Computational Techniques for Reachability Analysis of Max-Plus-Linear Systems ⋆

Dieky Adzkiya [a], Bart De Schutter [a], Alessandro Abate [a,b]

[a] *Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD, Delft, The Netherlands*

[b] *Department of Computer Science, University of Oxford, Wolfson Building, Parks Road, Oxford, OX1 3QD, United Kingdom*

Abstract

This work discusses a computational approach to reachability analysis of Max-Plus-Linear (MPL) systems, a class of discrete-event systems widely used in synchronization and scheduling applications. Given a set of initial states, we characterize and compute its "reach tube," namely the collection of set of reachable states (regarded step-wise as "reach sets"). By an alternative characterization of the MPL dynamics, we show that the exact computation of the reach sets can be performed quickly and compactly by manipulations of difference-bound matrices, and further derive worst-case bounds on the complexity of these operations. The approach is also extended to backward reachability analysis. The concepts and results are elucidated by a running example, and we further illustrate the performance of the approach by a numerical benchmark: the technique comfortably handles twenty-dimensional MPL systems (i.e., with twenty continuous state variables), and as such it outperforms the state-of-the-art alternative approaches in the literature.

Key words: Max-plus-linear systems; forward and backward reachability analysis; reach tube and reach set; piecewise affine systems; difference-bound matrices.

1 Introduction

Reachability analysis is a fundamental problem in the area of formal methods, systems theory, and performance and dependability analysis. It is concerned with assessing whether a certain state of a system is attainable from given initial states of the system. The problem is particularly interesting and compelling over models with continuous components – either in time or in (state) space. Over the first class of models, reachability has been widely investigated over discrete-space systems, such as timed automata [9,13], Petri nets [36,43], or hybrid automata [34]. On the other hand, much research has been directed to computationally push the envelope for reachability analysis of continuous-space models. Among the many approaches for deterministic dynamical systems, we report here the use of face lifting [21], the computation of flow-pipes via polyhedral approximations [18], later implemented in CheckMate [17], the formulation as solution of Hamilton-Jacobi equations [42] (related to the study of forward and backward reachability [41]), the use of ellipsoidal techniques [38], later implemented in [37], and the use of differential inclusions [10]. Techniques that have displayed scalability features (albeit at the expense of precision due to the use of over-approximations) are the use of low-dimensional polytopes [30] and the computation of reachability using support functions [29].

Max-Plus-Linear (MPL) systems are discrete-event systems [11,35] with continuous variables that express the timing of the underlying sequential events. MPL systems are used to describe the timing synchronization between interleaved processes, and as such are widely employed in the analysis and scheduling of infrastructure networks, such as communication and railway systems [33], production and manufacturing lines [45]. They are related to a subclass of timed Petri nets, namely timed-event graphs [11]. MPL systems are classically analyzed over properties such as transient and periodic regimes [11], or ultimate dynamical behavior [22]. They can be sim-

⋆ This work is supported by the European Commission STREP project MoVeS 257005, by the European Commission Marie Curie grant MANTRAS 249295, by the European Commission IAPP project AMBI 324432, by the European Commission NoE Hycon2 257462, and by the NWO VENI grant 016.103.020. This article represents an integrated and extended version of [5,6].

Email addresses: `d.adzkiya@tudelft.nl` (Dieky Adzkiya), `b.deschutter@tudelft.nl` (Bart De Schutter), `a.abate@tudelft.nl,aabate@cs.ox.ac.uk` (Alessandro Abate).

ulated (though not verified) via the max-plus toolbox Scilab [44].

Reachability analysis of MPL systems from a *single* initial condition has been investigated in [19,25,28] by leveraging the computation of the reachability matrix, which leads to a parallel with reachability for discrete-time linear dynamical systems. It has been shown in [26, Sec. 4.13] that the reachability problem for autonomous MPL systems with a single initial condition is decidable – this result does not hold for a general, uncountable set of initial conditions. Furthermore, the existing literature does not deal with backward reachability analysis.

Under the requirement that the set of initial conditions is expressed as a max-plus polyhedron [27,47], forward reachability analysis can be performed over the max-plus algebra. Similarly for backward reachability analysis of autonomous MPL systems, where in addition the system matrix has to be max-plus invertible. A matrix is max-plus invertible if and only if there is a single finite element (not equal to $-\infty$) in each row and in each column. Despite the requirements, computationally the approach based on max-plus polyhedra can be advantageous since its time complexity is polynomial. To the authors' best knowledge, there are no approaches for solving the backward reachability problem of nonautonomous MPL systems in the max-plus algebra. Let us also mention that reachability analysis has been used to determine a static max-plus linear feedback controller for a nonautonomous MPL system such that the trajectories lie within a given target tube [7, Sec. 4.3]. In each event step, the target tube is defined as a max-plus polyhedron [7, Eqs. (8) and (11)].

In this work, we generalize the results for reachability analysis of MPL systems. We extend the results for forward reachability by considering an arbitrary set of initial conditions. Additionally for backward reachability analysis, we are able to handle nonautonomous MPL systems and state matrices that are not max-plus invertible. The approach is as follows. We first alternatively characterize MPL dynamics by Piece-wise Affine (PWA) systems, and show that they can be fully represented by Difference-Bound Matrices (DBM) [23, Sec. 4.1], which are quite simple to manipulate computationally. We further claim that DBM are closed over PWA dynamics, which leads to being able to map DBM-sets through MPL systems. Then given a set of initial states, we characterize and compute its "reach tube," namely the union of sets of reachable states (aggregated step-wise as "reach sets"). The set of initial conditions is assumed to be a union of finitely many DBM, which contains the class of max-plus polyhedra (cf. Section 2.3) and of max-plus cones as a special case. The approach is also applied to backward reachability analysis. Due to the computational emphasis of this work, we provide a quantification of the worst-case complexity of the algorithms and of the operations that we discuss throughout the work. Interestingly, DBM and max-plus polyhedra have been used for reachability analysis of timed automata [15,40] and implemented in UPPAAL [13] and in `opaal` [20], respectively. Although related scheduling problems can be solved via timed automata [1], this does not imply that we can employ related techniques for reachability analysis of MPL systems since the two modeling frameworks are not comparable.

Computationally, the present contribution leverages a related, recent work in [3], which has explored an approach to analysis of MPL systems that is based on finite-state abstractions. In particular, the technique for reachability computation on MPL systems discussed in this work is implemented in the VeriSiMPL ("very simple") software toolbox, which is freely available at [2]. To the best of our knowledge, there does not exist any computational toolbox for general reachability analysis of MPL systems, nor is it possible to leverage current software for related timed-event graphs or timed Petri nets. As further elaborated later, reachability computation for MPL systems can be alternatively tackled using the Multi-Parametric Toolbox (MPT) [39]. In a numerical case study, we display the scalability of the tool versus model dimension, and benchmark its computation of forward reachability sets against the alternative numerical approach based on the MPT software [39].

This manuscript represents an extension of the results in [5,6] to forward and backward reachability of *non-autonomous* MPL systems. Further, this article provides a more thorough connection and indeed an extension of existing literature: we have proved that a given max-plus polyhedron can be expressed as a union of finitely many DBM (cf. Proposition 7). Moreover, we explicitly show that, under some assumptions, the number of PWA regions generated by $A^{\otimes N}$ is higher than that generated by A (cf. Proposition 12). This is an important result that allows elucidating the computational complexity of the discussed batch vs. one-shot procedures, which are later on implemented in the computational benchmark.

The article is structured as follows. Section 2 introduces models and preliminary notions. The procedure for forward and backward reachability analysis is discussed in Sections 3 and 4, respectively. Section 5 tests the developed approach over a computational benchmark, whereas a running case study is discussed throughout the manuscript. Finally, Section 6 concludes the work.

2 Models and Preliminaries

This section introduces the models under study (MPL systems), as well as the concepts of Piecewise-Affine (PWA) systems and of Difference-Bound Matrices (DBM), which will play a role in reachability computations.

2.1 Max-Plus-Linear Systems

Define $\mathbb{R}_\varepsilon$, ε and e respectively as $\mathbb{R} \cup \{\varepsilon\}$, $-\infty$ and 0. For $\alpha, \beta \in \mathbb{R}_\varepsilon$, introduce the two operations $\alpha \oplus \beta =$

$\max\{\alpha,\beta\}$ and $\alpha\otimes\beta=\alpha+\beta$, where the element ε is considered to be absorbing w.r.t. $\otimes$ [11, Definition 3.4]. Given $\beta\in\mathbb{R}$, the max-algebraic power of $\alpha\in\mathbb{R}$ is denoted by $\alpha^{\otimes\beta}$ and corresponds to $\alpha\beta$ in the conventional algebra. The rules for the order of evaluation of the max-algebraic operators correspond to those of conventional algebra: max-algebraic power has the highest priority, and max-algebraic multiplication has a higher priority than max-algebraic addition [11, Sec. 3.1].

The basic max-algebraic operations are extended to matrices as follows. If $A,B\in\mathbb{R}_\varepsilon^{m\times n}; C\in\mathbb{R}_\varepsilon^{m\times p}; D\in\mathbb{R}_\varepsilon^{p\times n}$; and $\alpha\in\mathbb{R}_\varepsilon$, then $[\alpha\otimes A](i,j)=\alpha\otimes A(i,j)$; $[A\oplus B](i,j)=A(i,j)\oplus B(i,j)$; and $[C\otimes D](i,j)=\bigoplus_{k=1}^{p}C(i,k)\otimes D(k,j)$, for $1\le i\le m$, $1\le j\le n$. Notice the analogy between $\oplus$, $\otimes$ and $+$, $\times$ for matrix and vector operations in the conventional algebra. Given $m\in\mathbb{N}$, the m-th max-algebraic power of $A\in\mathbb{R}_\varepsilon^{n\times n}$ is denoted by $A^{\otimes m}$ and corresponds to $A\otimes\cdots\otimes A$ (m times). Notice that $A^{\otimes 0}$ is an n-dimensional max-plus identity matrix, i.e. the diagonal and nondiagonal elements are e and ε, respectively. Given two sets $V,W\subseteq\mathbb{R}_\varepsilon^n$, the max-plus Minkowski sum $V\oplus W$ is defined as the set $\{v\oplus w : v\in V, w\in W\}$. In this paper, the following notation is adopted for reasons of convenience. A vector with each component that is equal to 0 (resp., $-\infty$) is also denoted by e (resp., ε). Furthermore, for practical reasons, the state space is taken to be $\mathbb{R}^n$, which also implies that the state matrix A has to be row-finite (cf. Definition 1).

An autonomous MPL system [11, Rem. 2.75] is defined as:

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

where $A\in\mathbb{R}_\varepsilon^{n\times n}$, $x(k-1)=[x_1(k-1)\dots x_n(k-1)]^T\in\mathbb{R}^n$ for $k\in\mathbb{N}$. The independent variable k denotes an increasing discrete-event counter, whereas the state variable x defines the (continuous) timing of the discrete events. Autonomous MPL systems are characterized by deterministic dynamics, namely they are unaffected by exogenous inputs in the form of control signals or of environmental non-determinism.

Definition 1 ([33, p. 20]) *A matrix $A\in\mathbb{R}_\varepsilon^{n\times n}$ is called regular (or row-finite) if A contains at least one element different from ε in each row. The matrix A is called irreducible if the nondiagonal elements of $\bigoplus_{k=1}^{n-1}A^{\otimes k}$ are finite (not equal to ε).*

If A is irreducible, there exists a unique max-plus eigenvalue $\lambda\in\mathbb{R}$ [11, Th. 3.23] and a corresponding eigenspace $E(A)=\{x\in\mathbb{R}^n : A\otimes x=\lambda\otimes x\}$ [11, Sec. 3.7.1].

Example *Consider the following autonomous MPL system from [33, Sec. 0.1], representing the scheduling of train departures from two connected stations $i=1,2$ ($x_i(k)$ is the time of the k-th departure at station i):*

$$x(k)=\begin{bmatrix}2&5\\3&3\end{bmatrix}\otimes x(k-1),\quad\text{or equivalently,}$$
$$\begin{bmatrix}x_1(k)\\x_2(k)\end{bmatrix}=\begin{bmatrix}\max\{2+x_1(k-1),5+x_2(k-1)\}\\\max\{3+x_1(k-1),3+x_2(k-1)\}\end{bmatrix}. \tag{2}$$

Matrix A is a row-finite matrix and irreducible since $A(1,2)\neq\varepsilon\neq A(2,1)$.

Proposition 2 ([33, Th. 3.9]) *Let $A\in\mathbb{R}_\varepsilon^{n\times n}$ be an irreducible matrix with max-plus eigenvalue $\lambda\in\mathbb{R}$. There exist $k_0,c\in\mathbb{N}$ such that $A^{\otimes(k+c)}=\lambda^{\otimes c}\otimes A^{\otimes k}$, for all $k\ge k_0$. The smallest k_0 and c verifying the property are defined as the length of the transient part and cyclicity, respectively.*

Proposition 2 allows to establish the existence of a periodic behavior. Given an initial condition $x(0)\in\mathbb{R}^n$, there exists a finite $k_0(x(0))$, such that $x(k+c)=\lambda^{\otimes c}\otimes x(k)$, for all $k\ge k_0(x(0))$. Notice that we can seek a specific length of the transient part $k_0(x(0))$, in general less conservative than the global $k_0=k_0(A)$, as in Proposition 2. Upper bounds for the length of the transient part k_0 and for its computation have been discussed in [31, Th. 10,Th. 13] and more recently in [16].

Example *In the numerical example* (2)*, from Proposition 2 we obtain a max-plus eigenvalue $\lambda=4$, cyclicity $c=2$, and a (global) length of the transient part $k_0=2$. The specific length of the transient part for $x(0)=[0,0]^T$ can be computed observing the trajectory*

$$\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}5\\3\end{bmatrix},\begin{bmatrix}8\\8\end{bmatrix},\begin{bmatrix}13\\11\end{bmatrix},\begin{bmatrix}16\\16\end{bmatrix},\begin{bmatrix}21\\19\end{bmatrix},\begin{bmatrix}24\\24\end{bmatrix},\begin{bmatrix}29\\27\end{bmatrix},\dots$$

Notice that the periodic behavior occurs immediately, i.e. $k_0([0,0]^T)=0$, and shows a period equal to 2, namely $x(2)=4^{\otimes 2}\otimes x(0)=8+x(0)$. Furthermore notice that $x(k+2)=8\otimes x(k)$, for $k\in\mathbb{N}\cup\{0\}$.

For the backward reachability analysis we introduce the quantity $k_\emptyset(x)$, for any given $x\in\mathbb{R}^n\setminus E(A^{\otimes c})$, as the smallest k such that the system of max-plus linear equations $A^{\otimes k}\otimes x'=x$ does not have a solution. (Practically, there is no point $x'\in\mathbb{R}^n$ that can reach x in $k_\emptyset$ steps or more.) The solution can be computed by using the method in [11, Sec. 3.2.3.2]. Otherwise if $x\in E(A^{\otimes c})$, $k_\emptyset$ is set to 0. It is easy to see that the quantity can be bounded as $k_\emptyset(x)\le k_0(A)-k_0(x)+1$, for each $x\in\mathbb{R}^n$. This (arguably counter-intuitive) definition will be useful for the ensuing work.

A nonautonomous MPL system [11, Corollary 2.82] is defined by embedding an external input u in the dynamics of (1) as:

$$x(k)=A\otimes x(k-1)\oplus B\otimes u(k), \tag{3}$$

where $A\in\mathbb{R}_\varepsilon^{n\times n}$, $B\in\mathbb{R}_\varepsilon^{n\times m}$, $x(k-1)\in\mathbb{R}^n$, $u(k)\in\mathbb{R}^m$, for $k\in\mathbb{N}$. In this work we regard the external input as an exogenous schedule (a control signal) affecting the dynamics. As suggested in [11, Sec. 2.5.4], the nonautonomous MPL system (3) can be transformed into an augmented MPL system with the following dynamics:

$$x(k) = \bar{A} \otimes \bar{x}(k-1), \tag{4}$$

where $\bar{A} = [A, B]$, $\bar{x}(k-1) = [x(k-1)^T, u(k)^T]^T$.

2.2 Piecewise-Affine Systems

This section discusses Piecewise-Affine (PWA) systems [46] generated by an autonomous and by a nonautonomous MPL system. In the ensuing work, PWA systems will play an important role in the forward and backward reachability analysis. PWA systems are characterized by a cover of the state space and by affine (linear, plus a constant) dynamics within each set of the cover.

Every MPL system characterized by a generic row-finite matrix $A \in \mathbb{R}_\varepsilon^{n\times p}$ can be expressed as a PWA system in the event domain [32, Sec. 3]. The affine dynamics, along with the corresponding region on the state space, can be constructed from the coefficients $g = (g_1, \dots, g_n) \in \{1, \dots, p\}^n$. For each i, the coefficient g_i characterizes the maximum term in the i-th state equation $x_i(k) = \max\{A(i,1)+x_1, \dots, A(i,p)+x_p\}$, that is $A(i,j)+x_j \leq A(i,g_i) + x_{g_i}$, $\forall j \in \{1,\dots,p\}$. It follows that the set of states corresponding to g, denoted by R_g, is

$$R_g = \bigcap_{\substack{1\leq i\leq n\\ 1\leq j\leq p}} \{x \in \mathbb{R}^n : A(i,j) + x_j \leq A(i,g_i) + x_{g_i}\}. \tag{5}$$

The affine dynamics that are active in R_g follow directly from the definition of g (see previous paragraph) as

$$x_i(k) = x_{g_i}(k-1) + A(i,g_i), \qquad 1 \leq i \leq n. \tag{6}$$

Given a row-finite state matrix A, Algorithm 1 describes a general procedure to construct a PWA system corresponding to an autonomous MPL system. Similarly, if we run the algorithm with the augmented matrix $\bar{A}$, we obtain a PWA system related to the nonautonomous MPL system. Correspondingly, the parameter p above equals n or $n+m$. On the side, notice that the affine dynamics associated with a dynamical system generated by Algorithm 1 are a special case of the general PWA dynamics as defined in [46, Sec. 1]. The complexity of Algorithm 1 will be formally assessed in the next section.

Algorithm 1 *Generating a PWA system from a generic row-finite max-plus matrix. The assignment $zeros(\cdot,\cdot)$ generates a matrix of a specified dimension, with components equal to e.*

input: $A \in \mathbb{R}_\varepsilon^{n\times p}$, *a row-finite max-plus matrix*
output: $\mathsf{R}, \mathsf{A}, \mathsf{B}$, *a PWA system over* $\mathbb{R}^p$

1: $\mathsf{R} \leftarrow \emptyset$, $\mathsf{A} \leftarrow \emptyset$, $\mathsf{B} \leftarrow \emptyset$ ▷ *notation* $\leftarrow$ *is assignment*
2: ***for all*** $(g_1, \dots, g_n) \in \{1, \dots, p\}^n$ ***do***
3: $\quad R_g \leftarrow \mathbb{R}^p$, $A_g \leftarrow zeros(n,p)$, $B_g \leftarrow zeros(n,1)$
4: $\quad$ ***for all*** $1 \leq i \leq n$ ***do*** ▷ *define regions* (5)
5: $\quad\quad$ ***for all*** $1 \leq j \leq p$ ***do***
6: $\quad\quad\quad R_g \leftarrow R_g \cap \{A(i,g_i) + x_{g_i} \geq A(i,j) + x_j\}$
7: $\quad\quad$ ***end for***
8: $\quad\quad A_g(i,g_i) \leftarrow 1$, $B_g(i) \leftarrow A(i,g_i)$ ▷ *eqn* (6)
9: $\quad$ ***end for***
10: $\quad$ ***if*** R_g *is not empty* ***then***
11: $\quad\quad \mathsf{R} \leftarrow \mathsf{R} \cup \{R_g\}$, $\mathsf{A} \leftarrow \mathsf{A} \cup \{A_g\}$, $\mathsf{B} \leftarrow \mathsf{B} \cup \{B_g\}$
12: $\quad$ ***end if***
13: ***end for***

Algorithm 1 works as follows. First, the output variables are initialized to empty sets (step 1). For each of the coefficients in g (step 2), the region R_g (step 6) and the corresponding affine dynamics (step 8) are computed. Notice that steps 6 and 8 refer to equations (5) and (6), respectively. If the obtained region is not empty (step 10), the procedure saves the region and the associated affine dynamics to the output variables (step 11).

Example *Considering the autonomous MPL example* (2)*, the nonempty regions of the PWA system are:* $R_{(1,1)} = \{x \in \mathbb{R}^2 : x_1 - x_2 \geq 3\}$, $R_{(2,1)} = \{x \in \mathbb{R}^2 : e \leq x_1 - x_2 \leq 3\}$, $R_{(2,2)} = \{x \in \mathbb{R}^2 : x_1 - x_2 \leq e\}$. *The affine dynamics corresponding to a region R_g are characterized by g, e.g. those for region $R_{(2,1)}$ are given by* $x_1(k) = x_2(k-1) + 5$, $x_2(k) = x_1(k-1) + 3$.

2.3 Difference-Bound Matrices

This section introduces the definition of a DBM [23, Sec. 4.1] and of its canonical-form representation. DBM provide a simple and computationally advantageous representation of the MPL dynamics, and will be further used in the next section to represent the reach sets, the backward reach sets and (for nonautonomous models) the set of inputs at each event step.

Definition 3 (Difference-Bound Matrix) *A DBM in $\mathbb{R}^n$ is the intersection of finitely many sets defined by $x_i - x_j \bowtie_{i,j} \alpha_{i,j}$, where $\bowtie_{i,j} \in \{<, \leq\}$, $\alpha_{i,j} \in \mathbb{R} \cup \{+\infty\}$, for $0 \leq i \neq j \leq n$ and the value of x_0 always equal to* 0.

The special variable x_0 is used to represent bounds over a single variable: $x_i \leq \alpha$ can be written as $x_i - x_0 \leq \alpha$. In the following, a "stripe" is defined as a DBM that does not contain x_0. Definition 3 can be likewise given over the input and the corresponding augmented space for nonautonomous MPL systems.

Each DBM admits an equivalent and unique representation in canonical form, which is a DBM with the tightest possible bounds [23, Sec. 4.1]. The Floyd-Warshall algorithm [24] can be used to obtain the canonical-form representation of a DBM. One advantage of the canonical-form representation is that it is easy to compute orthogonal projections w.r.t. a subset of its variables, which is simply performed by deleting rows and columns corresponding to the complementary variables [23, Sec. 4.1].

Another property of a DBM that will be used later is the check of its emptiness: this can be achieved by the

Bellman-Ford algorithm [14, Sec. 5]. The complexity of checking the emptiness of a DBM in $\mathbb{R}^n$ is cubic, i.e. $\mathcal{O}(n^3)$; however when a DBM is in canonical form, this complexity reduces to linear w.r.t. its dimension, i.e. $\mathcal{O}(n)$. Further, the intersection of two DBM is a DBM and can be simply obtained by intersecting their sets (cf. Definition 3) and checking for emptiness.

Let us now draw a connection between DBM and PWA dynamics, and particularly with those in (5)-(6). Each affine dynamics (6) can generate a DBM in $\mathbb{R}^p \times \mathbb{R}^n$, which comprises points $(x(k-1), x(k)) \in \mathbb{R}^p \times \mathbb{R}^n$ such that $x(k)$ is the image of $x(k-1)$. More precisely, the DBM is obtained by rewriting the expression of the affine dynamics as $\bigcap_{i=1}^n \{x_i(k) - x_{g_i}(k-1) \leq A(i, g_i)\} \cap \bigcap_{i=1}^n \{x_i(k) - x_{g_i}(k-1) \geq A(i, g_i)\}$. Furthermore each region (5) is a DBM in $\mathbb{R}^p$.

Looking back at Algorithm 1, its worst-case complexity can be precisely elaborated as follows. Notice that the maximum number of iterations in steps 2, 4 and 5 is p^n, n and p, respectively. Furthermore, the complexity in steps 6 and 10 is constant and amounts to $\mathcal{O}(p^3)$, respectively. Thus, the worst-case complexity of Algorithm 1 is $\mathcal{O}(p^n(np + p^3))$. The bottleneck resides in the worst-case cardinality of the collection of regions in the PWA expression of the MPL dynamics (p^n), which can often be tightened – for example, in the autonomous case if the matrix A has $n' \leq n$ non-ε elements in each row, then the complexity is $(n')^n$. Furthermore, in practice this worst-case is not incurred since many regions can happen to be empty (see step 10 in Algorithm 1). Computationally, we leverage a backtracking technique to improve the performance (cf. Table 1). The following result plays an important role in the computation of reachability for MPL systems.

Proposition 4 ([4, Th. 1]) *The image and the inverse image of a DBM w.r.t. affine dynamics (in particular the PWA expressions in (5)-(6) generated by an MPL system) is a DBM.*

Given a row-finite matrix $A \in \mathbb{R}_\varepsilon^{n \times p}$, the complexity of computing the image and the inverse image of a DBM can be quantified as $\mathcal{O}((n+p)^3)$. The following procedure computes the image of a DBM in $\mathbb{R}^p$, and uses the PWA system generated by $A \in \mathbb{R}_\varepsilon^{n \times p}$: 1) intersecting the DBM with each region of the PWA system; then 2) computing the image of nonempty intersections according to the corresponding affine dynamics (cf. Proposition 4). The worst-case complexity depends on the last step and amounts to $\mathcal{O}(p^n(n+p)^3)$.

Similarly, the inverse image of a DBM in $\mathbb{R}^n$ w.r.t. the MPL system characterized by $A \in \mathbb{R}_\varepsilon^{n \times p}$ can be computed via its PWA representation: 1) computing the inverse image of the DBM w.r.t. each affine dynamics of the PWA system (cf. Proposition 4); then 2) intersecting the inverse image with the corresponding region, which is a DBM; finally 3) collecting the nonempty intersections. The worst-case complexity is quantified again as $\mathcal{O}(p^n(n+p)^3)$. Proposition 4 can be extended as follows.

Corollary 5 *The image and the inverse image of a union of finitely many DBM w.r.t. the PWA system generated by an MPL system is a union of finitely many DBM.*

Computing the image and the inverse image of a union of q DBM can be done by computing the image and the inverse image of each DBM. Thus the complexity of both cases is $\mathcal{O}(qp^n(n+p)^3)$.

Remark 6 *Some of the above results can be generalized to DBM in $\mathbb{R}_\varepsilon^p$ and to matrices that are not row-finite by using similar proof techniques. One of them is the following: the image of a DBM in $\mathbb{R}_\varepsilon^p$ w.r.t. a matrix in $\mathbb{R}_\varepsilon^{n \times p}$ is a union of finitely many DBM in $\mathbb{R}_\varepsilon^n$.*

In Section 1 we have discussed an alternative approach to reachability analysis of MPL systems based on operations over max-plus polyhedra, and emphasized the limitations of such an approach. A max-plus polyhedron is defined as the max-plus Minkowski sum of a finitely-generated max-plus cone and a finitely-generated max-plus convex set [8, Sec. 2.2]. Finitely-generated max-plus cones in $\mathbb{R}_\varepsilon^n$ are a max-plus linear combination of finitely many vectors in $\mathbb{R}_\varepsilon^n$, i.e. $\alpha_1 \otimes w^1 \oplus \cdots \oplus \alpha_p \otimes w^p$. Equivalently, a max-plus cone can be represented as the image of $\mathbb{R}_\varepsilon^p$ w.r.t. a matrix in $\mathbb{R}_\varepsilon^{n \times p}$. A finitely-generated max-plus convex set is defined as the max-plus convex combination of finitely many vectors: its form is similar to that of a max-plus cone, with the additional requirement that $\alpha_1 \oplus \cdots \oplus \alpha_p = e$. Based on Remark 6 and using the homogeneous coordinates representation [8, Sec. 2.2], one can show the following proposition.

Proposition 7 *Every max-plus polyhedron can be expressed as a union of finitely many DBM.*

As a result, our approach is more general than the one based on max-plus polyhedra.

3 Forward Reachability Analysis

The goal of forward reachability analysis is to quantify the set of possible states that can be arrived at under the model dynamics, at a particular event step or over a set of consecutive events, from a set of initial conditions and possibly under the choice of control actions. Two main notions can be introduced.

Definition 8 (Reach Set) *Given an MPL system and a nonempty set of initial conditions $X_0 \subseteq \mathbb{R}^n$, the reach set X_N at the event step $N > 0$ is the set of all states $\{x(N) : x(0) \in X_0\}$ obtained via the MPL dynamics, possibly by application of any of the allowed controls.*

Definition 9 (Reach Tube) *Given an MPL system and a nonempty set of initial positions $X_0 \subseteq \mathbb{R}^n$, the reach tube is defined by the set-valued function $k \mapsto X_k$ for any given $k > 0$ where X_k is defined.*

Unless otherwise stated, in this work we focus on *finite-horizon* reachability: in other words, we compute the reach set for a finite index N (cf. Definition 8) and the reach tube for $k \in \{1, \ldots, N\}$, where $N < \infty$ (cf. Definition 9). While the reach set can be obtained as a by-product of the (sequential) computations used to obtain the reach tube, it can be as well calculated by a tailored procedure (one-shot).

In the computation of the quantities defined above, the set of initial conditions $X_0 \subseteq \mathbb{R}^n$ and the set of inputs at each event step $U_k \subseteq \mathbb{R}^m$ are assumed to be a union of finitely many DBM and a single DBM, respectively. As it will become clear later, this assumption will shape the reach set X_k at any event step $k > 0$ as a union of finitely many DBM. In the more general case of arbitrary sets for X_0 and U_k, these can be over- or under-approximated by DBM. Notice that MPL dynamics are known to be nonexpansive [33, Lemma 3.10]: thus if X_0 is (overapproximated by) a DBM, possible numerical errors associated with forward reachability computations do not accrue [41]. To pin down notations for the complexity calculations below, we assume that X_k is a union of q_k DBM for $k \in \{1, \ldots, N\}$ and in particular that the set of initial conditions X_0 is a union of q_0 DBM.

3.1 Sequential Computation of the Reach Tube

This approach uses the one-step dynamics for autonomous and nonautonomous MPL systems iteratively. In each step, we make use of the DBM representation and the PWA dynamics to compute the successive reach set.

With focus on autonomous MPL systems, given a set of initial conditions X_0, the reach set X_k is recursively defined as the image of X_{k-1} w.r.t. the MPL dynamics:

$$X_k = \mathcal{I}(X_{k-1}) = \{A \otimes x : x \in X_{k-1}\} = A \otimes X_{k-1}.$$

In the dynamical systems and automata literature, the mapping $\mathcal{I}$ is also known as *Post* [12, Definition 2.3]. From Corollary 5, if X_{k-1} is a union of finitely many DBM, then X_k is also a union of finitely many DBM. Then by induction, under the assumption that X_0 is a union of finitely many DBM, it can be concluded that the reach set X_k is a union of finitely many DBM, for each $k \in \mathbb{N}$.

Given a state matrix A and a set of initial conditions X_0, the general procedure for obtaining the reach tube works as follows: first, we construct the PWA system generated by A; then, for each $k \in \{1, \ldots, N\}$, the reach set X_k is obtained by computing $\mathcal{I}(X_{k-1})$. The reach tube is then obtained by aggregating the reach sets.

The worst-case complexity can be assessed as follows. As discussed above, the complexity to characterize the MPL system via PWA dynamics is $\mathcal{O}(n^{n+3})$. Furthermore, the complexity of computing $\mathcal{I}(X_{k-1})$ is $\mathcal{O}(q_{k-1} n^{n+3})$, for $k \in \{1, \ldots, N\}$. This results in an overall complexity of $\mathcal{O}(n^{n+3} \sum_{k=0}^{N-1} q_k)$. Notice that quantifying the cardinality q_k of the DBM union at each step k is not possible in general (cf. benchmark in Section 5).

Let us now look at cases where the structure of the MPL dynamics leads to savings for the computation of the reach tube. Recall that, given an X_0 and a finite $N \in \mathbb{N}$, in order to compute X_N, we need to calculate $X_1, \ldots, X_{N-1}$. Whenever the state matrix of an autonomous MPL system is irreducible, implying the existence of a periodic behavior (cf. Proposition 2), this can be simplified.

Proposition 10 *Let $A \in \mathbb{R}_\varepsilon^{n \times n}$ be an irreducible matrix with max-plus eigenvalue $\lambda \in \mathbb{R}$ and cyclicity $c \in \mathbb{N}$. There exists a $k_0(X_0) = \max_{x \in X_0} k_0(x)$, such that $X_{k+c} = \lambda^{\otimes c} \otimes X_k$, for all $k \geq k_0(X_0)$.*

Thus if the state matrix is irreducible, we only need to compute $X_1, \ldots, X_{k_0(X_0) \wedge N}$ in order to calculate X_N, for any $N \in \mathbb{N}$, where $k_0(X_0) \wedge N = \min\{k_0(X_0), N\}$. Furthermore if X_0 is a union of finitely many stripes, the *infinite-horizon* reach tube is also a union of finitely many stripes and can be computed explicitly in finite time, as elaborated in the following statement.

Theorem 11 *Let $A \in \mathbb{R}_\varepsilon^{n \times n}$ be an irreducible matrix with cyclicity $c \in \mathbb{N}$. If X_0 is a union of finitely many stripes, $\bigcup_{i=0}^{k_0(X_0)+c-1} X_i = \bigcup_{i=0}^{k} X_i$, for all $k \geq k_0(X_0) + c - 1$.*

PROOF. First we will show that X_k is a union of finitely many stripes for all $k \in \mathbb{N}$. Using the steps in the proof of Proposition 4, it can be shown that the image of a stripe w.r.t. affine dynamics (generated by an MPL system) is again a stripe. Following the arguments after Proposition 4, it can be shown that the image of a union of finitely many stripes w.r.t. the PWA system generated by an MPL system is a union of finitely many stripes. Since a stripe is a collection of equivalence classes [33, Sec. 1.4], then $X_0 \otimes \alpha = X_0$, for any $\alpha \in \mathbb{R}$. It follows from Proposition 10 that $X_{k+c} = X_k$, for all $k \geq k_0(X_0)$. $\square$

For *nonautonomous* MPL systems, given a set of initial conditions X_0, the reach set X_k depends on the reach set at event step $k-1$ and on the set of inputs at event step k:

$$X_k = \bar{\mathcal{I}}(X_{k-1} \times U_k) = \{\bar{A} \otimes \bar{x} : \bar{x} \in X_{k-1} \times U_k\}.$$

We can show by induction that the reach set X_k is a union of finitely many DBM, for $k \in \mathbb{N}$. In the base case ($k = 1$), since X_0 is a union of finitely many DBM and U_1 is a DBM, then $X_0 \times U_1$ is a union of finitely many DBM, which implies that its image X_1 is a union of finitely many DBM (cf. Corollary 5). A similar argument holds for the inductive step.

Given a state matrix A, an input matrix B, a set of initial conditions X_0, and a sequence of sets of inputs

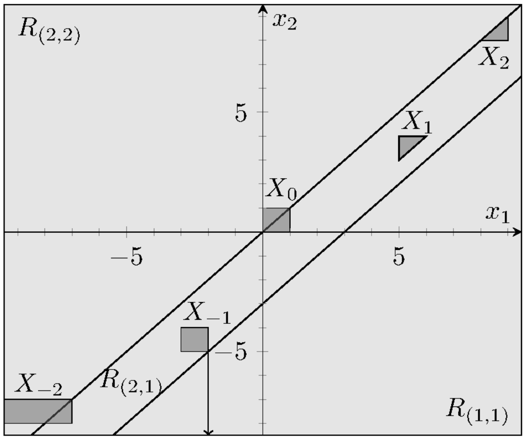

Fig. 1. Forward and backward reach tubes for the autonomous MPL system over 2 event steps. The down-pointing arrow in X_{-1} indicates a half-line: that set can be expressed as a union of two DBM. The rectangle (X_{-2}) at the bottom left is unbounded in the left direction.

$U_1,\ldots,U_N$, the general procedure for obtaining the reach tube works as follows: first, we construct the PWA system generated by $\bar{A}$; then for each $k \in \{1,\ldots,N\}$, the reach set X_k is obtained by computing the image of $X_{k-1} \times U_k$ w.r.t. the PWA system.

Let us quantify the complexity of the procedure. Constructing the PWA system can be done in $\mathcal{O}((n+m)^{n+3})$. For each $k \in \{1,\ldots,N\}$, the complexity of computing X_k critically depends on the image computation and is $\mathcal{O}(q_{k-1}(n+m)^{n+3})$. The overall complexity is $\mathcal{O}((n+m)^{n+3}\sum_{k=0}^{N-1} q_k)$.

Example *Let us consider the unit square as the set of initial conditions $X_0 = \{x \in \mathbb{R}^2 : 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1\}$. The set X_0 is intersected with the regions $R_{(2,2)}$ and $R_{(2,1)}$. The intersections are given by $X_0 \cap R_{(2,2)} = \{x \in \mathbb{R}^2 : x_1 - x_2 \leq 0, x_1 \geq 0, x_2 \leq 1\}$ and $X_0 \cap R_{(2,1)} = \{x \in \mathbb{R}^2 : x_1 - x_2 \geq 0, x_1 \leq 1, x_2 \geq 0\}$. The image of $X_0 \cap R_{(2,2)}$ and of $X_0 \cap R_{(2,1)}$ w.r.t. affine dynamics that are active in $R_{(2,2)}$ and respectively $R_{(2,1)}$ are $\{x \in \mathbb{R}^2 : x_1 - x_2 = 2, x_1 \geq 5, x_2 \leq 4\}$ and respectively $\{x \in \mathbb{R}^2 : x_1 - x_2 \leq 2, x_1 \geq 5, x_2 \leq 4\}$. The reach set at event step 1 is the union of both image regions, i.e. $X_1 = \{x \in \mathbb{R}^2 : x_1 - x_2 \leq 2, x_1 \geq 5, x_2 \leq 4\}$. By applying the same procedure, one can obtain $X_2 = \{x \in \mathbb{R}^2 : x_1 - x_2 \geq 0, x_1 \leq 9, x_2 \geq 8\}$. The reach sets for two event steps are shown in Fig. 1.*

The set of initial conditions can also be described as a stripe, for example $X_0 = \{x \in \mathbb{R}^2 : -1 \leq x_1 - x_2 \leq 1\}$. In this case, the reach sets are stripes given by $X_1 = \{x \in \mathbb{R}^2 : 1 \leq x_1 - x_2 \leq 2\}$ and $X_2 = \{x \in \mathbb{R}^2 : 0 \leq x_1 - x_2 \leq 1\}$.

3.2 One-Shot Computation of the Reach Set

In this section we design a procedure for computing the reach set for a specific event step N using a tailored (one-shot) procedure.

Let us focus on autonomous MPL systems: given a set of initial conditions X_0, we compute the reach set at event step N using

$$X_N = (\mathcal{I} \circ \cdots \circ \mathcal{I})(A) = \mathcal{I}^N(A) = \{A^{\otimes N} \otimes x : x \in X_0\}.$$

Using Corollary 5, it can be seen that the reach set X_N is a union of finitely many DBM. Given a state matrix A, a set of initial conditions X_0 with X_0 being a union of finitely many DBM and a finite index N, the general procedure for obtaining X_N is: 1) computing $A^{\otimes N}$; then 2) constructing the PWA system generated by $A^{\otimes N}$; finally 3) computing the image of X_0 w.r.t. the obtained PWA system.

The worst-case complexity of computing the N-th max-algebraic power of an $n \times n$ matrix (cf. Section 2.1) is $\mathcal{O}(\lceil \log_2(N) \rceil n^3)$. Since X_0 is in general a union of q_0 DBM, the overall complexity of the procedure is $\mathcal{O}(\lceil \log_2(N) \rceil n^3 + q_0 n^{n+3})$. In comparison with the complexity for computing the N-step reach tube, which amounted to an $\mathcal{O}(n^{n+3}\sum_{k=0}^{N-1} q_k)$, the one-shot procedure appears to be advantageous. However, notice that the bottleneck lies on the (exponential) complexity of Algorithm 1, which is applied to two different matrices ($A^{\otimes N}$ and A, respectively). Thus while in general comparing the performance of the sequential and one-shot approaches is difficult, Proposition 12 suggests that under some dynamical assumptions the number of PWA regions generated by $A^{\otimes N}$ is higher than that generated by A.

Proposition 12 *Let R_g and $R_{g'}$ be regions generated by $A \in \mathbb{R}_\varepsilon^{n\times n}$. If $\mathcal{I}(R_g) \subseteq R_{g'}$, then $R_g \subseteq R_{g''}$ for some region $R_{g''}$ generated by $A^{\otimes 2}$.*

PROOF. In this proof, the coefficients g, g', g'' are treated as functions from $\{1,\ldots,n\}$ to $\{1,\ldots,n\}$, e.g. $g : i \mapsto g_i$, for $1 \leq i \leq n$. Recall that the affine dynamics in R_g are

$$x_i(k-1) = x_{g(i)}(k-2) + A(i, g(i));$$

and the ones in $R_{g'}$ are

$$x_i(k) = x_{g'(i)}(k-1) + A(i, g'(i)).$$

Hence, the affine dynamics in $R_{g''}$ can be formulated as a composition of the affine dynamics in $R_{g'}$ and R_g as

$$\begin{aligned} x_i(k) &= x_{g(g'(i))}(k-2) + A(i, g'(i)) + A(g'(i), g(g'(i))), \\ &= x_{g''(i)}(k-2) + A^{\otimes 2}(i, g''(i)). \end{aligned}$$

Notice that $g'' = g \circ g'$, where $\circ$ denotes the function composition operator. $\square$

Of course obtaining a higher number of PWA regions relates to obtaining a reach set expressed with a higher number of DBM. The result above can be generalized to

$A^{\otimes N}$ as follows. Let $R_{g^{(0)}}, \ldots, R_{g^{(N-1)}}$ be regions generated by A. If $\mathcal{I}^i(R_{g^{(0)}}) \subseteq R_{g^{(i)}}$, for each $i \in \{1, \ldots, N-1\}$, then by induction it can be shown that there exists a region $R_{g^{(N)}}$ generated by $A^{\otimes N}$, such that $R_{g^{(0)}} \subseteq R_{g^{(N)}}$, where $g^{(N)} = g^{(0)} \circ \cdots \circ g^{(N-1)}$.

On the side, let us remark that if the MPL dynamics are characterized by an irreducible matrix A, then the above figures should substitute the quantity N with $N \wedge k_0(A)$.

A similar technique can be applied to *nonautonomous* MPL systems. Given a set of initial conditions X_0, the reach set at event step N is computed by using the following formula:

$$X_N = [A^{\otimes N}, A^{\otimes(N-1)} \otimes B, \ldots, B] \otimes (X_0 \times \cdots \times U_N).$$

From Corollary 5, the reach set X_N is again a union of finitely many DBM, since $X_0 \times U_1 \times \cdots \times U_N$ is a union of finitely many DBM. Notice that X_0 is a union of q_0 DBM and $U_1, \ldots, U_N$ are DBM.

Given a state matrix A, an input matrix B, a set of initial conditions X_0, a sequence of sets of inputs $U_1, \ldots, U_N$, the general procedure for obtaining X_N is: 1) generating $[A^{\otimes N}, A^{\otimes(N-1)} \otimes B, \ldots, B]$; then 2) constructing the PWA system generated by it; finally 3) computing the image of $X_0 \times U_1 \times \cdots \times U_N$ w.r.t. the PWA system.

Let us determine the complexity of the approach. In order to generate the matrix, first we compute $A^{\otimes i}$, for $i \in \{2, \ldots, N\}$; then $A^{\otimes i} \otimes B$, for $i \in \{1, \ldots, N-1\}$, which leads to a worst-case complexity $\mathcal{O}(Nn^3 + Nn^2 m)$. Since the size of the obtained matrix is $n \times (n+mN)$, the complexity of the second and third steps is $\mathcal{O}((n+mN)^{n+3})$ and $\mathcal{O}(q_0(n+mN)^{n+3})$, respectively. Unfortunately, this approach is not tractable for problems over long event horizons, since the maximum number of regions of the PWA system is $(n+mN)^n$ and grows exponentially w.r.t. the event horizon N. In this instance, using the sequential procedure (cf. Section 3.1) can be advantageous.

4 Backward Reachability Analysis

The objective of backward reachability analysis is to determine the set of states that enter a given set of final conditions, possibly under the choice of control inputs. This setup is of practical importance, for instance in seeking the set of initial conditions leading to a set of undesired states for any choice of the inputs, as well as in the transient analysis of irreducible MPL systems. Similar to the forward instance, two main notions are first introduced.

Definition 13 (Backward Reach Set) *Given an MPL system and a nonempty set of final positions $X_0 \subseteq \mathbb{R}^n$, the backward reach set X_{-N} is the set of all states $x(-N)$ that lead to X_0 in N steps of the MPL dynamics, possibly by application of any of the allowed controls.*

Definition 14 (Backward Reach Tube) *Given an MPL system and a nonempty set of final positions $X_0 \subseteq \mathbb{R}^n$, the backward reach tube is defined by the set-valued function $k \mapsto X_{-k}$ for any given $k > 0$ where X_{-k} is defined.*

Similar to the forward reachability instance, the set of final conditions $X_0 \subseteq \mathbb{R}^n$ and the set of control actions at each event step $U_{-k} \subseteq \mathbb{R}^m$ are assumed to be a union of finitely many DBM and a single DBM, respectively. In particular, we denote by q_{-k} the cardinality of the set of DBM representing X_{-k} for $k \in \{1, \ldots, N\}$ and assume that the set of final conditions X_0 is a union of q_0 DBM.

4.1 Sequential Computation of Backward Reach Tube

Let us focus on autonomous MPL systems: given a set of final conditions X_0, for each $k \in \{1, \ldots, N\}$ we determine the states that enter X_0 in k event steps by the following recursion:

$$X_{-k} = \mathcal{I}^{-1}(X_{-k+1}) = \{x \in \mathbb{R}^n : A \otimes x \in X_{-k+1}\}.$$

The mapping $\mathcal{I}^{-1}$ is also known in the literature as *Pre* [12, Definition 2.3]. Whenever X_0 is a union of finitely many DBM, by Corollary 5 it follows that the backward reach set X_{-k} is a union of finitely many DBM, for each $k > 0$. As in the forward reachability case, the procedure for obtaining the backward reach tube leverages the dynamics of the PWA system associated with matrix A and the recursion above.

The complexity of computing $\mathcal{I}^{-1}(X_{-k+1})$ at any $k \in \{1, \ldots, N\}$ is $\mathcal{O}(q_{-k+1} n^{n+3})$. This results in an overall worst-case complexity of $\mathcal{O}(n^{n+3} \sum_{k=1}^{N} q_{-k+1})$, where in general it is not feasible to precisely quantify the cardinality q_{-k+1} of the DBM union set at step k.

In general, given an X_0, in order to calculate X_{-N}, where N is finite, we have to determine $X_{-1}, \ldots, X_{-N+1}$, except if the autonomous MPL system is irreducible. The following result is directly shown by the definition of $k_\emptyset$.

Proposition 15 *Let $A \in \mathbb{R}_\varepsilon^{n \times n}$ be an irreducible matrix with cyclicity $c \in \mathbb{N}$. If $X_0 \cap E(A^{\otimes c})$ is empty, there exists a $k_\emptyset(X_0) = \max_{x \in X_0} k_\emptyset(x)$, such that X_{-k} is empty for all $k \geq k_\emptyset(X_0)$.*

Notice that if $X_0 \cap E(A^{\otimes c})$ is empty, from Proposition 15, X_{-k} is empty for $k \geq k_\emptyset(X_0)$. On the other hand if $X_0 \cap E(A^{\otimes c})$ is not empty, the backward reach set at or after $k_\emptyset(X_0)$ steps depends only on $X_0 \cap E(A^{\otimes c})$, i.e. it does not depend on $X_0 \setminus (X_0 \cap E(A^{\otimes c}))$. More precisely in the case of $X_0 \cap E(A^{\otimes c})$ is not empty and $k \geq k_\emptyset(X_0)$, we have $\mathcal{I}^k(X_{-k}) \subseteq X_0 \cap E(A^{\otimes c})$, thus $k_0(X_{-k}) \leq k$. Recall that $k_0(X_{-k}) = \max_{x \in X_{-k}} k_0(x)$.

Theorem 16 *Let $A \in \mathbb{R}_\varepsilon^{n \times n}$ be an irreducible matrix with max-plus eigenvalue $\lambda \in \mathbb{R}$ and cyclicity $c \in \mathbb{N}$, then $\lambda^{\otimes(-c)} \otimes X_{-k} \subseteq X_{-k-c}$, for all $k \geq k_\emptyset(X_0)$.*

PROOF. If $X_0 \cap E(A^{\otimes c})$ is empty, the proposition is trivially satisfied (cf. Proposition 15). Next, we assume that $X_0 \cap E(A^{\otimes c})$ is not empty and that $k \geq k_\emptyset(X_0)$.

We will prove that each element of $\lambda^{\otimes(-c)} \otimes X_{-k}$ enters the set of final conditions in $k + c$ event steps, i.e. $A^{\otimes(k+c)} \otimes \lambda^{\otimes(-c)} \otimes X_{-k} \subseteq X_0$. Observe that since $A^{\otimes k_0(X_{-k})} \otimes X_{-k} \subseteq E(A^{\otimes c})$, from Proposition 2 we conclude that $A^{\otimes(k_0(X_{-k})+c)} \otimes X_{-k} = A^{\otimes k_0(X_{-k})} \otimes X_{-k} \otimes \lambda^{\otimes c}$. The preceding observation and the fact that $k_0(X_{-k}) \leq k$ (see the discussion before this theorem) are used in the following steps:

$$
\begin{aligned}
& A^{\otimes(k+c)} \otimes X_{-k} \otimes \lambda^{\otimes(-c)} \\
&= A^{\otimes(k-k_0(X_{-k}))} \otimes (A^{\otimes(k_0(X_{-k})+c)} \otimes X_{-k}) \otimes \lambda^{\otimes(-c)} \\
&= (A^{\otimes(k-k_0(X_{-k}))} \otimes A^{\otimes k_0(X_{-k})}) \otimes X_{-k} \\
&= A^{\otimes k} \otimes X_{-k} \subseteq X_0. \qquad \square
\end{aligned}
$$

Remark 17 *Since the result in Theorem 16 is not as strong as Proposition 10, for backward reachability we do not obtain a result similar to that in Theorem 11.*

As a side note, by modifying the procedure for obtaining the backward reach tube we can add upon results on transient analysis in the literature by producing a partition of $\mathbb{R}^n$ based on the value of the index k_0. First the set of final conditions X'_0 is defined as $E(A^{\otimes c}) = \{x \in \mathbb{R}^n : k_0(x) = 0\}$. The eigenspace $E(A^{\otimes c})$ is a union of finitely many DBM, since $E(A^{\otimes c})$ [11, Sec. 3.7.2] is a max-plus cone and each max-plus cone can be expressed as a union of finitely many DBM (cf. Proposition 7). Then for each $k \in \mathbb{N}$, the backward reach set is obtained by

$$
X'_{-k} = \begin{cases} \mathcal{I}^{-1}(X'_0) \setminus X'_0, \text{ if } k = 1, \\ \mathcal{I}^{-1}(X'_{-k+1}), \text{ if } k > 1. \end{cases}
$$

Further notice that $X'_{-k} = \{x \in \mathbb{R}^n : k_0(x) = k\}$, for each $k \in \mathbb{N} \cup \{0\}$. The procedure is finite in time, since $X'_1 \cap E(A^{\otimes c})$ is empty (cf. Proposition 15). More precisely, X'_{-k} is empty for $k \geq k_\emptyset(X'_1) + 1$.

For *nonautonomous* MPL systems, given a set of final conditions X_0, the backward reach set X_{-k} depends on the backward reach set and on the set of inputs at event step $-k+1$:

$$
X_{-k} = \{x \in \mathbb{R}^n : \exists u \in U_{-k+1} : \bar{A} \otimes [x^T, u^T]^T \in X_{-k+1}\}.
$$

A practical procedure for computing the set X_{-k} is as follows: 1) compute the inverse image of X_{-k+1} w.r.t. the PWA system generated by $\bar{A}$, i.e. $\{\bar{x} \in \mathbb{R}^{n+m} : \bar{A} \otimes \bar{x} \in X_{-k+1}\}$; then 2) intersect the inverse image with $\mathbb{R}^n \times U_{-k+1}$; and finally 3) project the intersection over the state variables. As in the forward reachability case, by Corollary 5 it can be shown that the backward reach set X_{-k} is a union of finitely many DBM, for $k \in \mathbb{N}$.

Example *Let us consider the unit square as the set of final conditions $X_0 = \{x \in \mathbb{R}^2 : 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1\}$. The backward reach sets are the union of finitely many DBM given by $X_{-1} = \{x \in \mathbb{R}^2 : x_1 - x_2 \geq 3, x_1 = -2\} \cup \{x \in \mathbb{R}^2 : -3 \leq x_1 \leq -2, -5 \leq x_2 \leq -4\}$, $X_{-2} = \{x \in \mathbb{R}^2 : x_1 \leq -7, -8 \leq x_2 \leq -7\}$, and are shown in Fig. 1.*

Let us also consider the case of a stripe as the set of final conditions: $X_0 = \{x \in \mathbb{R}^2 : -1 \leq x_1 - x_2 \leq 1\}$. In this case, the backward reach sets are stripes described by $X_{-1} = \{x \in \mathbb{R}^2 : x_1 - x_2 \geq 1\}$ and $X_{-2} = \{x \in \mathbb{R}^2 : x_1 - x_2 \leq 1\}$.

4.2 One-Shot Computation of the Backward Reach Set

With focus on autonomous MPL systems, given a state matrix A, a set of final conditions X_0 and a finite index N, the states that are able to enter X_0 in N event steps are obtained similarly to those for the forward reachability case:

$$
X_{-N} = \{x \in \mathbb{R}^n : A^{\otimes N} \otimes x \in X_0\}.
$$

Further, by Corollary 5 it can be seen that the backward reach set X_{-N} is a union of finitely many DBM. Notice that because the complexity of computing the image and inverse image w.r.t. the MPL dynamics is the same (cf. Section 2.3), since the complexity of the approach critically depends on this operation, the overall complexity associated with the one-shot computation of the backward reach set amounts to that for the forward instance.

For *nonautonomous* MPL systems, given a set of final conditions X_0, the states that are able to enter X_0 in N event steps are computed by using the following formula:

$$
\begin{aligned}
X_{-N} = \{& x(-N) \in \mathbb{R}^n : \exists u(-N+1) \in U_{-N+1}, \ldots, \\
& u(0) \in U_0 \text{ s.t. } x(0) \in X_0\}.
\end{aligned}
$$

Given a state matrix A, an input matrix B, a set of final conditions X_0 that is a union of finitely many DBM, a sequence of sets of inputs $U_0, \ldots, U_{-N+1}$, the general procedure for obtaining X_{-N} is: 1) generating $[A^{\otimes N}, A^{\otimes(N-1)} \otimes B, \ldots, B]$; then 2) constructing the PWA system generated by it; 3) computing the inverse image of X_0 w.r.t. the PWA system; 4) intersecting the inverse image with $\mathbb{R}^n \times U_{-N+1} \times \cdots \times U_0$; and finally 5) projecting the intersection w.r.t. the state variables. The backward reach set X_{-N} is a union of finitely many DBM. The complexity of the approach is the same as the corresponding for the forward case.

5 Numerical Benchmark

5.1 Implementation and Setup of the Benchmark

The technique for forward and backward reachability computations on MPL systems discussed in this work is implemented in the VeriSiMPL ("very simple") software toolbox, which is freely available at [2]. VeriSiMPL is a software tool originally developed to obtain finite

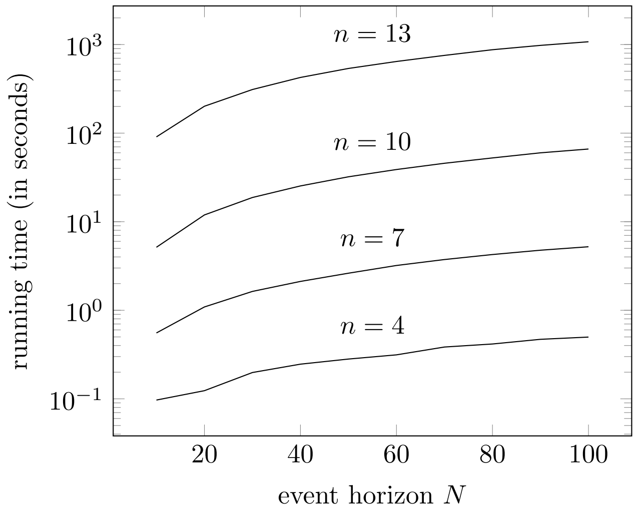

Fig. 2. Time needed to generate reach tube of autonomous models for different models size and event horizons, cf. Section 5.

abstractions of Max-Plus-Linear (MPL) systems, which enables their verification against temporal specifications via a model checker [3]. The algorithms have been implemented in MATLAB 7.13 (R2011b) and the experiments reported here have been run on a 12-core Intel Xeon 3.47 GHz PC with 24 GB of memory.

In order to test the practical efficiency of the proposed algorithms we compute the runtime needed to determine the reach tube of an autonomous MPL system, for event horizon $N = 10$ and an increasing dimension n of the MPL system. We also keep track of the number of regions of the PWA system generated from the MPL system. For any given n, we generate matrices A with 2 finite elements (in a max-plus sense) that are randomly placed in each row. The finite elements are randomly generated integers between 1 and 100. The test over a number of randomly generated dynamics goes against biasing the experimental outcomes and allows claiming the applicability of our technique over general MPL systems. The set of initial conditions is selected as the unit hypercube, i.e. $\{x \in \mathbb{R}^n : 0 \leq x_1 \leq 1, \ldots, 0 \leq x_n \leq 1\}$.

Over 10 independent experiments, Table 1 reports the average time needed to generate the PWA system and to compute the reach tube, as well as the corresponding average number of regions. As confirmed by Table 1, the time needed to compute the reach tube is monotonically increasing w.r.t. the dimension of the MPL system (as we commented previously this is not the case for the cardinality of reach sets, which hinges on the specific dynamics of the MPL systems). For a fixed model size and dynamics, the growth of the computation time for forward reachability is linear with the event horizon as also shown in Fig. 2. We have also performed reachability computations for the case of the set of initial conditions described as a stripe, which has led to results that are quite analogue to those in Table 1. Further, the nonautonomous and backward-reachability cases can be handled similarly.

Table 1
Numerical benchmark, autonomous MPL system: computation of the reach tube (average over 10 experiments)

size of MPL system	generation time for PWA system	number of regions of PWA system	generation time for reach tube	number of regions of X_{10} (q_{10})
3	0.09 [sec]	5.80	0.09 [sec]	4.20
4	0.09 [sec]	12.00	0.13 [sec]	6.10
5	0.14 [sec]	22.90	0.20 [sec]	6.10
6	0.25 [sec]	42.00	0.25 [sec]	3.40
7	0.52 [sec]	89.60	0.72 [sec]	13.40
8	0.91 [sec]	145.00	0.73 [sec]	3.20
9	2.24 [sec]	340.80	2.25 [sec]	4.10
10	4.73 [sec]	700.80	8.23 [sec]	12.30
11	10.42 [sec]	1.44×10^3	15.49 [sec]	3.20
12	20.67 [sec]	2.87×10^3	117.98 [sec]	25.60
13	46.70 [sec]	5.06×10^3	5.27 [min]	16.90
14	82.94 [sec]	9.28×10^3	15.80 [min]	59.90
15	3.48 [min]	2.01×10^4	25.76 [min]	10.10
16	7.90 [min]	4.91×10^4	84.79 [min]	23.50
17	15.45 [min]	9.07×10^4	3.17 [hr]	68.70
18	29.13 [min]	1.58×10^5	5.82 [hr]	21.00
19	67.07 [min]	3.48×10^5	7.13 [hr]	5.00

5.2 Comparison with an Alternative Computation

As discussed in the Introduction, to the best of the authors' knowledge, there exist no generally valid approach for forward reachability computation over MPL systems. This problem can be only alternatively assessed by leveraging the PWA characterization of the model dynamics (cf. Section 2). Forward reachability analysis of PWA systems can be best computed by the Multi-Parametric Toolbox (MPT, version 2.0) [39]. However, the toolbox has some implementation requirements: the state space matrix A has to be invertible – this is in general not the case for MPL systems; the reach sets X_k have to be bounded – in our case the reach sets can be unbounded, particularly when expressed as stripes; further, MPT deals only with full-dimensional polytopes – whereas the reach sets of interest may not necessarily be so; finally, MPT handles convex regions and over-approximates the reach sets X_k when necessary – our approach computes instead the reach sets exactly.

We have been concerned with benchmarking the proposed reachability computations with the described alternative. For the sake of comparison, we have constructed artificial examples (with invertible dynamics) and run both procedures in parallel, with focus on computation time rather than the obtained reach tubes. MPT can handle, in a reasonable time frame, models with dimension up to 10: in this instance (as well as lower-dimensional ones) we have obtained that our approach performs better (cf. Table 2). Notice that this

Table 2
Time for generation of the reach tube of a 10-dimensional autonomous MPL system for different event horizons (average over 10 experiments)

event horizon	20	40	60	80	100
VeriSiMPL	11.02 [s]	17.94 [s]	37.40 [s]	51.21 [s]	64.59 [s]
MPT	47.61 [m]	1.19 [h]	2.32 [h]	3.03 [h]	3.73 [h]

is despite MPT being implemented as object code in the *C* language, whereas VeriSiMPL runs as interpreted code in MATLAB: this leaves quite some margin of improvement to our techniques and software.

6 Conclusions and Future Work

This work has discussed a new computational technique for reachability analysis of Max-Plus-Linear systems, which in essence amounts to exact and fast manipulations of difference-bound matrices through piecewise affine dynamics. The procedure scales over 20-dimensional models thanks to a space partitioning approach that is adapted to the underlying model dynamics, as well as to a compact representation and fast manipulation of the quantities that come into play.

Computationally, we are interested in further optimizing the software for reachability computations, for example by leveraging symbolic techniques based on the use of decision diagrams, and by developing an efficient implementation in the *C* language.

References

[1] Y. Abdeddaïm, E. Asarin, and O. Maler. Scheduling with timed automata. *Theoretical Computer Science*, 354(2):272–300, 2006.

[2] D. Adzkiya and A. Abate. VeriSiMPL: Verification via biSimulations of MPL models. In K. Joshi, M. Siegle, M. Stoelinga, and P.R. D'Argenio, editors, *Proc. 10th Int. Conf. Quantitative Evaluation of Systems (QEST'13)*, volume 8054 of *Lecture Notes in Computer Science*, pages 253–256. Springer, Heidelberg, September 2013.

[3] D. Adzkiya, B. De Schutter, and A. Abate. Abstraction and verification of autonomous max-plus-linear systems. In *Proc. 31st Amer. Control Conf. (ACC'12)*, pages 721–726, Montreal, CA, June 2012.

[4] D. Adzkiya, B. De Schutter, and A. Abate. Finite abstractions of max-plus-linear systems. *IEEE Trans. Autom. Control*, 58(12):3039–3053, December 2013.

[5] D. Adzkiya, B. De Schutter, and A. Abate. Backward reachability of autonomous max-plus-linear systems. In *Proc. 12th Int. Workshop Discrete Event Systems*, pages 117–122, Cachan, May 2014.

[6] D. Adzkiya, B. De Schutter, and A. Abate. Forward reachability computation for autonomous max-plus-linear systems. In E. Ábrahám and K. Havelund, editors, *Tools and Algorithms for the Construction and Analysis of Systems (TACAS'14)*, volume 8413 of *Lecture Notes in Computer Science*, pages 248–262. Springer, Heidelberg, 2014.

[7] M. Ahmane and L. Truffet. Idempotent versions of Haar's lemma: links between comparison of discrete event systems with different state spaces and control. *Kybernetika*, 43(3):369–391, 2007.

[8] X. Allamigeon, S. Gaubert, and É. Goubault. Inferring min and max invariants using max-plus polyhedra. In M. Alpuente and G. Vidal, editors, *Static Analysis*, volume 5079 of *Lecture Notes in Computer Science*, pages 189–204. Springer, Heidelberg, 2008.

[9] R. Alur and D.L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[10] E. Asarin, G. Schneider, and S. Yovine. Algorithmic analysis of polygonal hybrid systems, part i: Reachability. *Theoretical Computer Science*, 379(12):231–265, 2007.

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

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

[13] G. Behrmann, A. David, and K.G. Larsen. A tutorial on UPPAAL. In M. Bernardo and F. Corradini, editors, *Formal Methods for the Design of Real-Time Systems (SFM-RT'04)*, volume 3185 of *Lecture Notes in Computer Science*, pages 200–236. Springer, Heidelberg, September 2004.

[14] R.E. Bellman. On a routing problem. *Quart. Appl. Math.*, 16:87–90, 1958.

[15] J. Bengtsson. *Clocks, DBMs, and States in Timed Systems*. PhD thesis, Uppsala University, 2002.

[16] B. Charron-Bost, M. Függer, and T. Nowak. Transience bounds for distributed algorithms. In V. Braberman and L. Fribourg, editors, *Formal Modeling and Analysis of Timed Systems (FORMATS'13)*, volume 8053 of *Lecture Notes in Computer Science*, pages 77–90. Springer, Heidelberg, 2013.

[17] CheckMate [Online]. Available at `http://users.ece.cmu.edu/~krogh/checkmate/`.

[18] A. Chutinan and B.H. Krogh. Computational techniques for hybrid system verification. *IEEE Trans. Autom. Control*, 48(1):64–75, January 2003.

[19] G. Cohen, S. Gaubert, and J.-P. Quadrat. Max-plus algebra and system theory: Where we are and where to go now. *Annual Reviews in Control*, 23(0):207–219, 1999.

[20] A.E. Dalsgaard, R.R. Hansen, K.Y. Jørgensen, K.G. Larsen, M.Chr. Olesen, P. Olsen, and J. Srba. opaal: A lattice model checker. In M. Bobaru, K. Havelund, G.J. Holzmann, and R. Joshi, editors, *NASA Formal Methods*, volume 6617 of *Lecture Notes in Computer Science*, pages 487–493. Springer, Heidelberg, 2011.

[21] T. Dang and O. Maler. Reachability analysis via face lifting. In T.A. Henzinger and S. Sastry, editors, *Hybrid Systems: Computation and Control (HSCC'98)*, volume 1386 of *Lecture Notes in Computer Science*, pages 96–109. Springer, Heidelberg, 1998.

[22] B. De Schutter. On the ultimate behavior of the sequence of consecutive powers of a matrix in the max-plus algebra. *Linear Algebra and its Applications*, 307(1-3):103–117, March 2000.

[23] D.L. Dill. Timing assumptions and verification of finite-state concurrent systems. In J. Sifakis, editor, *Automatic Verification Methods for Finite State Systems*, volume 407 of *Lecture Notes in Computer Science*, chapter 17, pages 197–212. Springer, Heidelberg, 1990.

[24] R. W. Floyd. Algorithm 97: Shortest path. *Commun. ACM*, 5(6):345, June 1962.

[25] S. Gaubert and R.D. Katz. Reachability and invariance problems in max-plus algebra. In L. Benvenuti, A. De Santis, and L. Farina, editors, *Positive Systems*, volume 294 of

Lecture Notes in Control and Information Science, chapter 4, pages 15–22. Springer, Heidelberg, April 2003.

[26] S. Gaubert and R.D. Katz. Reachability problems for products of matrices in semirings. *International Journal of Algebra and Computation*, 16(3):603–627, 2006.

[27] S. Gaubert and R.D. Katz. The Minkowski theorem for max-plus convex sets. *Linear Algebra and its Applications*, 421(2-3):356–369, 2007.

[28] M.J. Gazarik and E.W. Kamen. Reachability and observability of linear systems over max-plus. *Kybernetika*, 35(1):2–12, 1999.

[29] C. Guernic and A. Girard. Reachability analysis of hybrid systems using support functions. In A. Bouajjani and O. Maler, editors, *Computer Aided Verification (CAV'09)*, volume 5643 of *Lecture Notes in Computer Science*, pages 540–554. Springer, Heidelberg, 2009.

[30] Z. Han and B.H. Krogh. Reachability analysis of large-scale affine systems using low-dimensional polytopes. In J. Hespanha and A. Tiwari, editors, *Hybrid Systems: Computation and Control*, volume 3927 of *Lecture Notes in Computer Science*, pages 287–301. Springer, Heidelberg, 2006.

[31] M. Hartmann and C. Arguelles. Transience bounds for long walks. *Mathematics of Operations Research*, 24(2):414–439, May 1999.

[32] W. Heemels, B. De Schutter, and A. Bemporad. Equivalence of hybrid dynamical models. *Automatica*, 37(7):1085–1091, July 2001.

[33] B. Heidergott, G.J. Olsder, and J.W. 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.

[34] T.A. Henzinger and V. Rusu. Reachability verification for hybrid automata. In T.A. Henzinger and S. Sastry, editors, *Hybrid Systems: Computation and Control (HSCC'98)*, volume 1386 of *Lecture Notes in Computer Science*, pages 190–204. Springer, Heidelberg, 1998.

[35] H.P. Hillion and J.P. Proth. Performance evaluation of job-shop systems using timed event graphs. *IEEE Trans. Autom. Control*, 34(1):3–9, January 1989.

[36] M. Kloetzer, C. Mahulea, C. Belta, and M. Silva. An automated framework for formal verification of timed continuous Petri nets. *IEEE Trans. Ind. Informat.*, 6(3):460–471, 2010.

[37] A.A. Kurzhanskiy and P. Varaiya. Ellipsoidal toolbox. Technical report, EECS Department, University of California, Berkeley, May 2006.

[38] A.A. Kurzhanskiy and P. Varaiya. Ellipsoidal techniques for reachability analysis of discrete-time linear systems. *IEEE Trans. Autom. Control*, 52(1):26–38, January 2007.

[39] M. Kvasnica, P. Grieder, and M. Baotić. Multi-parametric toolbox (MPT), 2004.

[40] Q. Lu, M. Madsen, M. Milata, S. Ravn, U. Fahrenberg, and K.G. Larsen. Reachability analysis for timed automata using max-plus algebra. *The Journal of Logic and Algebraic Programming*, 81(3):298–313, 2012.

[41] I.M. Mitchell. Comparing forward and backward reachability as tools for safety analysis. In A. Bemporad, A. Bicchi, and G. Buttazzo, editors, *Hybrid Systems: Computation and Control (HSCC'07)*, volume 4416 of *Lecture Notes in Computer Science*, pages 428–443. Springer, Heidelberg, 2007.

[42] I.M. Mitchell, A.M. Bayen, and C.J. Tomlin. A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Trans. Autom. Control*, 50(7):947–957, July 2005.

[43] T. Murata. Petri nets: Properties, analysis and applications. *Proc. IEEE*, 77(4):541–580, April 1989.

[44] M. Plus. Max-plus toolbox of Scilab [Online], 1998. Available at `www.cmap.polytechnique.fr/~gaubert/MaxplusToolbox.html`.

[45] B.J.P. Roset, H. Nijmeijer, J.A.W.M. van Eekelen, E. Lefeber, and J.E. Rooda. Event driven manufacturing systems as time domain control systems. In *Proc. 44th IEEE Conf. Decision and Control and European Control Conf. (CDC-ECC'05)*, pages 446–451, December 2005.

[46] E. D. Sontag. Nonlinear regulation: The piecewise-linear approach. *IEEE Trans. Autom. Control*, 26(2):346–358, April 1981.

[47] K. Zimmermann. A general separation theorem in extremal algebras. *Ekonom.-Mat. Obzor*, 13(2):179–201, 1977.