Abstraction and Verification of Autonomous Max-Plus-Linear Systems

Dieky Adzkiya, Bart De Schutter, and Alessandro Abate

***Abstract*—This work investigates the use of finite abstractions for the verification of autonomous Max-Plus-Linear (MPL) models. Abstractions are characterized as finite-state labeled transition systems (LTS) and are obtained by first partitioning the state space of the MPL and associating states of the LTS to the partitions, then by defining relations among the vertices of the LTS, corresponding to dynamical transitions between the MPL state partitions, and finally by labeling the LTS edges according to one-step time properties of the events of the MPL model. In order to establish formal equivalences, the finite LTS abstraction is proven to either simulate or to bisimulate the original MPL model, the difference depending on its determinism. The computational performance of the abstraction procedure is tested on a benchmark. The work then studies properties of the original MPL model by verifying equivalent specifications on the finite LTS abstraction.**

I. Introduction

Max-Plus-Linear (MPL) systems are discrete-event models [1], [2] with a continuous state space characterizing the timing of the underlying discrete events (cfr. Section II). Classical dynamical analysis of MPL models is grounded on their algebraic [3] or geometric properties [4]. This work investigates a novel approach based on finite-state abstractions of autonomous MPL models, on the expression of general dynamical properties as specifications in a modal logic, and on the formal verification of such properties by model checking.

With regards to the abstraction procedure (cfr. Section III), we put forward a new technique that generates a finite-state labeled transition system (LTS). The vertices of the LTS are obtained by finite partitioning of the state space of the MPL model, whereas relations between two LTS states are defined by checking whether a trajectory of the original MPL model can transition between the corresponding partition regions. The obtained finite-state transition system can be either nondeterministic or deterministic. We prove that the first instance simulates the original MPL model, whereas the second one bisimulates it. Given a nondeterministic transition system, a refinement procedure based on state partitioning can attempt to generate a deterministic one, however this approach is in general nondecidable. Finally, the labels of the LTS model are defined in two possible ways: they either characterize the difference between the timing of an event for any two variables of the original model, or represent the time difference between consecutive events of the MPL model. The computational performance of the abstraction procedure is benchmarked (cfr. Section III-D) on a case study.

With focus on system properties, we advocate the use of Linear Temporal Logic (LTL) [5] to express time-dependent properties of the original MPL model. In particular, we focus on properties related to the cyclicity, transient time, and asymptotic behavior of the MPL model (cfr. Section IV). LTL specifications over LTS can be efficiently model checked by a number of existing software tools – in this work we use the SPIN model checker [6] for our purposes. Section IV elaborates an example to display the overall approach.

The authors are with the Delft Center for Systems and Control, TU Delft - Delft University of Technology, The Netherlands - {d.adzkiya,b.deschutter,a.abate}@tudelft.nl

This work is supported by the European Commission MoVeS project FP7-ICT-2009-5 257005, by the European Commission Marie Curie grant MANTRAS 249295, by the European Commission NoE HYCON2 FP7-ICT-2009-5 257462, and by the NWO VENI grant 016.103.020.

II. Models and Preliminaries

This section introduces the definition of an MPL model, recalls a few of its basic properties, and presents LTS models.

A. Max-Plus-Linear Systems

Define $\mathbb{R}_\epsilon$, ϵ and e respectively as $\mathbb{R} \cup \{\epsilon\}$, $-\infty$ and 0. A vector with each component equal to 0 (or $-\infty$) is also denoted by e (resp., ϵ). For $x, y \in \mathbb{R}_\epsilon$, we define $x \oplus y = \max\{x, y\}$ and $x \otimes y = x + y$. Let $G = A \oplus B$ (or $H = C \otimes D$), for matrices $A, B \in \mathbb{R}_\epsilon^{m \times n}$ (or $C \in \mathbb{R}_\epsilon^{m \times p}, D \in \mathbb{R}_\epsilon^{p \times n}$), then $G(i,j) = A(i,j) \oplus B(i,j)$ (or $H(i,j) = \bigoplus_{k=1}^{p} C(i,k) \otimes D(k,j)$), for $1 \le i \le m$ and $1 \le j \le n$. (Notice the analogy between $\oplus$, $\otimes$ and $+$, $\times$ for matrix and vector operations in standard algebra.) Given $r \in \mathbb{R}$ (or $m \in \mathbb{N}$), the max-algebraic power of $x \in \mathbb{R}$ (resp., $A \in \mathbb{R}_\epsilon^{n \times n}$) is denoted by $x^{\otimes^r}$ (or $A^{\otimes^m}$) and corresponds to rx in conventional algebra (resp., $A \otimes \cdots \otimes A$, m times). Notice that $A^{\otimes^0} = E_n$, where E_n is a max-plus identity matrix, i.e. $E_n(i,i) = e$ and $E_n(i,j) = \epsilon$, for $1 \le i \ne j \le n$.

An autonomous MPL model [7, p. 47] is defined as:

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

where $A \in \mathbb{R}_\epsilon^{n \times n}$, $x(k) \in \mathbb{R}_\epsilon^n$ for $k \in \mathbb{N}$. The independent variable k denotes an increasing discrete-event counter, whereas the state variable defines the (continuous) timing of the discrete events. For practical reasons, in this work we define the state space as $\mathbb{R}^n$ and assume to be working with a regular or row-finite max-plus matrix $A \in \mathbb{R}_\epsilon^{n \times n}$, namely with a matrix characterized by at least one element different from ϵ in each row. Related to matrix A is the notion of precedence (or communication) graph.

Definition 1 (Precedence graph, [8, Definition 2.8]): Consider $A \in \mathbb{R}_\epsilon^{n \times n}$. The precedence graph of A, denoted by $\mathcal{G}(A)$, is a weighted directed graph with vertices $1, \ldots, n$ and an arc (j, i) with weight $A(i,j)$ for each $A(i,j) \ne \epsilon$. □

Example: Consider the two-dimensional MPL model from [7, Section 0.1]:

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

the precedence graph of A is shown in Figure 1. □

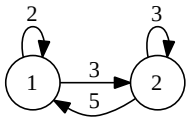

Fig. 1. Precedence graph for the MPL model in (2)

The notion of irreducible matrix, to be used shortly, can be given via a precedence graph.

Definition 2 (Irreducible matrix, [7, Theorem 2.14]): A matrix $A \in \mathbb{R}_\epsilon^{n\times n}$ is called irreducible if its precedence graph $\mathcal{G}(A)$ is strongly connected. □

Recall that a directed graph is strongly connected iff for any two pairs of different vertices i, j of the graph, there exists a path from i to j. From a max-plus algebraic perspective, a matrix $A \in \mathbb{R}_\epsilon^{n\times n}$ is irreducible if the nondiagonal elements of $\bigoplus_{k=1}^{n-1} A^{\otimes^k}$ are finite (not equal to ϵ).

Example: For the preceding example in (2), since $A(1,2) \neq \epsilon \neq A(2,1)$, matrix A is irreducible. Equivalently, notice that the precedence graph in Figure 1 is strongly connected. □

Proposition 1 (Cyclicity, [7, Theorem 3.9]): Let $A \in \mathbb{R}_\epsilon^{n\times n}$ be an irreducible matrix with eigenvalue λ and cylicity c, there is a $k_0 \in \mathbb{N}$, such that $A^{\otimes^{k+c}} = \lambda^{\otimes^c} \otimes A^{\otimes^k}$, for all $k \geq k_0$. □

Proposition 1 allows to establish the existence of a periodic regime. Given an initial condition $x(0) \in \mathbb{R}^n$, we can seek a $k_0(x(0))$ (the transient time), which is in general less conservative than the $k_0 = k_0(A)$ in Proposition 1.

Example: In the previous numerical example, characterized by an irreducible matrix A in (2), we have eigenvalue $\lambda = 4$, cyclicity $c = 2$, and transient $k_0(A) = 2$. □

From a graph theoretical point of view, the max-plus eigenvalue λ is defined as the maximum cycle mean of the precedence graph [8, Theorem 3.23]. Fast algorithms have been developed to compute this quantity [9, Section 4]. An upper bound for the transient time k_0 and its computation have been discussed in [10, Theorem 10,13].

In order to investigate the asymptotic behavior of an MPL model, we employ the concept of critical graph, which is constructed from the precedence graph.

Definition 3 (Critical graph, cyclicity [8, Definition 3.94]): For a matrix $A \in \mathbb{R}_\epsilon^{n\times n}$, the following notions are defined:

- A circuit ζ of the precedence graph $\mathcal{G}(A)$ is called *critical* if it has maximum average weight.
- The *critical graph* $\mathcal{G}^c(A)$ consists of those nodes and arcs of $\mathcal{G}(A)$ which belong to a critical circuit of $\mathcal{G}(A)$.
- The *cyclicity* of a strongly connected graph is the greatest common divisor of the lengths of all its circuits. The cyclicity of a general graph is the least common multiple of the cyclicities of all its strongly connected subgraphs.

□

Example: The MPL model in (2) admits the critical circuit $\{1 \to 2 \to 1\}$, thus the critical graph coincides with the critical circuit. Since the critical graph is strongly connected, the eigenvector is unique and the eigenspace is the multiplication (in a max-plus sense) of the eigenvector with a finite constant. The cyclicity of the critical graph is 2, as also results from Proposition 1, thus there exists a periodic regime with period 2. □

Definition 4 (Max-algebraic linear systems, [11, Section 2]): A max-algebraic linear system is defined as $A \otimes x = b$, where $A \in \mathbb{R}_\epsilon^{m\times n}$, $b = [b_1 \dots b_m]^T \in \mathbb{R}^m$, $x = [x_1 \dots x_n]^T \in \mathbb{R}^n$. If $b = e$, the max-algebraic linear system is normalized. □

The projective space is a quotient space generated by an equivalence relation $\sim$ [7, Section 1.4]. For each $x, y \in \mathbb{R}_\epsilon^n$, $x \sim y$ iff there is an $\alpha \in \mathbb{R}$, such that $y = \alpha \otimes x$. We use a bar to distinguish between an element x and its equivalence class $\bar{x}$. Formally, $\bar{x} = \{y \in \mathbb{R}_\epsilon^n : y \sim x\}$, for each $x \in \mathbb{R}_\epsilon^n$.

B. Labeled Transition Systems

Definition 5 ([5, Definition 2.1]): A Labeled Transition System (LTS) (S, L, δ, I, AP) consists of a set S of states, a set L of labels, a transition relation $\delta \subseteq S \times L \times S$, a set $I \subseteq S$ of initial states and a set AP of atomic propositions. □

Denote with $\mathcal{P}(S)$ the power set of a given set S. This work considers a special class of LTS. To begin with, often we will assume that $I = S$ and $AP = \mathbb{R}^n$. Moreover the set of labels, customarily taken to be discrete and finite, is here assumed to be $L = \mathcal{P}(AP)$ – in other words, the labels will be n-dimensional vectors of real numbers or of real-valued intervals.

III. LTS Abstractions of MPL Models

We abstract an MPL model into a finite-state LTS. The states of the LTS are obtained by partitioning the continuous state space, whereas its transitions depend on the dynamics between the partition regions. The labels of the LTS are defined in two distinguished manners and depend on the delay (timing) features of the MPL model. We derive exact equivalences or preorder relations between the LTS abstraction and the concrete MPL model. Finally, we test the abstraction procedure on a computational benchmark.

A. LTS States: Partitioning Procedure

We put forward two different approaches to partition the state space, then prove their equivalence: the first leverages a piecewise-affine approach, whereas the second is directly based on the MPL model.

1) Partitioning via PWA Expression: Each autonomous MPL model as in (1) can be expressed as a piecewise-affine (PWA) system in the event domain [12, Section 3]. The PWA

dynamics in each region are characterized by $(g_1, \ldots, g_n) \in \{1, \ldots, n\}^n$ or, more precisely, as:

$$x_i(k+1) = x_{g_i}(k) + A(i, g_i), \quad 1 \leq i \leq n, \tag{3}$$

where the value of $g_1, \ldots, g_n$ depends on $x(k)$.

We determine a covering of the MPL state space by constructing the collection of the regions corresponding to the PWA dynamics, as Algorithm 1 details (see also next paragraph).

Algorithm 1: Generation of state-space covering via piecewise-affine model

input : $A \in \mathbb{R}_\epsilon^{n \times n}$, a regular max-plus matrix
output: R, a collection of regions

$\mathsf{R} \leftarrow \emptyset$;
$fin(i) \leftarrow \{j : A(i,j) > \epsilon\}, \quad i = 1, \ldots, n$;
foreach $(g_1, \ldots, g_n) \in fin(1) \times \cdots \times fin(n)$ **do**
  $R_g \leftarrow \mathbb{R}^n$;
  for $i \leftarrow 1$ **to** n **do**
    foreach $j \in fin(i) \setminus \{g_i\}$ **do**
      $R_g \leftarrow R_g \cap \{x \in \mathbb{R}^n : A(i, g_i) + x_{g_i} \geq A(i,j) + x_j\}$;
    end
  end
  if $R_g \neq \emptyset$ **then** $\mathsf{R} \leftarrow \mathsf{R} \cup \{R_g\}$;
end

Notice that, since this collection is in general not pairwise disjoint, further refinement is needed to construct a partition: namely, if there are two intersecting regions, then we remove their intersection and create a new region defined as the intersection.

2) Partitioning via MPL Model: As discussed, each point in $\mathbb{R}^n$ possibly corresponds to more than one PWA model. We now directly determine a partitioning of the state space using a perturbation analysis over the MPL model. Given an autonomous MPL model characterized by a regular max-plus matrix $A \in \mathbb{R}_\epsilon^{n \times n}$ and a generic $x(0) \in \mathbb{R}^n$, we are interested in finding a perturbation vector $p = [p_1 \ldots p_n]^T$, such that $A \otimes (x(0) + p) = x(1) = A \otimes x(0)$. This equation reduces to the following normalized max-algebraic linear system: $A_p \otimes p = e$, where $A_p(i,j) = A(i,j) + x_j(0) - x_i(1)$, for $1 \leq i, j \leq n$. Notice that each element of A_p is nonpositive and that there exists a nonempty set of null (e) elements in each of its rows.

Notice that the PWA dynamics (3) of $x(0) \in \mathbb{R}^n$ are characterized by $(g_1, \ldots, g_n)$ iff $A_p(i, g_i) = e$, for $1 \leq i \leq n$. The region characterized by $(g_1, \ldots, g_n)$ in the PWA approach is the set of points $x(0) \in \mathbb{R}^n$ such that $A_p(i, g_i) = e$ for $1 \leq i \leq n$. We can characterize the region in the MPL approach by $(f_1, \ldots, f_n) \in (\mathcal{P}(\{1, \ldots, n\}) \setminus \emptyset)^n$, where $f_i = \{j : A_p(i,j) = e\}$, for $1 \leq i \leq n$. More precisely, the region characterized by $(f_1, \ldots, f_n)$ is defined as the set of $x(0) \in \mathbb{R}^n$ such that the matrix A_p corresponds to $(f_1, \ldots, f_n)$. Algorithm 2 details the procedure generating the collection of regions.

Algorithm 2: Generation of state-space partition via MPL model

input : $A \in \mathbb{R}_\epsilon^{n \times n}$, a regular max-plus matrix
output: R, a collection of regions

$\mathsf{R} \leftarrow \emptyset$;
foreach $(f_1, \ldots, f_n) \in (\mathcal{P}(\{1, \ldots, n\}) \setminus \emptyset)^n$ **do**
  $R_f \leftarrow \mathbb{R}^n$;
  for $r \leftarrow 1$ **to** n **do**
    foreach $1 \leq i \neq j \leq n$ **do**
      if $i, j \in f_r$ **then**
        $R_f \leftarrow R_f \cap \{x \in \mathbb{R}^n : x_i - x_j = A(r,j) - A(r,i)\}$;
      else if $i \in f_r$, $j \notin f_r$ **then**
        $R_f \leftarrow R_f \cap \{x \in \mathbb{R}^n : x_i - x_j > A(r,j) - A(r,i)\}$;
    end
  end
  if $R_f \neq \emptyset$ **then** $\mathsf{R} \leftarrow \mathsf{R} \cup \{R_f\}$;
end

The following property characterizes the output of Algorithm 2:

Proposition 2: The collection of regions generated by the MPL approach is a partition on $\mathbb{R}^n$. □

Algorithm 2 allows to derive a general representation of each region, which can be expressed as a set of linear inequalities of the following form:

$$\alpha \simeq x_i - x_j \simeq \beta, \tag{4}$$

where $1 \leq i < j \leq n; \alpha, \beta \in \mathbb{R}_\epsilon \cup \{-\epsilon\}$; and $\simeq$ is $<$ if $\alpha \neq \beta$, whereas $\simeq$ is $\leq$ if $\alpha = \beta$.

From the definition of regions obtained using the perturbation analysis, we conclude that:

Lemma 1: Each region generated by the PWA approach is a union of regions generated by the MPL approach. □

This allows to conclude that for each region f generated by the MPL approach, there exists a region g generated by the PWA approach such that $f \subseteq g$ (recall that the collection of regions generated by the PWA approach is a cover of $\mathbb{R}^n$).

We are ready to formulate the main result of this section.

Proposition 3: The partitions constructed by the PWA approach after refinement coincide with those obtained by the MPL approach. □

Example: Consider the MPL model defined in (2), the covering region characterized by $(1,1)$ is $\{x \in \mathbb{R}^2 : x_1 - x_2 \geq 3\}$ and it is generated by the PWA approach. By using the MPL approach, the partitioning region characterized by $(\{1\}, \{1\})$ and $(\{1,2\}, \{1\})$ are $R_3 = \{x \in \mathbb{R}^2 : x_1 - x_2 > 3\}$ and $R_4 = \{x \in \mathbb{R}^2 : x_1 - x_2 = 3\}$ respectively. Observe that the covering region characterized by $(1,1)$ is a union of partitioning regions characterized by $(\{1\}, \{1\})$ and $(\{1,2\}, \{1\})$. We obtain a total of 5 partitioning regions (Figure 2), where the remaining

partitioning regions are $R_1 = \{x \in \mathbb{R}^2 : x_1 - x_2 < e\}$, $R_2 = \{x \in \mathbb{R}^2 : x_1 - x_2 = e\}$ and $R_5 = \{x \in \mathbb{R}^2 : e < x_1 - x_2 < 3\}$. □

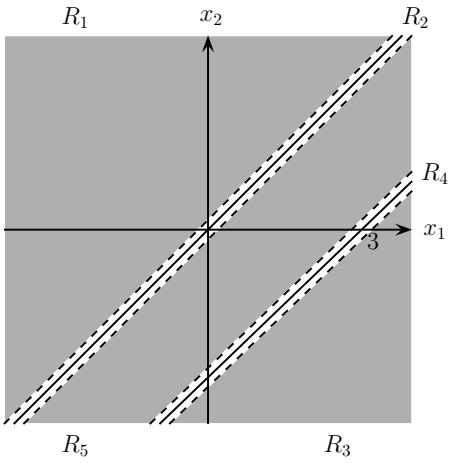

Fig. 2. Partition of $\mathbb{R}^2$ for the MPL model in (2)

We are going to quantify the worst-case time complexity of Algorithm 1 and 2. Since the maximum number of iterations in step 3 of Algorithm 1 is n^n, the time complexity is factorial. Similarly, the time complexity of Algorithm 2 is exponential, since the maximum number of iterations in step 2 is $(2^n-1)^n$.

However, this worst-case is rarely incurred in practice. We implement the partitioning procedure directly on the MPL model, since the complexity of set intersection is lower than the complexity of set difference. In order to improve the performance of the approach we apply the standard pruning tricks, which practically results in efficient outcomes, as shown with the benchmark in Section III-D.

B. LTS Transitions: Backward Reachability Analysis

In this section, we investigate a technique to determine the transition relations between two LTS states, that is between two abstract regions of the MPL state-space partition. At any given point in time k, there is a transition from region R to R' iff there exists a $x(k) \in R$ such that $x(k+1) \in R'$. Such a transition can be determined either with a forward-reachability approach, or with a backward one. According to the former, we calculate $R' \cap \{x(k+1) : x(k) \in R\}$, whereas if we use backward approach we compute $R \cap \{x(k) : x(k+1) \in R'\}$. The non-emptiness of the resulting set characterizes a transition from R to R'.

With focus on the backward-reachability approach, given two regions R and R', let r' be the number of inequalities in R', we calculate the inverse image of the j-th inequality at each iteration, $1 \leq j \leq r'$, by substituting the dynamical system in partition R into the inequality. Finally, we compute the intersection. Notice that the inverse image is compatible with set intersection, that is

$$\left\{x(k) \in \mathbb{R}^n : A_R x(k) + B_R \in \bigcap_{j=1}^{r'} R'_j\right\} = \bigcap_{j=1}^{r'} \left\{x(k) \in \mathbb{R}^n : A_R x(k) + B_R \in R'_j\right\},$$

where R'_j is the region defined by the j-th inequality, for $1 \leq j \leq r'$. Algorithm 3 details the approach (notice that transitions, for the moment, are defined with no labels):

Algorithm 3: Computation of transitions via backward reachability analysis

input : $\mathsf{R} = \{R_1, \ldots, R_r\}$, partition of the state space; $\mathsf{A} = \{A_1, \ldots, A_r\}$ and $\mathsf{B} = \{B_1, \ldots, B_r\}$, the corresponding PWA dynamics
output: $\delta \subseteq \mathsf{R} \times \mathsf{R}$, a transition relation

1 $\delta \leftarrow \emptyset$;
2 **foreach** $1 \leq i, j \leq r$ **do**
3 **if** $\{x \in R_i : A_i\, x + B_i \in R_j\} \neq \emptyset$ **then** $\delta \leftarrow \delta \cup \{(R_i, R_j)\}$;
4 **end**

Let us look into step 3 of Algorithm 3. We are interested in determining the existence of a transition between two regions R and R'. Assume that the dynamics in region R is described by the PWA model (3), where $h_i = A(i, g_i)$, for $1 \leq i \leq n$. Region R' is characterized as in (4). The following steps output a boolean value, equal to true if there is a relation from R to R', else false:

1) $R'_{pre} \leftarrow \mathbb{R}^n$.
2) For each inequality in R' do
   - If $g_i < g_j$ then $R'_{pre} \leftarrow R'_{pre} \cap \{x(k) \in \mathbb{R}^n : \alpha - h_i + h_j \simeq x_{g_i}(k) - x_{g_j}(k) \simeq \beta - h_i + h_j\}$.
   - If $g_i > g_j$ then $R'_{pre} \leftarrow R'_{pre} \cap \{x(k) \in \mathbb{R}^n : \beta + h_i - h_j \simeq x_{g_j}(k) - x_{g_i}(k) \simeq -\alpha + h_i - h_j\}$.
   - If $g_i = g_j$ and ($h_i - h_j \not\simeq \beta$ or $\alpha \not\simeq h_i - h_j$) then $R'_{pre} \leftarrow \emptyset$.
3) If $R \cap R'_{pre} \neq \emptyset$ then return true, else return false.

Recall that each region of the partition is represented by a system of linear inequalities, as in (4). It turns out that such a set can be expressed via a Difference-Bounded Matrix (DBM) [13, Section 4.1], which is a computationally efficient representation. Checking emptiness via a DBM representation can be easily done in polynomial time [13, Section 4.1].

The worst-case time complexity of Algorithm 3 is $\mathcal{O}(r^2 n^3)$, where r is the cardinality of the state-space partition, and for each iteration the complexity of checking emptiness is $\mathcal{O}(n^3)$.

Notice that the obtained transition system can be nondeterministic. Its relationship with the original MPL model is described next.

Proposition 4: The transition system obtained by the Algorithm 3 simulates the original MPL model. □

In general the opposite direction in the preceding statement does not hold true. In fact, whenever the transition system is nondeterministic we can find a region R such that the region has more than one outgoing transition (say that this happens at point $x \in R$). However for each $x \in \mathbb{R}^n$ the value of $A \otimes x$ is unique (A denoting again the MPL system matrix).

Proposition 5: If the transition system obtained by Alg. 3 is deterministic, it bisimulates the original MPL model. □

We can try to obtain a deterministic transition system by successive refinement: within a refinement step, each nondeterministic state is split and its (incoming and outgoing) transitions are updated. Whenever a deterministic transition system is obtained, then we can establish the preceding property. Unfortunately, such a procedure in general does not necessarily terminate, except for special instances discussed shortly.

Proposition 6: For an irreducible MPL model with associated transition system, if the transition system is deterministic over the periodic regime, it is globally deterministic. □

In the following, a matrix whose critical graph is composed of j strongly connected subgraphs and whose cyclicity is k will be denoted by scs$j$-cyc$k$.

Proposition 7: If an irreducible MPL model is scs1-cyc1, the related transition system is deterministic over the periodic regime. □

Proposition 8: For each 2-dimensional irreducible MPL system, the transition system is deterministic over the periodic regime. □

Example: We use backward reachability over the partition regions obtained in the previous example to obtain a nondeterministic transition system. After refinement we obtain a total of 9 partitioning regions, defined as R'_i, for $i = 1, \ldots, 9$. For $1 \leq i \leq 4$, $R'_i = R_i$ and the remaining partitioning regions are $R'_5 = \{x \in \mathbb{R}^2 : 2 < x_1 - x_2 < 3\}$, $R'_6 = \{x \in \mathbb{R}^2 : x_1 - x_2 = 1\}$, $R'_7 = \{x \in \mathbb{R}^2 : x_1 - x_2 = 2\}$, $R'_8 = \{x \in \mathbb{R}^2 : e < x_1 - x_2 < 1\}$ and $R'_9 = \{x \in \mathbb{R}^2 : 1 < x_1 - x_2 < 2\}$.

We are going to explicitly check the existence of a transition from R'_3 to R'_1: after substituting the dynamical system of region R'_3 into the inequalities characterizing R'_1, we obtain the relation $-1 < e$. Thus there is a unique outgoing transition from R'_3 and the destination is R'_1. The overall transition system for this example is shown on Figure 3, where transitions, for the moment, are defined with no labels. □

C. LTS Labels

We introduce labels on the transition system, thus obtaining an LTS. Labels are introduced in two different possible ways: they either characterize 1) the difference between the timing of an event for any two variables of the original model, or represent 2) the time difference between consecutive events of the MPL model:

1) labels are defined as all possible values of $x_i(k) - x_j(k)$, where $1 \leq i < j \leq n$. Given a partitioning region, we can easily compute the labels using its explicit representation. More precisely, the label of each state is defined as the system of linear inequalities characterizing the region. Thus a label is a vector of real-valued intervals. If the dimension of the state space is n, the maximum number of elements in the vector is $n(n-1)/2$.
2) labels are defined as all the possible values of $x_i(k+1) - x_i(k)$, for $1 \leq i \leq n$. A label is again a vector of real-valued intervals: if the dimension of the state space is n, the maximum number of elements in the vector is n. In the nondeterministic case, evaluating the transitions labels requires proper subdivision of the partition regions.

Notice that labels can be of two types, (deterministic) vectors or (nondeterministic) intervals. If the partitioning region is a single equivalence class, we have a deterministic label, because $(A \otimes x') - x' = (A \otimes x'') - x''$, for each $x', x'' \in \bar{x}$, where $A \in \mathbb{R}_{\epsilon}^{n \times n}$ is a regular max-plus matrix. Otherwise, we obtain a nondeterministic label.

In general, the labeling procedure involves collecting information from expressions as in (4) and (3), respectively. The time complexity hinges on the number of transition relations in the LTS.

Example: Consider the transition system of Figure 3, we are going to determine the label of the transition from R'_3 to R'_1. By substituting the dynamical system of R'_3, i.e. $x_1(k+1) = x_1(k) + 2$, $x_2(k+1) = x_1(k) + 3$, to $x(k+1) - x(k)$ and by applying the definition of R'_3 if necessary, we obtain $x_1(k+1) - x_1(k) = 2$ and $x_2(k+1) - x_2(k) > 6$. The complete LTS of this example is shown on Figure 3. □

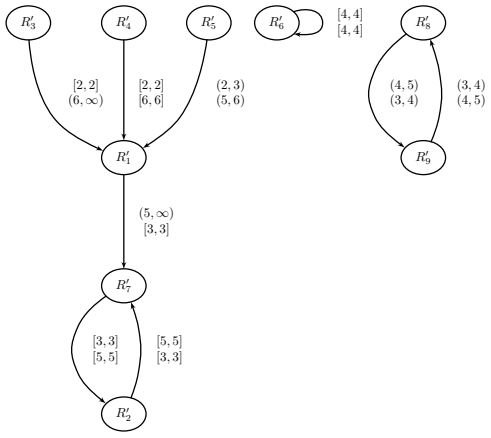

Fig. 3. LTS for the MPL model in (2)

D. Computational Benchmark

In order to test the efficiency of the proposed algorithms we compute the runtime needed to perform abstractions of MPL systems into LTS, for an increasing dimension n of the MPL models. We also keep track of the number of states and of transitions of the obtained LTS. For any given n, we generate sparse matrices (in max-plus sense) with 2 finite elements randomly placed in each row. The finite elements are randomly generated integers between 1 and 100.

Table I reports the time needed to construct the LTS, as well as the number of states and of transitions in the LTS (mean and maximum over 10 experiments). We have run MATLAB code on a dual-core AMD Opteron 2.8 GHz PC with 8 GB of memory.

TABLE I
NUMERICAL BENCHMARK – {MEAN; MAXIMAL} VALUES

size of MPL model	generation of states	generation of transitions	generation of labels	number of states of LTS	number of transitions of LTS
3	{0.36; 0.39} [sec]	{0.09; 0.10} [sec]	{0.02; 0.02} [sec]	{14.20; 15}	{28.00; 37}
4	{0.62; 0.73} [sec]	{0.67; 1.41} [sec]	{0.07; 0.10} [sec]	{45.00; 65}	{163.60; 245}
5	{1.33; 1.78} [sec]	{4.12; 10.04} [sec]	{0.30; 0.70} [sec]	{120.20; 195}	{740.40; 1,613}
6	{3.14; 5.14} [sec]	{39.24; 93.91} [sec]	{1.54; 3.41} [sec]	{304.00; 513}	$\{3.67; 8.47\} \times 10^3$
7	{10.64; 15.85} [sec]	$\sim$ {6.63; 20.31} [min]	{12.02; 23.68} [sec]	{1,029.00; 1,755}	$\{2.57; 4.88\} \times 10^4$
8	{20.31; 25.58} [sec]	$\sim$ {22.44; 62.11} [min]	{38.03; 122.78} [sec]	$\{1.94; 3.11\} \times 10^3$	$\{7.75; 26.53\} \times 10^4$
9	{95.78; 138.18} [sec]	$\sim$ {5.13; 11.50} [hr]	$\sim$ {3.92; 9.37} [min]	$\{7.93; 10.94\} \times 10^3$	$\{4.03; 10.15\} \times 10^5$

IV. VERIFICATION OF MPL MODELS

To specify timed properties for trajectories of the MPL model, we use Linear Temporal Logic (LTL) [5, Chapter 5]. LTL formulas are recursively defined over a set of atomic propositions (AP), by Boolean operators and temporal operators. The set of atomic propositions in our case corresponds to the set of labels of the LTS: $AP = L$. Boolean operators are $\neg$ (negation), $\wedge$ (conjunction), and $\vee$ (disjunction), whereas temporal operators are $\bigcirc$ (next), $\mathcal{U}$ (until), $\Box$ (always), and $\Diamond$ (eventually). A formula ϕ, which in general is (recursively) determined by application of the above operators, is interpreted over traces (trajectories) generated by the LTS. In particular it is of interest to check if (the trajectories of) an LTS satisfies a given formula (or "specification") – this procedure is known as "model checking".

In this work we use the SPIN model checker [6] to verify given specifications on a LTS. Given an MPL model implemented in the MATLAB environment, we first abstract the MPL as an LTS within MATLAB, then export the obtained data structure into the PROMELA language and feed this model, along with an LTL formula that expresses a specification for the model, to SPIN.

Example: We are going to investigate properties of the MPL model expressed in (2). Recall that Section II has already looked at the cyclicity and the periodic regime of this model. First, we construct the LTS from the given MPL model. The LTS is shown on Figure 3 (labels here refer to time delay between consecutive events, cfr. Sec. III-C). As discussed above, the collection of atomic propositions is the set of transition labels in the LTS, thus its cardinality is finite.

Inspecting the LTS in Figure 3, we conclude that the eigenspace and the periodic regime with period 2 correspond to R'_6 and $R'_2 \cup R'_7 \cup R'_8 \cup R'_9$, respectively. In order to reach the eigenspace, the initial condition must be an eigenvector. Notice that, because R'_6 is a one-dimensional region, whenever a point in it is perturbed by a vector $[p_1, p_2]^T : p_1 \neq p_2$, the state will be driven outside the eigenspace.

In order to identify the eigenspace, we can use the formula $\bigvee_{\phi \in AP}(\Box\phi \ \wedge \ |\phi| = 1)$, where $|\cdot|$ denotes the cardinality of a set. In this example, this LTL formula is verified by R'_6. If we want to identify the periodic regime with period c, then we can use $\Psi = \bigvee_{\phi \in AP} \Box(\phi \wedge \bigcirc^c \phi)$, where $\bigcirc^c$ denotes the application of the next operator c times. In this example, $c = 2$ and the LTL formula is verified by R'_2, R'_6, R'_7, R'_8, R'_9.

We can characterize the set $\{x \in \mathbb{R}^n : k_0(x) \leq k\}$, for $k \in \mathbb{N} \cup \{0\}$, by computing the satisfiability set of the LTL formula $\Diamond^{\leq k} \ \Psi$. By extension, we can formulate the value of $k_0(A)$ as a function of LTL formula: $k_0(A) = \arg\min_k \left\{\Diamond^{\leq k} \ \Psi\right\}$. Along with the above properties related to the periodic regime of the MPL model, we may be interested in model checking general formulas, such as the following reach-avoid specification: $\Diamond\psi_1 \vee \Box\neg\psi_2$, where $\psi_1 \in AP$ denotes the incoming label of R'_2, whereas ψ_2 the union of those of R'_3 and R'_4. The satisfiability set results in $R'_1 \cup R'_2 \cup R'_5 \cup R'_7$. □

REFERENCES

[1] F. Baccelli, G. Cohen, and B. Gaujal, "Recursive equations and basic properties of timed Petri nets," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 1, no. 4, pp. 415–439, June 1992.

[2] H. Hillion and J. Proth, "Performance evaluation of job-shop systems using timed event-graphs," *IEEE Transactions on Automatic Control*, vol. 34, no. 1, pp. 3–9, January 1989.

[3] S. Gaubert and R. Katz, "Reachability and invariance problems in max-plus algebra," in *Positive Systems*, ser. Lecture Notes in Control and Information Sciences. Springer Berlin / Heidelberg, 2003, vol. 294, pp. 791–793.

[4] R. 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, February 2007.

[5] C. Baier and J.-P. Katoen, *Principles of Model Checking*. The MIT Press, 2008.

[6] G. Holzmann, *The SPIN Model Checker: Primer and Reference Manual*. Addison-Wesley, 2003.

[7] 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.

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

[9] J. Cochet-Terrasson, G. Cohen, S. Gaubert, M. McGettrick, and J. Quadrat, "Numerical computation of spectral elements in max-plus algebra," in *Proceedings of the IFAC Conference on System Structure and Control*, Nantes, France, July 1998, pp. 699–706.

[10] M. Hartmann and C. Arguelles, "Transience bounds for long walks," *Mathematics of Operations Research*, vol. 24, pp. 414–439, February 1999.

[11] P. Butkovic, "Max-algebra: the linear algebra of combinatorics?" *Linear Algebra and its Applications*, vol. 367, pp. 313–335, 2003.

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

[13] 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. Springer Berlin / Heidelberg, 1990, vol. 407, pp. 197–212.