems [2] and biological systems [3].

There are a lot of results over MPL systems on the literature. Classical results study the transient and steady-state behavior of MPL systems using algebraic or geometric approaches. Baccelli et al [1] discuss the computation of eigenvalue and eigenvectors of MPL systems by using algebraic techniques. The propagation of delay with applications to railway systems has been studied in [2]. The use of geometric techniques to synthesize a controller for MPL systems has been discussed in [4]. Recently, some authors have developed a framework for the formal verification of MPL systems [5]–[7]. The verification algorithms have been implemented in a freely available software tool VeriSiMPL [8], [9].

The paper [5] discusses finite abstraction of MPL systems. One of the key steps consists of constructing a partition of the state space based on the MPL dynamics. The partition is obtained by refining (via intersection) the regions of a Piecewise Affine (PWA) system generated by a given MPL system - the regions originally form a cover of the state space. PWA systems are characterized by a finite collection of convex polytopes, where the dynamics in each convex polytope is affine [10]. In this work, we propose a procedure to construct a PWA system from an MPL system such that the obtained regions of the PWA system already form a partition of the state space. The procedure is based on [5, Proposition 3], which states that the difference between two adjacent regions is a Difference-Bound Matrix (DBM). DBM is a subclass of polyhedra where the inequalities are described either as the difference of two variables or as a single variable [11].

This paper is structured as follows. Section II discusses some modeling frameworks and current results in the literature. The proposed transformation algorithm and its application to a simple case study is discussed in Section III. Finally, Section V describes concluding remarks.

II. Models and Preliminaries

In this section, we describe some models and some related concepts. First, we introduce Max-Plus-Linear (MPL) systems in Section II-A. Then, the notion of Difference-Bound Matrices (DBM) is described in Section II-B. Finally, in Section II-C we discuss the known procedure to transform an MPL system to a Piecewise Affine (PWA) system.

A. Max-Plus-Linear Systems

First of all, we define $\mathbb{R}$ as the set of real numbers, $\varepsilon := -\infty$ and $\mathbb{R}_\varepsilon := \mathbb{R} \cup \{\varepsilon\}$. Then we also define the following two binary operators

$$a \oplus b := \max\{a, b\}, \qquad a \otimes b := a + b,$$

where $a, b \in \mathbb{R}_\varepsilon$. The structure $(\mathbb{R}_\varepsilon, \oplus, \otimes)$ is called max-plus algebra [1], [2]. Notice that ε and 0 are the neutral element of $\oplus$ and $\otimes$, respectively. The definition of $\oplus$ and $\otimes$ can be extended to matrices and vectors, as in the classical algebra:

$$[A \oplus B](i,j) = A(i,j) \oplus B(i,j),$$
$$[A \otimes C](i,j) = \bigoplus_{k=1}^{n} A(i,k) \oplus B(k,j),$$

where $A, B \in \mathbb{R}_\varepsilon^{m\times n}$ and $C \in \mathbb{R}_\varepsilon^{n\times p}$. Notation $A(i,j)$ represents the entry of matrix A at row i column j.

An autonomous Max-Plus-Linear (MPL) system is defined as:

$$x(k+1) = A \otimes x(k), \tag{1}$$

where state vector $x \in \mathbb{R}^n$ and state matrix $A \in \mathbb{R}_\varepsilon^{n\times n}$. In MPL systems, the independent variable k represents the occurrence index of events, whereas $x(k)$ represents the time of k-th occurrence of all events. A matrix in $\mathbb{R}_\varepsilon^{n\times n}$ is called row finite if there is at least one finite entry in each row.

Example 1. *Throughout this paper, we use the following two-dimensional MPL system that represents a simple railway network [2]:*

$$x(k+1) = \begin{bmatrix} 2 & 5 \\ 3 & 3 \end{bmatrix} \otimes x(k) \tag{2}$$

This work has been supported by Penelitian Dana Departemen 1145/PKS/ITS/2017. D. Adzkiya is also with the PUI – PT Mechatronics and Industrial Automation – Research Center, Institut Teknologi Sepuluh Nopember, Kampus ITS Sukolilo-Surabaya 60111, Indonesia.

B. Difference-Bound Matrices

Difference-Bound Matrices (DBM) are a subclass of polyhedra, where the inequalities are in the form of the difference of two variables or a single variable [11]. DBM has been used in the verification of timed automata [12] and finite abstraction of MPL systems [5]. The formal definition of DBM is as follows

Definition 1 (Difference-Bound Matrices [11])**.** *A Difference-Bound Matrix in $\mathbb{R}^n$ is the intersection of finitely many sets defined as $x_j - x_i \bowtie_{i,j} \alpha_{i,j}$ where $\bowtie_{i,j} \in \{<, \leq\}$ represents a strict and nonstrict inequality sign, $\alpha_{i,j} \in \mathbb{R} \cup \{+\infty\}$ denotes the upper bound, for $i, j \in \{0, \ldots, n\}$ and value of the special variable x_0 is always equal to 0. The sets are subsets of $\mathbb{R}^n$ that are characterized by the values of variables $x_1, \ldots, x_n$.*

DBM has an advantage compared to polyhedra in terms of the computational complexity. More precisely, in polyhedra, the time complexity of many operations is exponential. On the other hand, many operations over DBM have polynomial time complexity, such as computing the intersection of two DBM, checking whether a DBM is empty, computing the complement of a DBM.

C. Piecewise Affine Systems Generated by Max-Plus-Linear Systems

Piecewise Affine (PWA) systems are defined by multiple affine dynamics, where each affine dynamics is active on a predefined polyhedron. In this section, we discuss the transformation from MPL systems to PWA systems (with refinement), such that the PWA regions are a partition of the state space $\mathbb{R}^n$ [5]. The transformation consists of two steps: construction of PWA systems where the PWA regions are not pairwise disjoint and refinement of PWA regions.

First, we discuss the construction of PWA systems from MPL systems [5]. The autonomous MPL system in (1) can be expressed as a PWA system in the event domain [13] under the condition that the state matrix is row finite. The regions and the corresponding affine dynamics are characterized by coefficients $g = (g_1, \ldots, g_n) \in \{1, \ldots, n\}^n$ [5], [9]. For each $i \in \{1, \ldots, n\}$, the coefficient g_i represents the maximal term in the i-th state equation $x_i(k) = \max\{A(i,1) + x_1, \ldots, A(i,n) + x_n\}$, that is $A(i,j) + x_j \leq A(i,g_i) + x_{g_i}$ for all $j \in \{1, \ldots, n\}$. Here, x_j represents the j-th entry of vector x.

By using the above scheme, the states corresponding to coefficients g, denoted by R_g, can be expressed explicitly as follows

$$R_g = \bigcap_{i=1}^{n} \bigcap_{j \neq g_i} \{x \in \mathbb{R}^n : A(i,j) + x_j \leq A(i,g_i) + x_{g_i}\}. \quad (3)$$

The affine dynamics that is active in R_g is given by

$$x_i(k) = x_{g_i}(k-1) + A(i,g_i), \qquad i \in \{1, \ldots, n\}.$$

The complete algorithm to generate the collection of regions in PWA systems generated by MPL systems is as follows.

Require: row finite state matrix $A \in \mathbb{R}_\varepsilon^{n \times n}$
Ensure: collection of PWA regions R
  initialize R with the empty set, i.e. $\mathsf{R} \leftarrow \emptyset$
  for all $g \in \{1, \ldots, n\}^n$ **do**
    generate R_g according to (3)
    if R_g is not empty **then**
      store R_g in R, i.e. $\mathsf{R} \leftarrow \mathsf{R} \cup \{R_g\}$
    end if
  end for

In general, the regions of the PWA system generated by an MPL system are not pairwise disjoint. We can use the refinement procedure described in [5, Fig. 6] to obtain a partition of the state space. The procedure uses the notion of adjacent regions, which is formally defined as follows.

Definition 2 (Adjacent Regions [5, Definition 9])**.** *Let R_g and $R_{g'}$ be regions generated by an n-dimensional state space matrix. The two regions are adjacent, denoted by $R_g > R_{g'}$, if there exists a single $1 \leq i \leq n$ such that $g_i > g'_i$ and $g_j = g'_j$ for each $j \neq i$.*

The refinement algorithm works as follows. For any pair of adjacent regions, the refinement procedure removes the intersection from the region corresponding to lower coefficient [5, Fig. 6], as shown below.

Require: a collection of PWA regions R
Ensure: a collection of partitioning regions R
  for all $R_g \in \mathsf{R}$ **do**
    for all $R_{g'} \in \mathsf{R} \setminus \{R_g\}$ **do**
      if $R_g > R_{g'}$ **then**
        $R_{g'} \leftarrow R_{g'} \setminus R_g$
      else if $R_{g'} > R_g$ **then**
        $R_g \leftarrow R_g \setminus R_{g'}$
      end if
    end for
  end for

III. Proposed Transformation Procedure

In this section, we describe the procedure to construct a PWA system from an MPL system such that the regions in the PWA system are a partition of the state space. Then, we illustrate the procedure on a simple example.

First of all, the procedure relies on Proposition 1, which states that the difference of two adjacent regions is a DBM.

Proposition 1 (Difference of Adjacent Regions [5, Proposition 3])**.** *If $R_g > R_{g'}$, then $R_{g'} \setminus R_g$ is a DBM described by $R_{g'} \cap \{x \in \mathbb{R}^n : A(i,g'_i) + x_{g'_i} > A(i,g_i) + x_{g_i}\}$.*

According to Proposition 1 and the refinement procedure of PWA regions (cf. Section II-C), the PWA regions that form a partition are a slight modification of (3). Remember that in [5, p. 3045], the intersection is assigned to the region corresponding to higher coefficient.

$$\begin{aligned}\bar{R}_g = &\bigcap_{i=1}^{n} \bigcap_{j<g_i} \{x \in \mathbb{R}^n : A(i,j) + x_j \leq A(i,g_i) + x_{g_i}\} \cap \\ &\bigcap_{i=1}^{n} \bigcap_{j>g_i} \{x \in \mathbb{R}^n : A(i,j) + x_j < A(i,g_i) + x_{g_i}\}. \quad (4)\end{aligned}$$

In (4), there are two conditions:

- If a region (j) has coefficient smaller than this region (g_i), namely $j < g_i$, then the intersection is assigned to this region. In this case, the form is the same as in (3) and the inequalities are not strict.
- If a region (j) has coefficient higher than this region (g_i), namely $j > g_i$, then the intersection is not assigned to

this region. Thus, the intersection has to be removed from this region. In this case, according to Proposition 1, the inequalities are strict.

The complete algorithm to compute the collection of regions in PWA systems generated by MPL systems, and such that the PWA regions are a partition of the state space, is as follows.

Require: row finite state matrix $A \in \mathbb{R}_\varepsilon^{n\times n}$
Ensure: collection of partitioning regions $\bar{\mathsf{R}}$
initialize $\bar{\mathsf{R}}$ with the empty set, i.e. $\bar{\mathsf{R}} \leftarrow \emptyset$
for all $g \in \{1,\dots,n\}^n$ **do**
  generate $\bar{R}_g$ according to (4)
  if $\bar{R}_g$ is not empty **then**
    store $\bar{R}_g$ in $\bar{\mathsf{R}}$, i.e. $\bar{\mathsf{R}} \leftarrow \bar{\mathsf{R}} \cup \{\bar{R}_g\}$
  end if
end for

Example 2. *When we apply the procedure to the MPL system in* (1)*, we obtain* $\bar{R}_{(1,1)} = \{x \in \mathbb{R}^2 \mid x_1 - x_2 > 3\}$, $\bar{R}_{(1,2)} = \emptyset$, $\bar{R}_{(2,1)} = \{x \in \mathbb{R}^2 \mid 0 < x_1 - x_2 \leq 3\}$ *and* $\bar{R}_{(2,2)} = \{x \in \mathbb{R}^2 \mid x_1 - x_2 \leq 0\}$. *The graphical representation is shown in Fig. 1.*

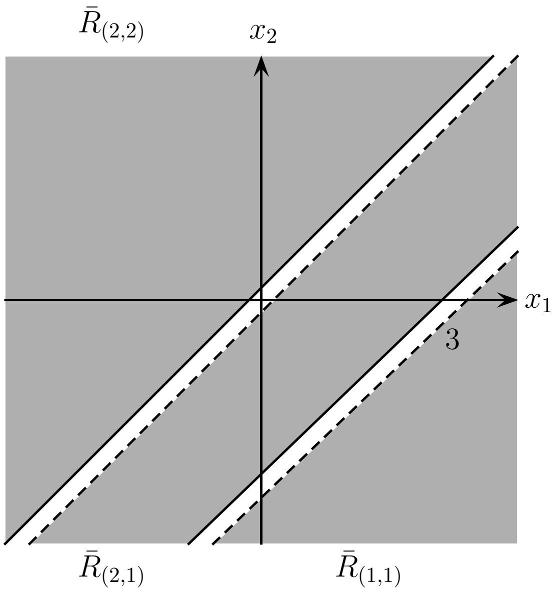

Fig. 1. Graphical representation of the partition generated by the autonomous MPL system in (1).

IV. Computational Benchmark

In this section, we compare the computational time of the procedure to construct a partition without refinement discussed in Section III against the procedure to construct a partition with refinement discussed in [5]. Our scenario is as follows. For each dimension n, we generate 10 random full matrices A (i.e. all entries are finite), where each entry is an integer between 1 and 100. Then we determine the mean and maximal computational time over 10 experiments. The algorithms have been implemented in MATLAB by using the backtracking technique to improve the performance. The experiments have been run on an Intel® Core™ i7-4720HQ CPU @2.60GHz with 16 GB of memory.

As you can see from Table I, the computational time of the proposed procedure (without refinement) is significantly faster compared to the procedure with refinement from [5].

TABLE I
Mean and maximal computational time of constructing PWA systems from MPL systems over 10 experiments.

	With refinement		Without refinement	
n	**Mean**	**Maximum**	**Mean**	**Maximum**
3	0.010002 [s]	0.012857 [s]	0.0051881 [s]	0.00614 [s]
4	0.0415318 [s]	0.044975 [s]	0.033309 [s]	0.037264 [s]
5	0.2236356 [s]	0.240583 [s]	0.1120053 [s]	0.117824 [s]
6	1.2117465 [s]	1.31256 [s]	0.5257748 [s]	0.540758 [s]
7	10.4736843 [s]	11.142775 [s]	2.3186673 [s]	2.372729 [s]

V. Conclusions and Future Work

We have proposed an algorithm to generate a Piecewise-Affine (PWA) system from a Max-Plus-Linear (MPL) system such that the regions are a partition of the state space. In order to speed up the computational time, the algorithm does not use any refinement process. This fact has been shown empirically through our computational benchmark. We are planning to integrate the procedure to the next version of VeriSiMPL.

References

[1] F. Baccelli, G. Cohen, G. Olsder, and J.-P. Quadrat, *Synchronization and Linearity, An Algebra for Discrete Event Systems*. John Wiley and Sons, 1992.

[2] B. Heidergott, G. Olsder, and J. van der Woude, *Max Plus at Work–Modeling and Analysis of Synchronized Systems: A Course on Max-Plus Algebra and Its Applications*. Princeton University Press, 2006.

[3] C. A. Brackley, D. S. Broomhead, M. C. Romano, and M. Thiel, "A max-plus model of ribosome dynamics during mRNA translation," *Journal of Theoretical Biology*, vol. 303, no. 0, pp. 128–140, Jun. 2012.

[4] R. D. Katz, "Max-plus (A, B)-invariant spaces and control of timed discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 52, no. 2, pp. 229–241, Feb. 2007.

[5] D. Adzkiya, B. De Schutter, and A. Abate, "Finite abstractions of max-plus-linear systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 12, pp. 3039–3053, Dec. 2013.

[6] ——, "Computational techniques for reachability analysis of max-plus-linear systems," *Automatica*, vol. 53, no. 0, pp. 293–302, 2015.

[7] S. Esmaeil Zadeh Soudjani, D. Adzkiya, and A. Abate, "Formal verification of stochastic max-plus-linear systems," *IEEE Transactions on Automatic Control*, vol. 61, no. 10, pp. 2861–2876, Oct. 2016.

[8] D. Adzkiya and A. Abate, "VeriSiMPL: Verification via biSimulations of MPL models," in *Proceedings of the 10th International Conference on Quantitative Evaluation of Systems (QEST'13)*, ser. Lecture Notes in Computer Science, K. Joshi, M. Siegle, M. Stoelinga, and P. D'Argenio, Eds., vol. 8054. Springer, Heidelberg, Sep. 2013, pp. 253–256. [Online]. Available: http://sourceforge.net/projects/verisimpl/

[9] D. Adzkiya, Y. Zhang, and A. Abate, "VeriSiMPL 2: An open-source software for the verification of max-plus-linear systems," *Discrete Event Dynamic Systems*, vol. 26, no. 1, pp. 109–145, 2016.

[10] E. D. Sontag, "Nonlinear regulation: The piecewise-linear approach," *IEEE Transactions on Automatic Control*, vol. 26, no. 2, pp. 346–358, Apr. 1981.

[11] D. Dill, "Timing assumptions and verification of finite-state concurrent systems," in *Automatic Verification Methods for Finite State Systems*, ser. Lecture Notes in Computer Science, J. Sifakis, Ed. Springer, Heidelberg, 1990, vol. 407, ch. 17, pp. 197–212.

[12] R. Alur and D. Dill, "A theory of timed automata," *Theoretical Computer Science*, vol. 126, no. 2, pp. 183–235, 1994.

[13] W. Heemels, B. De Schutter, and A. Bemporad, "Equivalence of hybrid dynamical models," *Automatica*, vol. 37, no. 7, pp. 1085–1091, Jul. 2001.