On the Steady State Control of Timed Event Graphs with Firing Date Constraints

Vinicius Mariano Gonçalves, Carlos Andrey Maia, Laurent Hardouin

Abstract—**Two algorithms for solving a specific class of steady state control problems for Timed Event Graphs are presented. In the first, asymptotic convergence to the desired set is guaranteed. The second algorithm, which builds on from the recent developments in the spectral theory of min-max functions, guarantees Lyapunov stability since the distance between the actual state and the desired set never increases. Simulation results show the efficiency of the proposed approach in a problem of moderate complexity.**

Index Terms—**Petri Nets; Timed Event Graphs; Max-plus algebra; Geometrical Invariance; Steady State Control; Lyapunov Stability;**

I. Introduction

A. Presentation and state of art

Timed Event Graphs (**TEG**s hereafter, see Section II for the definition) is an appropriate formalism for modeling some systems, as for example train scheduling [18], manufacturing systems [7], semi-conductor production process [6], car industry [5] and ribosome dynamics [9]. These kinds of systems can have their dynamics described by linear state-space models in Max-plus Algebra (see Section II for the definition), with two isomorphic possible approaches: daters (event domain) and counters (time domain) [8]. In this context, it may be desirable that a certain set of constraints in the state space holds. This could be done by using the state variables to design a control law, in analogy with classical control theory.

Several authors studied this problem when the constraints can be written as a set of max-plus linear equations. In [3], the authors deal with the problem in the time domain, treating a specific subset of linear constraints. These results were further generalized by the authors in [4]. In [7], working in the event domain and assuming that all the states are measurable and controllable, they were further generalized by treating maximum duration constraints. In [20], a max-plus geometric control method was used to find a linear feedback controller that guarantees certain constraints (in the dater domain). In [24] and [23], also working in the dater domain, the authors deal with the problem using eigenvectors and by solving a max-plus affine equation, respectively. Finally, the previous work of the authors, [16], also proposed a linear feedback controller, which is a solution of a max-plus affine equation. The most general form of max-plus linear constraints was considered in [24], [20], [16], but in the first two the set must be computed explicitly as an image of a matrix and, although important recent advances have been made in the recent years, this is still a hard problem. See the complexity of algorithms such as [21], [2], [10], [29], which compute the entire set of solutions of max-plus linear equations. [16], specially, proposes a more tractable approach in which the constraints can be manipulated in their implicit formulation as a max-plus linear equation. It is also worth mentioning that the related problem of observers in the max-plus setting was discussed in [22].

All these works have in common the structure of the proposed controller: a linear static state feedback. Further, they consider different classes of max-plus linear constraints. By the author's knowledge, the most generic case, that is, when each constraints in the state x can be written as a generic two-sided max-plus linear equation $Ex = Dx$, was considered in an implicit form, more computationally convenient, solely in a previous work of the authors [16]. The present paper proposes two algorithms to solve a steady state version of the most general problem considered in [16], in which the constraints are required to hold only from a given step onwards for *any* given initial condition. The authors believe they are the first to consider this kind of problem explicitly. In the first algorithm, the asymptotic stability is ensured, implying the convergence to the desired set. In the second one, which uses optimization techniques to improve the rate of convergence of the previous method, thus far only the Lyapunov stability is ensured, since the error between the state and the desired set is non-decreasing. These optimization techniques borrow from recent developments on the generalized spectral theory of min-max functions (see [15], [1] and the references therein).

Further, a broader class of constraints, in comparison with the previously mentioned works, is considered: *max-plus multiplicatively invariant sets* (see Definition 1). A special case of these constraints, the one considered in [16], is discussed in more detail. The resulting controller is not, in general, a linear state feedback. By the author's knowledge, no published work thus far neither proposes a method for steady-state control of the firing dates of TEGs, for the general problem considered in [16], nor considers this generalized class of constraints. Finally, aiming to illustrate the results developed in this paper, the two proposed methodologies are applied in a problem of moderate complexity.

B. Contributions

Succinctly, the contributions of this paper are:

- An approach to solve the steady-state control problem

Vinicius Mariano Gonçalves and Carlos Andrey Maia are with Programa de Pós Graduação de Engenharia Elétrica, Universidade Federal de Minas Gerais (UFMG) (e-mails: mariano@cpdee.ufmg.br, maia@cpdee.ufmg.br), BRAZIL. Laurent Hardouin is with Laboratoire Angevin d'Ingénierie des Systèmes (LARIS-ISTIA), Université d'Angers (e-mail: laurent.hardouin@univ-angers.fr), FRANCE.

for max-plus linear event-invariant systems: the Periodic Synchronizer (Algorithm 6.1). This approach offers important advantages: it is computationally inexpensive, finite step asymptotic stability is guaranteed and it is possible to derive a bound to this number of steps, a bound which can be shown to be in many cases strict by experimental observations and thus not conservative. The method also has some desirable robustness properties. An improvement to the speed of convergence of this algorithm, the Feedback Accelerator (Algorithm 7.1), is also proposed. This approach is computationally inexpensive and shown to be, in the worst case, non-detrimental to the problem in some aspects (Subsection VII-B). Experimental results, however, show that it can in some cases improve the performance of the system (Section IX);

- Another optimized algorithm, also computationally inexpensive in average, for solving the proposed steady state control problem: the Chebyshev-Optimized Feedback (Algorithm 8.1). In this algorithm, thus far only the Lyapunov stability is guaranteed. Simulations under many different approaches, as well as comparisons with the previously mentioned algorithm and its improvement, however, illustrate the efficiency of the method.

II. Definitions

A *Timed Event Graph* is a subclass of timed Petri nets in which all places have a single upstream and single downstream transitions. The *Max-plus Algebra* is the idempotent semiring (dioid)

$$\mathbb{Q}_{\mathtt{max}} \equiv (\mathbb{Q} \cup \{-\infty\}, \oplus, \otimes)$$

in which $\oplus$ is the maximum and $\otimes$ is the traditional sum. One also defines the *complete dioid* induced by Max-plus algebra, $\overline{\mathbb{Q}}_{\mathtt{max}} \equiv (\mathbb{Q} \cup \{-\infty, \infty\}, \oplus, \otimes)$. One denotes the neutral element of the sum, $-\infty$, as $\bot$ and $+\infty$ as $\top$. One defines also $\top \otimes \bot = \bot \otimes \top = \bot$. As in the traditional algebra, the symbol $\otimes$ is usually omitted. Further, a matrix composed entirely of $\bot$, entirely of $\top$ and entirely of 0, of convenient dimension, is denoted as $\boldsymbol{\bot}$ (bolded $\bot$), $\boldsymbol{\top}$ (bolded $\top$) and $\mathbf{0}$, respectively. If M is a matrix, the entry in the i^{th} row and j^{th} columns is denoted as M_{ij} or $\{M\}_{ij}$, whichever is more convenient. Square brackets are used to denote a sequence of objects, for instance matrices, sets, functions, etc, as in $x[k]$. The dimension of the matrices will only be specified when necessary. A matrix is said to be lower (upper) bounded if no entry is equal to $\bot$ ($\top$).

The *max-plus identity matrix* of appropriate order is denoted by I, more specifically $I_{ij} = 0$ if $i = j$ and $\bot$ otherwise. The *pointwise infimum* is denoted by $\wedge$. $+$ and $-$ will have their usual meaning as traditional sum and subtraction/opposite. Further, $\preceq$ is the *natural order* in the dioid, $M \preceq N$ if and only if $M \oplus N = N$ and $M \succeq N$ if and only if $N \preceq M$. Also, $N \npreceq M$ and $M \nsucceq M$ means that $N \preceq M$ does not hold and $M \succeq N$ does not hold, respectively.

$\backslash\!\!\circ$, $\circ\!\!/$ are used to denote the *left and right residuation of the product*, respectively. For scalars $\alpha, \beta \in \mathbb{Q}_{\mathtt{max}}$ (see [8]), the residuation can be computed as:

$$\alpha \circ\!\!/ \beta = \beta \backslash\!\!\circ \alpha \equiv \begin{cases} \alpha - \beta \text{ if } \beta \neq \bot; \\ \top \text{ if } \beta = \bot \ . \end{cases}$$

And hence, for matrices:

$$M \backslash\!\!\circ N \equiv \max_{MX \preceq N} X \text{ or, equivalently,}$$
$$\{M \backslash\!\!\circ N\}_{ij} = \bigwedge_k M_{ki} \backslash\!\!\circ N_{kj},$$

$$M \circ\!\!/ N \equiv \max_{XN \preceq M} X \text{ or, equivalently,}$$
$$\{M \circ\!\!/ N\}_{ij} = \bigwedge_k M_{ik} \circ\!\!/ N_{jk}.$$

Note that, even if the entries of M and N are in $\mathbb{Q}_{\mathtt{max}}$, $M \backslash\!\!\circ N$ and $M \circ\!\!/ N$ have, in general, entries in $\overline{\mathbb{Q}}_{\mathtt{max}}$.

The *absolute value* of a scalar λ is denoted by $|\lambda|$. $\|x\|_2 \equiv \sqrt{\sum_i x_i^2}$ (the square and square root being in the usual sense) is the *Euclidean norm* of a vector, while $\|x\|_\infty \equiv \max_i |x_i|$ is the p-norm with $p = \infty$. M^T is the *transpose* of M. For a square matrix M, $\rho(M)$ is the *spectral radius* (greatest eigenvalue) of M. If $\lambda \neq \bot$, then $\lambda^{-1} \equiv -\lambda$. The *Kleene closure* of M is defined as $M^* \equiv \bigoplus_{i=0}^{\infty} M^i$. For a natural number n and a square matrix M, M^n, *the* n^{th} *power of* M, is defined recursively as $M^0 \equiv I$ and $M^n \equiv MM^{n-1}$.

A *semimodule*, over a given dioid, is an analogous of vector spaces over semirings, that is, a set of elements x together with a scaling $(\lambda, x) \mapsto \lambda x$ and sum $(x, y) \mapsto x \oplus y$ operations which preserve some properties in the context of this given dioid. See [20] for the formal definition. Finally, $\operatorname{Im} M$, the *image of* M, is the semimodule generated by the max-plus column span of the matrix M, that is, if $M \in \mathbb{Q}_{\mathtt{max}}^{n\times m}$, $\operatorname{Im} M \equiv \{Mv \mid v \in \mathbb{Q}_{\mathtt{max}}^m\}$.

III. The problem

A. Problem statement

Problem 1: (*Steady state control problem*) Consider a TEG whose dater evolution is given by the dynamical system

$$S: \begin{cases} x[k+1] = Ax[k] \oplus Bu[k+1] \text{ for } k \geq 0 \\ x[0] = x_{\mathtt{ic}} \end{cases} \tag{3}$$

with $x[k] \in \mathbb{Q}_{\mathtt{max}}^n$, $A \in \mathbb{Q}_{\mathtt{max}}^{n\times n}$, $B \in \mathbb{Q}_{\mathtt{max}}^{n\times m}$ and $x_{\mathtt{ic}} \in \mathbb{Q}_{\mathtt{max}}^n$. One assumes, therefore, that all the states are measurable and can be used to compute the control action.

The objective is to design a non-decreasing controller $u[k] \in \mathbb{Q}_{\mathtt{max}}^m$ ($u[k+1] \succeq u[k]$) and causal (that is, it does not requires the prediction of events), such that the state $x[k]$ belongs to a particular set $\mathcal{X}_{\mathtt{cons}}$ for all $k \geq k'$, for a given k'. This set is required to be *max-plus multiplicatively invariant*. ■

Definition 1: (*Max-plus multiplicatively invariant sets*) A set $\mathcal{X}$ is said to be *max-plus multiplicatively invariant* if $x \in \mathcal{X} \Rightarrow \lambda x \in \mathcal{X}$ for any scalar $\lambda \in \mathbb{Q}_{\mathtt{max}}$. ■

Note that for a max-plus multiplicatively invariant set there always exist max-plus homogeneous functions $L(x)$, so

$L(\lambda x) = \lambda L(x)$ for any scalar λ, and $R(x)$ such that $\mathcal{X}$ is characterized as the solution set of $L(x) = R(x)$. For an example of such functions, let $\mathbb{QP}^n_{\max}$ be the set of all vectors in $\mathbb{Q}^n_{\max}$ such that $\max_i x_i = 0$ and $\overline{\mathcal{X}} = \mathcal{X} \cap \mathbb{QP}^n_{\max}$. Let $L(x) = \max_i x_i$ (which is max-plus homogeneous) and $R' : \mathbb{QP}^n_{\max} \mapsto \{0, -1\}$ defined as 0 if $x \in \overline{\mathcal{X}}$ and -1 otherwise. In this case, $R(x) = (\max_i x_i) R'((\max_i x_i)^{-1} x)$ for $x \neq \perp$ and $R(\perp) = \perp$ is max-plus homogeneous and the solution set to $L(x) = R(x)$ is exactly $\mathcal{X}$. In practice, however, such functions naturally appear from the problem description, as it will be shown for a particular kind of max-plus multiplicatively invariant sets: a semimodule.

It is also important to define two concepts.

Definition 2: (*Non-degenerate in the control sense*) A problem as in Problem 1 is said to be *non-degenerate in the control sense* if no column of the matrix B is null. ■

A problem being non-degenerate in the control sense is not a restrictive hypothesis. In fact, it can be assumed without loss of generality: if there is a null column in B, say the i^{th} one, then the i^{th} control input plays no role in the system and can be removed.

Definition 3: (*Coupled*) A problem as in Problem 1 is said to be *coupled* if the existence of a solution implies that, unless $x[k] \neq \perp$, $x_i[k] - x_j[k]$ stays bounded for all i, j and k. ■

In this paper, one is only interested in coupled problems. Indeed, it could be argued that otherwise they are meaningless in practice or can be broken in independent subproblems, which then can be solved separately. If the problem is not coupled, in the steady state and under control there will be disjoint sets of transitions, specifically those created from the quotient by the equivalence relation "is coupled with" such that i is coupled with j if and only if $|x_i[k] - x_j[k]|$ is bounded, that operate in different rates. This means that no interesting synchronization was imposed between these disjoint subsets.

For instance, specifications of the form $x[k] \succeq Qx[k]$ are discussed in [20]. That paper argues that, frequently, in practical applications the entries Q_{ij} of this matrix can be chosen to be different than $\perp$ (by replacing it by a very large negative number, for instance). This alone would imply that $x_j[k] - x_i[k] \leq -Q_{ij}$ and $x_i[k] - x_j[k] \leq -Q_{ji}$ and thus $|x_i[k] - x_j[k]| \leq \max(-Q_{ij}, -Q_{ji})$, which is finite under the consideration that $Q_{ij} \neq \perp$ for all i and j. This, in turn, implies that the problem is coupled. Putting bound constraints is a way to ensure that the problem is coupled, but not the only one. Indeed, the system itself may guarantee that all the trajectories have $x_i[k] - x_j[k]$ bounded.

An example of a non-coupled problem is a system composed of two completely independent machines in which, as a constraint, one is required to produce one piece at every 2 minutes and the other at every 1 minute. This problem is not coupled because if $x_1[k]$ and $x_2[k]$ represent the time of completion of the k^{th} pieces for the first and second machines, respectively, then $|x_1[k] - x_2[k]|$ grows roughly with $2^k - 1^k = 1^k = k$, which is unbounded. There is no interesting additional requirement in the firing dates, at least in steady state, that can be imposed between them.

Taking these observations in consideration, the following hypothesis will be posed.

Hypothesis 3.1: (*Non-degenerate in the control sense and coupled problems*) It will be assumed that any problem as in Problem 1 is non-degenerate in the control sense and coupled. ■

At last, it is important to emphasize that in some applications guaranteeing the desired constraints *only* in steady state is prohibitive, since the violation of them can imply inadmissible consequences. For instance, a manufacturing system for which one of the constraints is that a piece cannot stay in the oven more than, say, 3 minutes: violation of this constraint may imply the loss of the piece. In this case, the approaches referenced in Section I may be more appropriate. Nevertheless, the proposed approaches *can also* be used to this purpose provided that it is possible to choose a convenient initial condition $x_{\mathtt{ic}}$.

B. Non-decreasing input

The proposed problem (Problem 1) requires that $u[k+1] \succeq u[k]$, an important requirement to ensure that the designed control input is realizable.

First of all, one assumes that $A \succeq I$. In practice, this assumption can be considered without loss of generality, because it simply implies that the firings are ordered, that is, $x[k] \succeq x[k-1]$. If $A \not\succeq I$, one can simply replace A by $A \oplus I$. This is true because it will not change the non-decreasing trajectories of the system, which are exactly the ones that are relevant in practical applications. In this way, the system naturally "implements" a causalisation.

Proposition 1: (Causalisation) Assume that $A \succeq I$. Consider two trajectories of the system represented by Equation (3), $x_{\mathtt{n}}[k]$ and $x_{\mathtt{c}}[k]$, with the same initial condition, $x_{\mathtt{n}}[0] = x_{\mathtt{c}}[0]$, in which in the former the input $u[k+1]$ is used and in the latter $\bigoplus_{i=0}^{k} u[i+1]$. Then $x_{\mathtt{c}}[k] = x_{\mathtt{n}}[k]$ for all $k \geq 0$.

Proof: The proof follows by induction. Suppose $x_{\mathtt{c}}[k] = x_{\mathtt{n}}[k]$. Since $A \succeq I$, the following equalities hold

$$x_{\mathtt{c}}[k+1] = Ax_{\mathtt{c}}[k] \oplus B\left(\bigoplus_{i=0}^{k} u[i+1]\right) = Ax_{\mathtt{c}}[k] \oplus B\left(\bigoplus_{i=0}^{k} u[i+1]\right) \oplus x_{\mathtt{c}}[k].$$

Further

$$x_{\mathtt{c}}[k+1] = Ax_{\mathtt{c}}[k] \oplus B\left(\bigoplus_{i=0}^{k} u[i+1]\right) \oplus x_{\mathtt{c}}[k] = (Ax_{\mathtt{c}}[k] \oplus Bu[k+1]) \oplus x_{\mathtt{c}}[k] \oplus B\left(\bigoplus_{i=0}^{k-1} u[i+1]\right).$$

Expanding $x_{\mathtt{c}}[k] = Ax_{\mathtt{c}}[k-1] \oplus B\left(\bigoplus_{i=0}^{k-1} u[i+1]\right)$ it is clear that $x_{\mathtt{c}}[k] \oplus B\left(\bigoplus_{i=0}^{k-1} u[i+1]\right) = x_{\mathtt{c}}[k]$. Hence

$$x_{\mathtt{c}}[k+1] = (Ax_{\mathtt{c}}[k] \oplus Bu[k+1]) \oplus x_{\mathtt{c}}[k] = (Ax_{\mathtt{c}}[k] \oplus Bu[k+1]) = (Ax_{\mathtt{n}}[k] \oplus Bu[k+1]) = x_{\mathtt{n}}[k+1]$$

in which $A \succeq I$ and $x_{\mathrm{c}}[k] = x_{\mathrm{n}}[k]$ were used. Since it obvious holds for $k = 0$, because they have the same initial condition, the proof is completed. ■

So, using $u[k+1]$, which can be possibly not non-decreasing, produces the same output as the non-decreasing $\bigoplus_{i=0}^{k} u[i+1]$. This means that it is possible to disregard the non-decreasing requirement when the control sequence $\{u[k+1]\}$ is designed. The causalisation can be made *when implementing*, generating a new non-decreasing sequence $\{\bigoplus_{i=0}^{k} u[i+1]\}$.

So, *such a procedure will be assumed from now on* and the non-decreasing property will not be mentioned anymore in this text.

C. Causality

Another concern is the causality of the control law. For instance, consider the (feedback) control law $u_1[k] = -1 + x_1[k]$, which reads as "fire u_1 for the k^{th} time one time unit *before* x_1 fires for the k^{th} time". This control law requires a prediction of events.

Suppose that the control law is written as $Fx[k]$, a feedback control. Then, in order to this term to be causal F must be a *causal* matrix.

Definition 4: (Causal matrix) A matrix F is said to be *causal* if all its entries are non-negative or $\perp$. ■

In this text, the feedback terms will come as solutions of inequalities of the form $BF \preceq M$ for specific matrices M, and it is advantageous to have BF as close as possible to M. Let $\mathcal{F}$ be the set of all causal matrices. Therefore, the problem asks for

$$\max_{F \in \mathcal{F}} BF \preceq M. \tag{4}$$

Since the set of causal matrices is closed under (max-plus) addition (if $F[1], F[2] \in \mathcal{F}$, so is $F[1] \oplus F[2]$), the problem presented in Equation (4) has a solution. To this, it is necessary to define the concept of *causal projection* [12].

Definition 5: (Causal projection) The causal projection of a matrix F, denoted by $C_{\mathrm{p}}(F)$, is obtained from F by exchanging all negative entries to $\perp$. ■

Taking this into consideration, one proceeds by computing $B \backslash\!\!\!\circ M$, that is, solving by residuation the problem presented in Equation (4) disregarding the causality constraint, and then applying the causal projection to this matrix. Thus

$$F = C_{\mathrm{p}}(B \backslash\!\!\!\circ M). \tag{5}$$

Finally, it is important to mention that the concept of causal feedbacks was also discussed and considered in [4].

IV. Mean-payoff games

A. Basic facts

Before presenting the two methods for solving the proposed problem (Problem 1), it is necessary to discuss the concept of *mean-payoff games*. This concept will be useful in both methods that will be proposed to solve Problem 1.

Consider a directed bipartite graph with two disjoint sets of nodes, say "CIRCLE" nodes (n nodes $i = 1, 2, .., n$) and "SQUARE" nodes (m nodes $j = 1, 2, .., m$). A game is played in which, initially, a pawn is in one SQUARE node j. A player, MIN, plays by moving the pawn to a CIRCLE node i and receives from the other player, MAX, an amount G_{ij}. After, it is time for the player MAX to move the pawn to a SQUARE node j' and then to receive from MIN an amount $H_{ij'}$. Finally, a turn ends and this zero-sum game proceeds again with a move from MIN player, and so on.

Given a number of turns k, one defines $v_j[k]$ as the *value*[1] of the finite horizon game for player MAX in which k turns are played and the starting SQUARE node is j. The mean payoff version of this game is of special interest (called *mean-payoff game*). There the payoff of an infinite trajectory ($k \to \infty$) is defined as the average payment (payments received minus payments made) per turn received by player MAX. In this case, the value of this game at the starting SQUARE node j, the scalar χ_j, is the limsup of the ratio (in the traditional algebra) $v_j[k]/k$ as k goes to infinity (see [1] for a deeper discussion).

There is a close connection between mean-payoff games and max-plus linear equations [1]. First, every max-plus linear equation $Py = Qy$ can be written as $Gy \preceq Hy$, since one can write equivalently $Py \preceq Qy$ and $Qy \preceq Py$, which clearly can be rewritten as $Uy \preceq Vy$. Therefore, when a max-plus linear equation is written in this form, $Gy \preceq Hy$, the above-described game provides useful information about the solutions of this equation. Indeed, the vector χ can be obtained by studying the dynamic programming operator for this game (see [14] for an example of a pseudo-polynomial algorithm for computing the vector χ), the function $f(y) = G \backslash\!\!\!\circ (Hy)$, which is a *min-max* function [19]. Hence, one can define χ as a functional $\chi(f)$ which takes the dynamic programming operator of the mean-payoff game and return a vector with the value in each state, a state being a SQUARE node.

Define the n^{th} composition of f as $f^n(y) \equiv f(f^{n-1}(y))$ with $f^0(y) \equiv y$. Then, for any upper and lower bounded vector y (see [1])

$$\chi(f) = \lim_{N \to \infty} \frac{f^N(y)}{N}. \tag{6}$$

The following definition is important.

Definition 6: (Non-degenerate min-max function) A min-max function f is said to be *non-degenerate* if as long as y is lower and upper bounded, so is $f(y)$. ■

In practical applications, the min-max functions f are non-degenerate. Hence the following theorem, which is fundamental for this paper, is shown in [1]:

Theorem 1: (Support of a solution, see Theorem 3.2 in [1]) Let $f(y) = G \backslash\!\!\!\circ (Hy)$ be a non-degenerate min-max function which is the dynamic programming operator of the mean payoff game associated to the max-plus linear equation $Gy \preceq Hy$. A solution y of $Gy \preceq Hy$ such that $y_i \neq \perp$ exists if and only if $\chi_i(f) \geq 0$ (i is a *winning state*, by definition). Moreover if $\mathcal{R} = \{i \mid \chi_i(f) \geq 0\}$ there exists a vector y solution of $Gy \preceq Hy$ such that $y_i \neq \perp$ for all $i \in \mathcal{R}$. ■

Also, the following comes easily as a corollary from Lemma 2.2 in [17], since any min-max function is a topical function.

[1] In the sense of the MINIMAX theorem, see [25].

Corollary 1: of Lemma 2.2 in [17] *(Bounded value for non-degenerate functions)* Suppose a min-max function f is non-degenerate. Then $\chi(f)$ is upper and lower bounded.

Proof: Indeed, Lemma 2.2 in [17] says that $\|\chi(f)\|_\infty \leq \min_y \|f(y) - y\|_\infty$. So, for instance, $\|\chi(f)\|_\infty \leq \|f(0)\|_\infty$. $\|f(0)\|_\infty$ is bounded by the non-degeneracy hypothesis. And the proof is complete. ■

The concept of mean-payoff games is useful for solving two specific problems that will appear on this text.

B. Modified two-sided eigenproblem

Problem 2: (*Two sided-eigenproblem, see [15]*) A *two-sided eigenproblem* can be stated as the problem of solving the equation $Uy = \lambda V y$ for the unknowns λ and $y \neq \perp$. ■

Given a two-sided eigenproblem $Uy = \lambda V y$, it can be written as

$$\begin{pmatrix} U \\ \lambda V \end{pmatrix} y \preceq \begin{pmatrix} \lambda V \\ U \end{pmatrix} y$$

and hence as $G(\lambda)y \preceq H(\lambda)y$. The problem of finding a non-trivial solution $y \neq \perp$ to this problem can then be stated as the problem of finding λ such that *at least* one of the scalars $\chi_j(f_\lambda)$ is non-negative. In this case, this can be written equivalently as $\max_j \chi_j(f_\lambda) \geq 0$. Due to the structure of these particular matrices $G(\lambda)$ and $H(\lambda)$, it can be shown that $\chi_j(f_\lambda) \leq 0$ and hence the problem can be stated as simply $\max_j \chi_j(f_\lambda) = 0$, since there is no hope in obtaining positive values (see [15]).

The function $s(\lambda) \equiv \max_j \chi_j(f_\lambda)$ is the *spectral function* associated to the mean-payoff game [15]. The problem is then reduced to find a zero of this spectral function, which is piecewise affine Lipschitz, non-convex (in general) and non-positive [15]. Once a λ is found, any algorithm for solving max-plus linear equations can be used to find a y, which will be guaranteed to be $\neq\perp$, such that $Uy = \lambda V y$. See [13], [14], [15], [21], [2], [11], [10], [29] and the references therein for examples of algorithms to solve max-plus linear equations.

A pseudo-polynomial method for finding zeros to this spectral function, and hence solving Problem 2, was developed in [15]. Although the method requires that the entries of U and V are integers or $\perp$, this can be assumed without loss of generality in the dioid $\mathbb{Q}_{\max}$ which has rational or $\perp$ entries. Indeed, if U, V do not have all entries integers or $\perp$, but also non-integer rationals, one can always redefine the units of the problem by multiplying (in traditional algebra) all the entries of U,V by an integer d such that all entries of the new matrices U', V' are integers or $\perp$.

However, in this work one is interested in finding solutions of $Uy = \lambda V y$ such that a *subset* of entries y_i of y, those in which the index i is in a given set $\mathcal{I}$, are non $\perp$. This is a *modified two-sided eigenproblem*.

Problem 3: (*Modified two sided-eigenproblem*) A *modified two sided-eigenproblem*, for a given $\mathcal{I}$, is the problem of solving $Uy = \lambda V y$ such that $y_i \neq \perp \ \forall i \in \mathcal{I}$. ■

It will be argued further in this text, specifically in Proposition 5, that for some particular cases of problems $Uy = \lambda V y$, the solution for the associated two-sided eigenproblem is also a solution for the associated modified one for a special subset $\mathcal{I}$. These cases will be the only ones present in this text, so it is enough consider only Problem 2.

C. Distance between max-plus affine spaces

Definition 7: (Chebyshev distance, see [15]) The *Chebyshev distance* $d(x, y)$ between two vectors x and y with common support $\mathcal{R}$, that is, $x_i \neq\perp$ if and only if $y_i \neq\perp$, is defined as

$$d(x, y) \equiv \max_{i \in \mathcal{R}} |x_i - y_i|$$

in which $\mathcal{R} = \{i \mid x_i \neq\perp\}$.

■

If x, y are upper and lower bounded, the Chebyshev distance $d(x, y)$ coincides with the distance induced by the ∞-norm $\|x - y\|_\infty$. Since all the vectors x, y that will be handled in this context are lower and upper bounded, the notation $\|x - y\|_\infty$ will be used instead of $d(x, y)$.

One can then consider the following problem:

Problem 4: (*Computing the Chebyshev distance between max-plus affine spaces*) Consider two max-plus affine spaces $\mathcal{U} = u \oplus \operatorname{Im} U$ and $\mathcal{V} = v \oplus \operatorname{Im} V$. The problem of *computing the Chebyshev distance between $\mathcal{U}$ and $\mathcal{V}$*, $d(\mathcal{U}, \mathcal{V})$, is defined as

$$d(\mathcal{U}, \mathcal{V}) \equiv \min_{\substack{u' \in \mathcal{U} \\ v' \in \mathcal{V}}} \|u' - v'\|_\infty .$$

■

To handle this problem, an important property of the Chebyshev distance must be presented.

Property 1: (see Equation 26 in [15], or even by inspection) For two lower-and-upper bounded vectors $x, y \in \mathbb{Q}^n_{\max}$

$$\|x - y\|_\infty = \min\{\delta \in \mathbb{R} \mid \ \delta x \succeq y \ , \ \delta y \succeq x\}.$$

Further, the strict equality is achieved for at least one index in the first or in the second inequality. ■

In order to deal with Problem 4, the following proposition will be important.

Proposition 2: (*Characterization of the components of* χ) Let $f(y) = G\backslash(Hy)$ be non-degenerate. The following holds

$$\chi_j(f) = \max\{\lambda \in \mathbb{R} \ \mid \exists y \in \mathbb{Q}^n_{\max}, \ y_j \neq\perp, \ \lambda y \preceq f(y)\}.$$

Proof: The proof is inspired by the ones presented in [1]. Clearly, using the fact that min-max functions are non-decreasing, one has that $\lambda y \preceq f(y)$ implies $\lambda^N y \preceq f^N(y)$. Let y be a vector in which $y_j \neq\perp$, and z a vector obtained by replacing all $\perp$ entries of y by any finite rational entry. As $y \preceq z$ one has $f(y) \preceq f(z)$, again using the fact min-max functions are non-decreasing, and hence $f^N(y) \preceq f^N(z)$. Therefore $\lambda^N y \preceq f^N(z)$. Dividing, as in traditional algebra, both sides by N and taking the limit as $N \to \infty$, one concludes using Equation (6), the fact that z only has rational entries and the fact that y_j is finite that $\lambda \preceq \chi_j(f)$.

It will be shown that this value $\chi_j(f)$ is achievable. Indeed, consider the function $f'(y) = \chi_j(f)^{-1} f(y)$. Notice that the operation $\chi_j(f)^{-1}$ is valid: $\chi_j(f)$ is not $\perp$ since f is non-degenerate. See Corollary 1. By virtue of Equation (6), it is clear that if f is max-plus homogeneous, $f(\alpha y) = \alpha y$ for any scalar α, then $\chi(\alpha f) = \alpha \chi(f)$. $f(y) = G\diagdown(Hy)$ is clearly max-plus homogeneous and therefore one has that $\chi_j(f') = \chi_j(f)^{-1}\chi_j(f) = 0 \geq 0$. Then, Theorem 1 states that there exists a w with $w_j \neq \perp$ such that $Gw \preceq \chi_j^{-1} Hw$ and therefore $\chi_j Gw \preceq Hw$. After residuation one concludes that $\chi_j w \preceq G\diagdown(Hw) = f(w)$. This shows that $\lambda = \chi_j(f)$ is achievable with this vector w. And the proposition is proved. ■

Proposition 2 and Property 1 allow one to conclude the following.

Proposition 3: (*Computing the distance of max-plus affine spaces*) Let $\mathcal{U} = u \oplus \operatorname{Im} U$ and $\mathcal{V} = v \oplus \operatorname{Im} V$ be affine spaces. Let

$$G = \begin{pmatrix} u & U & \perp \\ v & \perp & V \end{pmatrix},\ H = \begin{pmatrix} v & \perp & V \\ u & U & \perp \end{pmatrix} \quad (7)$$

and $f(w) = G\diagdown(Hw)$. Suppose f is non-degenerate. Then

$$d(\mathcal{U}, \mathcal{V}) = \chi_0(f)^{-1}$$

in which 0 is the index of the first column of G, H.

Proof: One has

$$d(\mathcal{U}, \mathcal{V}) = \min_{\hat{u},\hat{v}} \|(u \oplus U\hat{u}) - (v \oplus V\hat{v})\|_\infty. \quad (8)$$

Let $t \neq \perp$. Using the fact that the distance $\|u' - v'\|_\infty$ is invariant by max-plus multiplications by a $t \neq \perp$: $\|u' - v'\|_\infty = \|u't - v't\|_\infty$

$$d(\mathcal{U}, \mathcal{V}) = \min_{\hat{u},\hat{v},t\neq\perp} \|(ut \oplus U\hat{u}t) - (vt \oplus V\hat{v}t)\|_\infty.$$

Let $\hat{u}t = \widetilde{u}$, $\hat{v}t = \widetilde{v}$ and $w = (t\ \widetilde{u}^T\ \widetilde{v}^T)^T$. Using Property 1 with n being the number of columns of G, H

$$d(\mathcal{U}, \mathcal{V}) = \min\{\delta \in \mathbb{R} \mid \exists w \in \mathbb{Q}^n_{\max},\ t \neq \perp,\ \delta H w \succeq Gw\}.$$

Or, letting $\lambda = \delta^{-1}$ and using the fact that $w_0 = t$

$$d(\mathcal{U}, \mathcal{V})^{-1} = \max\{\lambda \in \mathbb{R} \mid \exists w \in \mathbb{Q}^n_{\max},\ w_0 \neq \perp,\ \lambda G w \preceq Hw\}.$$

Using residuation

$$d(\mathcal{U}, \mathcal{V})^{-1} = \max\{\lambda \in \mathbb{R} \mid \exists w \in \mathbb{Q}^n_{\max},\ w_0 \neq \perp,\ \lambda w \preceq G\diagdown(Hw)\}.$$

And hence, in light of Proposition 2 (f is non-degenerate by hypothesis), $\lambda = \chi_0(f)$ and thus $\delta = \lambda^{-1} = \chi_0(f)^{-1}$. And the proposition is proved. ■

By a quick glance in the algorithm proposed in [14] for computing the value of mean payoff games, it is clear that if the entries of G and H are rational numbers or $\perp$ (hence in $\mathbb{Q}_{\max}$) and f is non-degenerate, $\chi_0(f)$ is a rational number, since the rationals are closed for all the operations (for which the number is finite) in the algorithm: maxima/minima, traditional sums/subtractions and traditional products/divisions. Hence, $\chi_0(f) \in \mathbb{Q}$.

It is also of interest to compute the vectors $\hat{u}, \hat{v}$ which ensure this distance in Equation (8). Once the distance δ is found, one needs to solve the max-plus linear equation $\delta H w \succeq Gw$ with $w_0 \neq \perp$. Note that a vector with this characteristic, $w_0 \neq \perp$, exists due to the definition of the distance. A suggestion to find a solution is to use the iterative procedure given in [1]: in order to solve a max-plus linear equation $Jw \preceq Kw$ iterate

$$w[k+1] = J\diagdown(Kw[k]) \wedge w[k], \quad (9)$$

which is convergent to the greatest solution smaller than or equal to the initial guess $w[0]$. The problem is that the sequence in Equation (9) can take an infinite amount of steps to converge. This happens only if there is an entry of w that necessarily must be $\perp$. For instance, the equation $w \preceq (-1)w$, in which w is a scalar, is such an example because it will take $k \to \infty$ to $w[k+1] = (-1)w[k]$ to converge to the only solution ($w = \perp$).

This behaviour can be avoided with the help of Theorem 1 and also the handy results in [30]. In the latter work it is shown that, if the hypothesis that the entries of J and K are either integers or $\perp$ holds true, the cycle-time vector χ associated to the function $f(w) = J\diagdown(Kw)$ can be computed in finite time, more precisely, in pseudo-polynomial time. This hypothesis can always assumed to be the true if one is working in $\mathbb{Q}_{\max}$, since the units can be scaled to integers (see the discussion in Subsection IV-B).

The technique shown in [30] relies in the iterations of the function $f(w)$ (value iteration). So, in possession of this information, one can use Theorem 1 to characterize the entries j in which $\chi_j < 0$ so these respective entries in w_j can be set to $\perp$ and removed from w, obtaining a new (thus far) unknown vector w'. In this way, the reduced equation, after removing the variables in which $w_j = \perp$, $J'w' \preceq K'w'$ has a solution w' in which all the entries are non-$\perp$ and the iteration algorithm given by Equation (9), which will assuredly converge in a finite number of steps, can be used in this new equation. Finally, owing to the fact that $Jw \preceq Kw$ is equivalent to $(J^T\ I^T)^T w \preceq (K^T\ I^T)^T w$, since the constraint $w \preceq w$ is innocuous, the dynamic programming operator for this equation is $f'(w) = J\diagdown(Kw) \wedge w$ and hence the sequence in Equation (9) can be used to both compute the cycle-time vector χ and to find the solution w.

Since the sequence in Equation (9) has the property of converging to the greatest solution smaller than or equal to the initial guess $w[0]$, one can set $w_0[0] = 0$ and all the other entries as sufficiently large numbers and the algorithm will converge to a solution in which w_0 is already 0 and hence $\hat{u}$ and $\hat{v}$ can be extracted directly from w. Nevertheless, if a solution with $w_0 \neq 0$ is found (it will be a finite number, however), one can compute a new solution w' in which $w'_0 = 0$ by simple subtracting (in traditional algebra) each entry of w by w_0: $w' = w_0^{-1} w$.

V. (A,B) Max-plus Geometrical Invariant Sets

A. Approach for general max-plus multiplicatively invariant sets

A key concept for solving Problem 1 is the one of *(A,B) max-plus geometrical invariance* of sets.

Definition 8: ((A,B) max-plus geometrical invariant sets, see [20]) A set $\mathcal{N} \subseteq \mathbb{Q}_{\max}^n$ is said to be (A,B) *max-plus geometrical invariant* if it is a semimodule and for every $x \in \mathcal{N}$ there exists an $u \in \mathbb{Q}_{\max}^m$ such that $Ax \oplus Bu \in \mathcal{N}$. ∎

Now a proposition, which will be helpful for finding an (A,B) max-plus geometrical invariant in the set of constraints $\mathcal{X}_{\text{cons}}$, will be presented.

Proposition 4: (An (A,B) max-plus geometrical invariant set): Let $\mathcal{X}_{\text{cons}}$ be a max-plus multiplicatively invariant set and L, R max-plus homogeneous functions such that $\mathcal{X}_{\text{cons}}$ is characterized by the equation $L(x) = R(x)$ (such functions always exist, see the comment after Definition 1) . Consider a solution triple $\{\lambda, q, w\}$ to the equation

$$\begin{aligned} L(q) &= R(q); \\ Aq \oplus Bw &= \lambda q. \end{aligned} \tag{10}$$

Then the set $\mathcal{N}(q) = \{\alpha q \mid \alpha \in \mathbb{Q}_{\max}\}$ is an (A,B) max-plus geometrical invariant set in $\mathcal{X}_{\text{cons}}$.

Proof: By the first Equation in (10), it is clear $\mathcal{N}(q) \subseteq \mathcal{X}_{\text{cons}}$. It only remains to prove that this set is an (A,B) max-plus geometrical invariant set.

Therefore, it is sufficient and necessary to show that for any $x \in \mathcal{N}(q)$, that is, $x = \alpha q$ for a given α, there exists u such that $Ax \oplus Bu$ remains in $\mathcal{N}(q)$. If $x = \alpha q$, choose $u = \alpha w$, and it is clear by the second Equation in (10) that $Ax \oplus Bu = \lambda x$ and thus in $\mathcal{N}(q)$. Further, $\mathcal{N}(q)$ is clearly an one-dimensional semimodule. And the proof is complete. ∎

Solving Equation (10) for general functions L, R can be a hard problem. A very general technique for solving this equation for a wide class of such functions L and R is formulating them as an Extended Linear Complementarity Problem and then solving it with an appropriate solver, as in [27]. However, this approach can take a prohibitive amount of time.

B. A special case: finitely generated semimodules

This text will discuss an important special case of a max-plus multiplicatively invariant set: a finitely generated semimodule of $\mathbb{Q}_{\max}$, which can be characterized implicitly as the set of all solutions x of a max-plus linear equation (see [22])

$$Ex = Dx \tag{11}$$

for given matrices $E, D \in \mathbb{Q}_{\max}^{q \times n}$ (that is, $L(x) = Ex$, $R(x) = Dx$). This is the most general form of max-plus linear constraints, and was considered in [20], [24], [16]. The first two papers, however, do not consider the implicit formulation given by Equation (11). They instead require that all solutions of Equation (11) are found, which may be in general a time consuming problem. Nevertheless, in this case, finding a solution to Equation (10) is equivalent to solving the equation

$$\begin{aligned} Eq &= Dq; \\ Aq \oplus Bw &= \lambda q. \end{aligned} \tag{12}$$

Equation (12) is max-plus nonlinear for the parameters $\{\lambda, q, w\}$. However, it can be transformed in a two-sided eigenproblem (Problem 2). To this, multiply both sides of the first equation in Equation (12) by λ. The resulting set of equations will be equivalent as long as λ is invertible ($\lambda \neq \perp$), which is a very weak assumption. Now, substitute the second equation in the first one, but only in the *left* side of the first equation. The resulting equation is

$$\begin{aligned} EAq \oplus EBw &= \lambda Dq; \\ Aq \oplus Bw &= \lambda q. \end{aligned} \tag{13}$$

If one defines $y \equiv (q^T \ w^T)^T \in \mathbb{Q}_{\max}^{n+m}$, it is clear that Equation (13) can be written as $Uy = \lambda V y$ for appropriate matrices $U, V \in \mathbb{Q}_{\max}^{(n+q)\times(n+m)}$, namely

$$U = \begin{pmatrix} EA & EB \\ A & B \end{pmatrix}, \qquad V = \begin{pmatrix} D & \perp \\ I & \perp \end{pmatrix} \tag{14}$$

and hence the problem of solving Equation (12) can be stated as a two-sided eigenproblem. However, soon it will be clear that one needs to find a solution triple $\{\lambda, q, w\}$ of Equation (12) such that q has no $\perp$ entries, that is, a modified two-sided eigenproblem (Problem 3) in which $\mathcal{I} = \{1, 2, ..., n\}$ (the entries of y relative to q). However, it turns out that for some particular cases of problems, a solution to a two-sided eigenproblem is also a solution to the modified one. Indeed:

Proposition 5: (Modified two-sided eigenproblem can be reduced to a traditional one) Consider a problem as in Problem 1 under Hypothesis 3.1 with the set $\mathcal{X}_{\text{cons}}$ given by the solutions of the equation $Ex = Dx$. In this case, the respective Equation (13) is such that any solution $y \equiv (q^T \ w^T)^T$ such that $y \neq \perp$ has q without any $\perp$ entries. Hence, a solution to the two-sided eigenproblem associated to Equation (13) is also a solution to the associated modified two-sided eigenproblem in which $\mathcal{I}$ is the set of entries respective to q in y.

Proof: If $y \neq \perp$, then either q or w (possibly both) has at least a single non-$\perp$ entry.

Suppose the former, that is, q has at least a single non-$\perp$ entry. Under this supposition, the fact that the associated problem is coupled implies that q has no $\perp$ entries. This is true because $x[k] = \lambda^k q$ is a possible trajectory to the system (pick $x[0] = q$ and $u[k+1] = \lambda^k q$), and hence the coupled hypothesis implies that $q_i - q_j$ is always bounded. Since there exists an i such that $q_i \neq \perp$, this implies that q_j is also not $\perp$. This concludes the proof for the case that q has at least a single non-$\perp$ entry.

Suppose the latter, that is, w has at least a single non-$\perp$ entry. Now, according to Equation (13), $q \succeq Bw$ and, since the problem is non-degenerate in the control sense, Bw is not the $\perp$ vector and hence q has at least one non-$\perp$ entry. In this case, the same argument in the previous case holds and q has no $\perp$ entry. This concludes the proof. ■

It is important to mention that [20] presents an iterative method, for which the number of steps is finite under certain assumptions, that computes the greatest (A, B) max-plus geometrical invariant set in a given finitely generated semimodule, considering the knowledge of a matrix M which column span generates this set. However, this computation can be very complex in terms of memory and time consuming. In fact, even obtaining M can be a complex problem, since in general this semimodule is described implicitly as a max-plus linear equation, and writing this semimodule as $\operatorname{Im} M$ requires finding *all* the solutions to this equation, for which no very efficient algorithm is known yet (again, the reader is invited to see the complexity of algorithms such as [21], [2], [10], [29], which compute the entire set of solutions of max-plus linear equations). In contrast, the proposed method computes only a subspace of this set, but is simpler and therefore can be, generally, used in problems of higher dimensions.

VI. The Periodic Synchronizer

A. Formalization

One shall show that Problem 1 can be solved using a periodic input $u[k+1] = h\lambda^{k+1}w$. To this end, it is important to recall the following property of the Kleene closure:

Property 2: (see Lemma 2.2 in [18]) For any matrix $M \in \mathbb{Q}_{\max}^{n\times n}$ with non-positive spectral radius there exists a natural number $r' \leq n$ such that for all $r \geq r'$

$$M^* = \bigoplus_{i=0}^{r} M^i. \tag{15}$$

■

Taking this into consideration, one makes the following definition.

Definition 9: (*Convergency number*) For a given matrix M with $\rho(M) \leq 0$, the *convergency number of* M, $\kappa(M)$, is defined as the smallest r' in Property 2 that ensures Equation (15). ■

Note that computing $\kappa(M)$ often comes as a byproduct of the computation of M^*, and therefore can be done in polynomial time.

Hence, it is possible to state the principal result of this section.

Proposition 6: (Periodic synchronizer) Consider Problem 1, with $\mathcal{X}_{\texttt{cons}}$, L and R as in Proposition 4, and assume that there exists a solution triple of Equation (10), $\{\lambda, q, w\}$ such that q is lower bounded and $\lambda > \rho(A) \neq \perp$. Then, for any scalar $h \neq \perp$, the periodic input $u[k+1] = h\lambda^{k+1}w$ guarantees that there exists a k' such that $x[k] \in \mathcal{X}_{\texttt{cons}}$ (i.e $L(x[k]) = R(x[k]))$ $\forall k \geq k'$, thus solving Problem 1.

Proof: Using $u[k+1] = h\lambda^{k+1}w$, multiplying both sides of this equation by $\lambda^{-(k+1)}$ and using the change of variables $\hat{x}[k] = \lambda^{-k}x[k]$, the system in Equation (3) can be written as

$$\hat{x}[k+1] = (\lambda^{-1}A)\hat{x}[k] \oplus hBw. \tag{16}$$

Hence, for any k, by iterating Equation (16):

$$\hat{x}[k+1] = \underbrace{(\lambda^{-1}A)^{k+1}x_{\texttt{ic}}}_{G[k]} \oplus \underbrace{h\left(\bigoplus_{i=0}^{k}(\lambda^{-1}A)^i\right)Bw}_{H[k]}.$$

Now, it is possible to note the following facts:

1) Let $k \geq \kappa(\lambda^{-1}A)$. By the definition of $\kappa(\lambda^{-1}A)$ (see Definition 9), $H[k] = h(\lambda^{-1}A)^*Bw$;
2) By the second equation in Equation (10) and the fact that $\lambda > \rho(A)$, one concludes that $(\lambda^{-1}A)^*Bw = \lambda q$;
3) Since $\lambda > \rho(A)$, $G[k] \to \perp$ as $k \to \infty$. Considering this and also that, by Fact 2, $H[k] = \lambda hq$ for $k \geq \kappa(\lambda^{-1}A)$ and that $H[k]$ is lower bounded by hypothesis (since q is lower bounded by hypothesis and $\lambda, h \neq \perp$), there exists a finite $r \geq \kappa(\lambda^{-1}A)$ such that $H[k] \succeq G[k]$ for all $k \geq r$;

Take $k' = r+1$. Due to three previous facts, it is clear that for all $k \geq k'$, $\hat{x}[k] = H[k-1] = h(\lambda^{-1}A)^*Bw = h\lambda q$ or equivalently $x[k] = h\lambda^{k+1}q$ for all $k \geq k'$. Therefore, $x[k]$ is a scalar multiple of q for $k \geq k'$. Further, q is a member of the constraint set due to the first equation in Equation (10), and since the constraint set is max-plus multiplicatively invariant, so $x[k] \in \mathcal{X}_{\texttt{cons}}$ for $k \geq k'$. And the proof is complete. ■

With the results so far, one can readily conclude the following.

Corollary 2: of Proposition 6: (Bound on the number of steps and convergency for $\lambda = \rho(A)$). Let $t \equiv \kappa(\lambda^{-1}A)$ and h in Proposition 6 be such that

$$h\lambda q \succeq (\lambda^{-1}A)^{t+1}x_{\texttt{ic}} \tag{17}$$

(such h always exists because λq is lower bounded) then convergence to the desired set occurs in at most $t+1$ steps. This is also true even when $\lambda = \rho(A)$, provided that Equation (10) is modified to

$$\begin{aligned} L(q) &= R(q); \\ q &= (\lambda^{-1}A)^*B(\lambda^{-1}w). \end{aligned} \tag{18}$$

Proof: Note that if Equation (17) holds, then r in Fact 3 in Proposition 6 can be taken to be $\kappa(\lambda^{-1}A)$. And the first assertion is proved.

In order to show that convergency occurs even when $\lambda = \rho(A)$, one analyzes the three facts presented in Proposition 6. Fact 1 holds even when $\lambda = \rho(A)$. Fact 2 is not true for $\lambda = \rho(A)$, since the second equation in Equation (10) has an infinite number of solutions for q in this case. This is handled by the modified Equation (18) which ensures the uniqueness. Finally, in Fact 3, it is not true that $G[k] \to \perp$ as $k \to \infty$ if $\lambda = \rho(A)$, but Equation (17) guarantees that there exists a s such that $H[k] \succeq G[k]$ for all $k \succeq s$ $(s = \kappa(\lambda^{-1}A))$. ■

B. The algorithm

The proposed algorithm generates, as input, a *periodic* signal in the system till the system eventually *synchronizes* (and thus the name) with this input and becomes periodic too. The algorithm can then be presented in a succinct way as follows:

Algorithm 6.1: Control algorithm: periodic synchronizer

1) Solve Equation (10) (for $\lambda > \rho(A)$), or Equation (18) (for $\lambda = \rho(A)$) with q with no $\perp$ entries (see Problem 3 and Subsection IV-B) obtaining a triple $\{\lambda, q, w\}$;
2) (Optional if $\lambda > \rho(A)$, necessary if $\lambda = \rho(A)$) Find h such that Equation (17) holds, thus insuring convergence in at most $\kappa(\lambda^{-1}A) + 1$ steps;
3) Take the control input of the system as $u[k+1] = h\lambda^{k+1}w$.

Perhaps the most striking feature of the Periodic Synchronizer, specially to control theorists, is the fact that for $\lambda > \rho(A)$ it operates in open loop: no information on the current state is needed to perform the control. For $\lambda = \rho(A)$, however, it is necessary to consider the initial condition, see Corollary 2 and Equation (17). Nevertheless, the procedure is robust in the sense that a sporadic perturbation in the event occurrence dates can be considered as a new initial condition, and then eventually the system will again converge to the desired set, since convergence is ensured for any initial condition. This behavior will be illustrated in Subsection IX-B. This discussion suggests that TEGs are, in some sense, inherently "stable" since a periodic input eventually synchronizes to a periodic output. The interesting question in the context is exactly if there exists an input which synchronizes with an output in the desired set.

Of course, the periodic input u that guarantees the desired behavior depends on the parameters A, B of the system. Thus, if there is a longstanding change of behavior of the system as, for example, a modification on the matrices A,B, this periodic input may be unable to solve the desired problem. Due to this fact, it is advisable to maintain some on-line identification system, which operates in closed-loop, that eventually recomputes the new inputs when there is a significant change in the system structure.

Finally, it is important to mention that [18] proposes a method similar to the Periodic Synchronizer. The method was presented in the context of creating timetables to trains, but could be in principle adapted to any kind of system that fits in its framework. Although the authors do not discuss this subject, the problem of generating timetables can be rephrased as a control problem and then it is comparable to the proposed approach. There, the method is applied to systems such that $B = I$ and its aim is to make the system follow a periodic timetable $x[k] = d[k] = \lambda^k q$ for a given scalar $\lambda > \rho(A)$ and vector q. Hence, as opposed to this paper, the state constraints are given explicitly instead of a solution of an equation as $L(x) = R(x)$. It is shown that it is necessary and sufficient to guarantee the timetable that $\lambda q \succeq Aq$ and $\lambda > \rho(A)$. Indeed, in the case that $B = I$, the second equation in Equation (18) reduces to $\lambda q = Aq \oplus w$ which is exactly $\lambda q \succeq Aq$. Hence, the result shown in the present paper generalizes the one found in [18], at least as far as the "sufficiency" of the result is concerned. In addition, the assumption $B = I$ can be quite restrictive. Further in this paper, a control problem inspired in a TEG presented in the same reference, [18], is solved with the Periodic Synchronizer. However, in this case $B \neq I$ and hence it does not fit in the framework presented in [18].

C. Computational complexity

In terms of computational complexity, the only potentially complex part of Algorithm 6.1 is solving Equation (10) with the constraint that q has no $\perp$ entries. For the general problem the complexity is hard to describe since it depends on the functions L and R.

If constraints as Equation (11) are considered, the issue is then solving Equation (13) with the same constrain in q. This is a modified two-sided eigenproblem, Problem 3. Under Hypothesis 3.1, Problem 3 reduces to Problem 2 (see Proposition 5). In this case, it was shown in [15] that solving the Problem 2 associated to the equation $Uy = \lambda Vy$ has pseudo-polynomial complexity. Let $U, V \in \mathbb{Q}_{\max}^{s \times t}$, in which U and V has only $\perp$ and integer entries (this can be assumed without loss of generality, see Subsection IV-B). In this case, let $TSE(s, t, U, V)$ be the computational complexity of solving Problem 2 associated to the equation $Uy = \lambda Vy$ (see [15] for the exact expression for $TSE(s, t, U, V)$). Hence, if $A \in \mathbb{Q}_{\max}^{n \times n}$, $B \in \mathbb{Q}_{\max}^{n \times n}$ and $E, D \in \mathbb{Q}_{\max}^{q \times n}$, then the complexity of this algorithm is given by $TSE(n+q, n+m, U, V)$, which is pseudo-polynomial (see [15]), with U and V given by Equation (14).

VII. The Feedback Accelerator

A. Formalization

The Periodic Synchronizer presented in the previous section may generate a long transient state till the system stabilizes. According to the previous discussions in Subsection VI-A, there are two factors that impact the number k', which is an upper bound of the number of steps taken for convergence to the desired set. The first one is the convergence number, $t \equiv \kappa(\lambda^{-1}A)$. The second is the number r in Proposition 6, which depends on the system parameters and also the choice of h (it is easy to see that r is non-increasing with the parameter h. Since the latter is generally easily controlled by choosing the appropriate h according to Equation (17), it is advantageous to focus the effort in improving the former.

One relatively simple idea to try to improve the convergence number is to "pre-close" the loop with a linear feedback, while still keeping a free term on the control input: $u[k+1] = Fx[k] \oplus g[k+1]$. Thus, using this input on the system given by Equation (3):

$$S_{\mathrm{acc}} : \begin{cases} x[k+1] = (A \oplus BF)x[k] \oplus Bg[k+1] \text{ for } k > 0 \\ x[0] = x_{\mathrm{ic}} \end{cases}$$

Let $A_{\mathrm{acc}} \equiv A \oplus BF$. The objective is thus to design F in a way that the convergence number $\kappa(\lambda^{-1}A_{\mathrm{acc}})$ is as smaller as possible. After this, Algorithm 6.1 can be used in this new system using the control input $g[k]$.

Ideally, $A \oplus BF = \alpha Q^*$ for a matrix Q and a scalar α. Indeed, if $\lambda \geq \alpha$, the convergence number $\kappa(\lambda^{-1}(A \oplus BF))$

would be 1, the smallest possible, except for the very particular case of a diagonal matrix with non-positive entries, which has a convergence number of 0, and the Periodic Synchronizer would converge very quickly. However, this equation for the unknowns F, Q, α is not max-plus non-linear and hard to solve.

Inspired by the ideas on Section V, one can use $\alpha = \lambda$ and $Q = (\lambda^{-1}A)$. The resulting equation is now max-plus affine and easily solvable, if it has a solution. However, frequently such solution does not exists. Thus, one possibility is to weaken the problem by finding the greatest F solution to the inequation

$$A_{\mathtt{acc}} = A \oplus BF \preceq \lambda(\lambda^{-1}A)^* \tag{19}$$

and if a solution to the non-weakened equation exists, such F will also be found by solving this weakened equation. See Section 3.2.3.2 in [8]. Since naturally $\lambda(\lambda^{-1}A)^* \succeq A$, Equation (19) is equivalent to $BF \preceq \lambda(\lambda^{-1}A)^*$.

Since the causality of F is necessary, the discussion in Subsection III-C is applicable. In this case, $M = \lambda(\lambda^{-1}A)^*$. Hence, one can obtain the causal F as

$$F = \mathcal{C}_{\mathrm{p}}(B \backslash\!\!\circ (\lambda(\lambda^{-1}A)^*)). \tag{20}$$

B. Effects on the system

A natural question is how this linear feedback (Equation (20)) affects the system: if it is really capable of improving the system performance, in what regard, and if it can also occasionally be deleterious.

First, it can be proved that this approach at least maintains the convergence number. This can be shown using the following result.

Proposition 7: (Inequality in the convergence number) Let $X \succeq Y$, $X \preceq Y^*$, with $\rho(X), \rho(Y) \leq 0$. Then $\kappa(X) \preceq \kappa(Y)$.

Proof: Let $t \equiv \kappa(Y)$. In this case $Y^* = \bigoplus_{i=0}^{t} Y^i$. Since $X \succeq Y$, $\bigoplus_{i=0}^{t} X^i \succeq \bigoplus_{i=0}^{t} Y^i$. By $Y^* \succeq X$, one can conclude that $(Y^*)^k = Y^* \succeq X^k$. Finally, one has that $\bigoplus_{i=0}^{t} X^i \succeq \bigoplus_{i=0}^{t} Y^i = Y^* \succeq X^k$, for any k. This final conclusion, $\bigoplus_{i=0}^{t} X^i \succeq X^k$, implies that $X^* = \bigoplus_{i=0}^{t} X^i$. This implies that $\kappa(X)$ is at most t and the proof is complete. ∎

Using $Y = \lambda^{-1}A$, $X = \lambda^{-1}A_{\mathtt{acc}}$, one can see by using Proposition 7 that, for any $A_{\mathtt{acc}}$ such that Equation (19) holds, $\kappa(\lambda^{-1}A_{\mathtt{acc}}) \preceq \kappa(\lambda^{-1}A)$. Thus, as claimed, the feedback approach in the worst case maintains the convergence number.

Another concern is that the approach may reduce the set of solutions of Equation (10). Thus, one may ask about how these two equations, the one with A and other with $A_{\mathtt{acc}}$, compare with each other in regard to the solution set. It turns out that, as long as λ is fixed (since $A_{\mathtt{acc}}$ depends on λ) and strictly greater than $\rho(A)$, the solutions sets are equal, and so the linear feedback neither increases nor decreases the number of solutions.

Proposition 8: (Equality of sets) For a given $\lambda > \rho(A)$, for any $A_{\mathtt{acc}}$ such that Equation (19) holds, the solution set $\{q, w\}$ of

$$Aq \oplus Bw = \lambda q \tag{21}$$

is the same as the one of

$$A_{\mathtt{acc}} q \oplus Bw = \lambda q. \tag{22}$$

Proof: Since $\lambda > \rho(A)$, Equation (21) is equivalent to $q = (\lambda^{-1}A)^* B(\lambda^{-1}w)$, while Equation (22) is equivalent to $q = (\lambda^{-1}A_{\mathtt{acc}})^* B(\lambda^{-1}w)$.

Now, let $Y = \lambda^{-1}A$, $X = \lambda^{-1}A_{\mathtt{acc}}$. So as in Proposition 7, $X \succeq Y$, $X \preceq Y^*$. Due to the monotonicity of the Kleene Closure, applying it to both sides one concludes from the first equation that $X^* \succeq Y^*$ and from the second $X^* \preceq Y^*$. Thus $X^* = Y^*$ and then $(\lambda^{-1}A)^* = (\lambda^{-1}A_{\mathtt{acc}})^*$. This concludes the proof. ∎

Again, if $\lambda = \rho(A)$, Equation (18) must be used instead of Equation (10). In this case, it follows as a corollary of the result concluded in Proposition 8, namely that $(\lambda^{-1}A)^* = (\lambda^{-1}A_{\mathtt{acc}})^*$, that the solutions sets are equal.

Finally, it is important to note that the number of steps taken for convergence may not be an adequate measure of performance: a low number of steps may be needed for convergence, but the time elapsed since the initial firings till such step may be large since the gap of time between consecutive firings can be very large. Conversely, a high number of steps may be necessary for convergency but the elapsed time since the initial firings can be small, since the gap between consecutive firings can be very small.

Thus, one may use as a measure of performance the *elapsed time vector* $T \equiv x[k] - x[0]$, in which k is the first step in which convergence occurs. Thus, a natural question is how the T of the original system compares with the $T_{\mathtt{acc}}$ of the accelerated one, given the same initial condition $x[0]$.

Consider that h is chosen according to Equation (17). In Algorithm 6.1, convergency occurs with $x[k] = h\lambda^{k+1}q$, with any h such that Equation (17) holds. If one is interested in a small elapsed time vector, the smallest h such that it holds can be chosen. Further, due to Proposition 8, the same $(\lambda^{-1}A)^*Bw$ can be chosen for both the normal and accelerated system. Thus, the comparison is reduced between the right side of Equation (17) for the normal and accelerated system, that is, $(\lambda^{-1}A)^{t+1}x_{\mathtt{ic}}$ and $(\lambda^{-1}A_{\mathtt{acc}})^{s+1}x_{\mathtt{ic}}$, with $t = \kappa(\lambda^{-1}A)$ and $s = \kappa(\lambda^{-1}A_{\mathtt{acc}})$. It turns out that, thus far, there is no formal guarantee that this index will be no greater, since the accelerated convergence number s can be smaller than t but the matrix $A_{\mathtt{acc}}$ can be larger than A. Thus, one cannot say in general that $(\lambda^{-1}A_{\mathtt{acc}})^{s+1}x_{\mathtt{ic}} \preceq (\lambda^{-1}A)^{t+1}x_{\mathtt{ic}}$.

Despite the feedback appealing features in control theory, note that Propositions 7 and 8 ensure that, in terms of convergence number and size of the (A, B) max-plus geometrical invariant, there is no loss, but maybe no gain either, when this approach is used. In terms of elapsed time, however, the approach can be detrimental. However, experimental results in Section IX will show that, at least for a particular example and under many difference choices of initial conditions, the proposed approach improves not only the convergence number but also this index.

C. The algorithm

The algorithm will be presented in a succinct form below:

> *Algorithm 7.1: Periodic Synchronizer with Feedback Accelerator*
> 1) Compute $F = C_{\mathrm{p}}(B \diamond (\lambda(\lambda^{-1}A)^*))$;
> 2) Close the system loop with $u[k+1] = Fx[k] \oplus g[k+1]$;
> 3) Use the Periodic Synchronizer in the new system using the input $g[k]$.

It is very important to remark that if the Feedback Accelerator technique is used, the spectral radius of the new system matrix $A_{\mathtt{acc}}$ can be eventually the *same* as the λ used. Thus, in this case it is indispensable to use h as in Equation (17) (see the discussion at the end of Subsection VI-A).

D. Computational complexity

It is clear that the difference in computational complexity between Algorithm 6.1 and Algorithm 7.1 is the computation of the Feedback Accelerator, F. This requires the computation of two scalings of a matrix, a Kleene Closure, a residuation and a causal projection. If $A \in \mathbb{Q}_{\mathtt{max}}^{n\times n}$, $B \in \mathbb{Q}_{\mathtt{max}}^{n\times n}$, all of them can be done in polynomial time. The two scalings by λ can be done in time $\mathcal{O}(n^2)$. The Kleene Closure can be computed in time $\mathcal{O}(n^3)$ using, for example, the Floyd-Warshall algorithm [26], [28]. The residuation can be computed in $\mathcal{O}(nm^2)$ and the causal projection in $\mathcal{O}(nm)$. Thus, the final complexity of computing F, provided that λ is given, is polynomial: $\mathcal{O}(n(n^2 + m^2))$.

VIII. The Chebyshev-Optimized Feedback

Algorithm 6.1 requires a special class of (A, B) max-plus geometrical invariant set as in Proposition 4: one composed of scalar multiples of a vector q. The present section aims to establish a more general approach which works with more general (A, B) max-plus geometrical invariant sets generated from the image of a matrix N. As it will be shown in Section IX, even when applied to the same problem it can be more efficient than Algorithm 6.1 and its improvement, Algorithm 7.1.

A. An inspiration from the traditional algebra

Suppose that one wishes to solve an analogue of Problem 1, but in the traditional linear time invariant system context. More precisely, x is required to be in the left null space of a matrix $G \in \mathbb{R}^{q\times n}$. Note that the max-plus equivalent to this null space constraint is Equation (11).

Let $N \in \mathbb{R}^{n\times p}$ be a generator matrix for a subset, not necessarily the entire set, of the right null space of G, so $GN = 0$. Thus, given the state $x[k]$ at a given step k one can design the controller $u[k+1]$ in a way that at the next step $k+1$ the state $x[k+1]$ is as close as possible to the column span of N. If the Euclidean metric is used to measure distance, the controller comes as solution of the following convex quadratic optimization problem (in which all the operators must be considered as in the traditional algebra)

$$\min_{u[k+1],v[k+1]} \|(Ax[k] + Bu[k+1]) - Nv[k+1]\|_2^2.$$

Simple computations show that the solution[2] is of the kind $u[k+1] = Fx[k]$, a linear feedback, for a constant matrix F.

B. Formalization

The discussed approach could be translated to the Max-plus algebra, but the manipulations would be hard and cumbersome since the Euclidean norm is not the most appropriate for handling problems in this setting. A more appropriate measure for algebraic manipulations is the ∞-norm $\|\cdot\|_\infty$, which induces the Chebyshev distance between two vectors. Thus, one approach to bring $Ax[k] \oplus Bu[k+1]$ to the span of N is minimizing the Chebyshev distance

$$\min_{u[k+1],v[k+1]} \|(Ax[k] \oplus Bu[k+1]) - Nv[k+1]\|_\infty.$$

This can be seen as the problem of computing the distance between the max-plus affine space $Ax[k] \oplus \operatorname{Im} B$ and the max-plus affine space (which is, in fact, a semimodule) $\operatorname{Im} N$. This problem, Problem 4, was already discussed in Subsection IV-C. In this case, using Equation (7)

$$G[k] = \begin{pmatrix} Ax[k] & B & \perp \\ \perp & \perp & N \end{pmatrix}, \quad H[k] = \begin{pmatrix} \perp & \perp & N \\ Ax[k] & B & \perp \end{pmatrix}. \tag{23}$$

It remains to show that the function $f_k(w) = G[k] \diamond (H[k]w)$ is non-degenerate. Indeed, in the context of control, it can be assumed without loss of generality that both vectors $Ax[k]$ and the matrix N are lower bounded. The former represents a firing time and hence cannot have $\perp$ entries. The same also holds for the latter because a $\perp$ row on N, say the i^{th} one, implies that the desired constraint set requires that the i^{th} state $x_i[k]$ must be always $\perp$ (since any vector in $\operatorname{Im} N$ will have this property), a highly implausible demand. Hence, it is clear that if w is lower bounded, so is Hw. Now, it can also be assumed without loss of generality that no column of B is $\perp$, since otherwise the corresponding control entry plays no role in the system and then can be removed. This, together with the fact that $Ax[k]$ and matrices N are lower bounded, implies that the matrix $G[k]$ has no column composed entirely of $\perp$. Hence, $G[k] \diamond (H[k]w)$ is lower bounded as long as w is lower bounded. Thus, the methodology discussed in Subsection IV-C can be used.

[2] There may be infinitely many solutions if there is a huge freedom in both the choice of $u[k+1]$ and $v[k]$ (assuming all columns of the matrix $(B \;\; -N) \in \mathbb{R}^{n\times(m+p)}$ linearly independent, if and only if $m + p > n$). In this case, it is necessary to introduce a weighting factor $\alpha\|u[k+1]\|_2^2 + \beta\|v[k]\|_2^2$, $\alpha, \beta > 0$ to ensure uniqueness.

C. The algorithm

Algorithm 8.1: Chebyshev-Optimized Feedback

1) With the current state $x[k]$, create the matrices $G[k]$ and $H[k]$ as presented in Equation (23);
2) Let $f_k(w) = G[k] \diagdown (H[k]w)$, compute $\delta = \chi_0(f_k)^{-1}$ (Problem 4, see Subsection IV-C);
3) Solve the max-plus linear equation

$$(\delta Ax[k])t \oplus \delta Br \succeq Ns; \qquad (24)$$
$$\delta Ns \succeq (Ax[k])t \oplus Br$$

for the unknowns t, r and s, with $t \neq \perp$ (Suggestion: use the iterative procedure given by Equation (9), with the initial $t[0] = 0$ and $r[0]$ and $s[0]$ as vectors with sufficiently large entries, so when convergence is achieved one has $t = 0$);
4) Compute $u[k+1] = t^{-1}r$;
5) Take the control input of the system as $u[k]$;
6) Set k to $k+1$ and go to Step 1.

Note that, as opposed to the Periodic Synchronizer, the Chebyshev-Optimized Feedback operates in closed-loop at each step. Further, that it can be the case that the control actions $u[k]$ generated by the algorithm are not non-decreasing (that is, $u[k+1] \succeq u[k]$ may not hold), but the causalisation procedure discussed in Subsection III-B implies that this is not a problem, since the causalised control action $u'[k] = \bigoplus_{i=0}^{k} u[i]$ can be used instead.

D. Computational complexity

Disregarding the complexity of computing the generator matrix N, the two most critical parts of Algorithm 8.1 are the computation of the spectral radius of $f_k(w)$ at Step 1 and the computation of a solution to Equation (24) at Step 2.

The first problem, in Step 1, can be solved by computing the value of the associated mean-payoff game, for which there are pseudo-polynomial algorithms (see [30]). Indeed, let $MPG(s,t,U,V)$ be the complexity of computing the values of the mean-payoff game associated to $Uy \preceq Vy$, $U, V \in \mathbb{Q}_{\max}^{s\times t}$, a complexity for which the exact expression can be found in [30]. Then at each time in Step 1 it is necessary to solve a mean-payoff game associated with $G[k]w \preceq H[k]w$, with $G[k], H[k] \in \mathbb{Q}_{\max}^{2n\times(n+m+p)}$ as presented in Equation (23) (remember that $A \in \mathbb{Q}_{\max}^{n\times n}$, $B \in \mathbb{Q}_{\max}^{n\times m}$ and $N \in \mathbb{Q}_{\max}^{n\times p}$). This implies the complexity $MPG(2n, n+m+p, G[k], H[k])$ at each step, which is pseudo-polynomial.

The second problem, in Step 2, of solving Equation (24) can also be solved in pseudo-polynomial time using, for example, the algorithm presented in [14]. In the same way, if $SMA(s,t,U,V)$ is the complexity of solving the equation $Uy \preceq Vy$, which is pseudo-polynomial, then the complexity of Step 2, solving $\delta G[k]w \preceq H[k]$, is $SMA(2n, n+m+p, \delta G[k], H[k])$ which is also pseudo-polynomial.

Hence, at each k, the complexity of computing the control input $u[k]$ is pseudo-polynomial in Algorithm 8.1.

E. Lyapunov stability

The main concern about Algorithm 8.1 is the convergence for the desired set $\operatorname{Im} N$. It turns out that, if N is an (A,B) max-plus geometrical invariant, at least the Lyapunov stability can be guaranteed. This is because the error, measured using the Chebyshev distance between the actual state and the desired set, does not increase.

In order to prove this, it is necessary to show a preliminary proposition.

Proposition 9: (*Inequality in the Chebyshev distance*) Let $\operatorname{Im} N$ be an (A,B) max-plus geometrical invariant set. Then, for all $x \in \mathbb{Q}_{\max}^{n}$ there exists $u \in \mathbb{Q}_{\max}^{m}$ such that

$$d(Ax \oplus Bu, \operatorname{Im} N) \leq d(x, \operatorname{Im} N). \qquad (25)$$

■

Proof: Let $N \in \mathbb{Q}_{\max}^{n\times k}$. According to Property 1, for a given pair (x,u), $d(Ax \oplus Bu, \operatorname{Im} N)$ is the smallest ξ of the set of all pairs $\{\xi, v\}$ (with $v \in \mathbb{Q}_{\max}^{k}$), denoted by $\mathcal{L}(\boldsymbol{S}, N, x, u)$, such that

$$\xi(Ax \oplus Bu) \succeq Nv;$$
$$\xi Nv \succeq (Ax \oplus Bu).$$

The proof relies in showing that, for all x, there exist suitable parameters u, z, dependent on x, such that the pair $\{d(x, \operatorname{Im} N), z\} \in \mathcal{L}(\boldsymbol{S}, N, x, u)$. This implies directly Equation (25), since the left member of Equation (25) is the smallest ξ in $\mathcal{L}(S, N, x, u)$.

For simplicity, define $\zeta \equiv d(x, \operatorname{Im} N)$. Thus, also according to Property 1, there exists a w, dependent on x, such that

$$\zeta x \succeq Nw; \qquad (26)$$
$$\zeta Nw \succeq x. \qquad (27)$$

The proof starts with Equation (26). Since $\operatorname{Im} N$ is an (A,B) max-plus geometrical invariant semimodule, there exists a matrix $U \in \mathbb{Q}_{\max}^{m\times k}$ such that $\operatorname{Im}(AN \oplus BU) \subseteq \operatorname{Im} N$ and hence $AN \oplus BU = NP$ for a matrix P. Pre-multiplying Equation (26) by A and summing up BUw in both sides, one obtains

$$\zeta Ax \oplus BUw \succeq (AN \oplus BU)w = Nz. \qquad (28)$$

In the last step, one uses the fact that $AN \oplus BU = NP$, and thus $z = Pw \in \mathbb{Q}_{\max}^{k}$ is suitable. Note that z depends on x, since w does. Now, clearly $\zeta \geq 0$ because it is a Chebyshev distance, which is always non-negative. Therefore $\zeta Ax \oplus \zeta BUw = \zeta(Ax \oplus BUw) \succeq \zeta Ax \oplus BUw \succeq Nz$ and thus

$$\zeta(Ax \oplus BUw) \succeq Nz. \qquad (29)$$

Now, it is necessary to work with Equation (27). Again, one proceeds by pre-multiplying by A, but now summing up ζBUw in both sides, obtaining

$$\zeta ANw \oplus \zeta BUw = \zeta Nz \succeq Ax \oplus \zeta BUw \qquad (30)$$

again using the fact that $\mathrm{Im}\,(AN \oplus BU) \subseteq \mathrm{Im}\,N$, with the same $z = Pw$ as in Equation (28). Now, again it is necessary to use the fact that $\zeta \geq 0$. Then $Ax \oplus \zeta BUw \succeq Ax \oplus BUw$. Therefore, using this together with Equation (30)

$$\zeta Nz \succeq Ax \oplus BUw. \tag{31}$$

With $u = Uw$, which depends on x, since w does, Equations (29) and (31) prove the claim that $\{\zeta, z\} \in \mathcal{L}(S, N, x, u)$. This concludes the proof.

■

The proof of the Lyapunov stability of Algorithm 8.1 proceeds then as a corollary of this proposition.

Corollary 3: of Proposition 9: (Lyapunov Stability) If $\mathrm{Im}\,N$ is an (A, B) max-plus geometrical invariant set, the use of the control input $u[k+1]$ designed using Algorithm 8.1 induces a closed-loop dynamical system which is Lyapunov stable on the Chebyshev distance, for all initial conditions.

Proof: The proof uses a Lyapunov-like approach, considering the positive semidefinite function $V(x[k]) \equiv d(x[k], \mathrm{Im}\,N)$ as a measure of distance between the current state and the desired set. Now, it is always possible to (at least) maintain the distance between one step to another, since the $u[k+1]$ generated by Algorithm 8.1 is at least not worse than the particular choice $u' = Uw$ (with U and $w = w(x[k])$ as in Proposition 9), which, according to Proposition 9, guarantees that the distance $V(x[k+1])$ is less than or equal to the actual one $V(x[k])$.

■

However, it is important to remark that, as mentioned in Section I, thus far there is no formal guarantee that the input generated by Algorithm 8.1 will guide the system to the desired set, which would configure asymptotic stability.

The next section will present an illustrative example showing the performance of the Algorithm 8.1 in a wide number of situations, as well as comparisons with Algorithm 6.1 and its improvement, Algorithm 7.1.

IX. Illustrative Example

A. The Problem

The following example was inspired from the TEG model present in Chapter 8 of [18]. It is of moderate complexity and thus can illustrate well the proposed methodology.

Consider the sub-system of the Dutch railway system composed of ten train stations: Amsterdam Central Station (Asd), Groningen (Gr), Zwolle (Zl), Hertogenbosch (Ht), Utrecht Central Station (Ut), Enschede (Es), Nijmegen (Nm), Amersfoot (Amf), Deventer (Dv) and Arnhem (Ah). Four train lines serve these stations, as shown in Figure 1, with the line written in each arc. For instance, line 1 has the route Asd ⇒ Amf ⇒ Zl ⇒ Gr ⇒ Zl ⇒ Amf ⇒ Asd.

The TEG which models this problem can be seen in Figure 2 (see [18] for the details). Each transition represents a *departure* for a given route. For instance, $x_1[k]$, relative to the 1^{st} transition, is the date of departure of the k^{th} train to the route Asd ⇒ Amf, which has the sum between the dwell time and run time equal to 34 minutes. $x_6[k]$, for instance, is the opposite run Amf ⇒ Asd, which takes 36 minutes. This shows that the timings are not, generally, symmetric between two given stations. Each token represents a group of passengers *in route*, but not generally a physical train. For instance, from the transition associated to x_1 emerges two group of passengers, all of them going on the same train: one that will pick the route Amf ⇒ Zl at the current line 1 (the token in the place pointing at the transition associated to x_2) and the other group that will pick a new line, 3, to the route Amf ⇒ Dv (the token in the place pointing at the transition associated to x_{16}). The initial marking is chosen in a way that it is possible to have an arrival of train at each station from hour to hour.

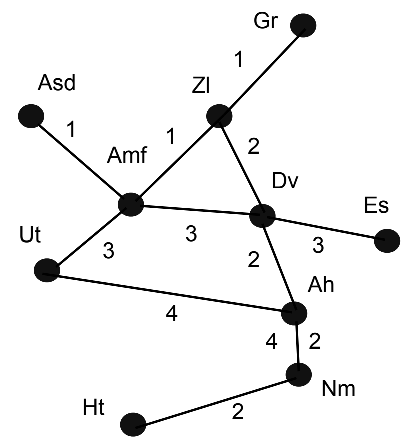

Fig. 1. A sub-system of the Dutch railway with ten stations.

Suppose that it is possible to delay all departures, that is, all the 24 transitions are directly controllable. This means the introduction of 24 new independent control inputs and the addition, in each transition of the TEG in Figure 2, of a new triple of arc, place and token. See Figure 3 for the example of this addition applied only in the transitions of line 4, because drawing the resulting TEG for the whole system would be too cumbersome.

It is possible to write the recursive state space equations for the firing dates x of this TEG, in the form of Equation (3). Note that, as there are two tokens in the place between the transitions associated to x_3 and x_4, it is necessary to add at least a new state variable, namely, one representing the delayed $x_3[k-1]$. However, due to the kind of constraint that will be required further, it is necessary to consider one delay for all the variables. Thus, the augmented state space has 48 variables. Further, it is important to stress that, while all the transitions are *directly* controllable, not all states of $x[k]$ are: the ones respective to the delayed firings, $x_{i+24}[k] = x_i[k-1]$ for $1 \leq i \leq 24$, are not.

Suppose that it is desirable that the trains arrive at each station from hour to hour with at most 2 minutes of delay. Note that, as mentioned previously, the initial marking of the TEG permits that this hourly schedule is possible, but not necessarily ensure it. This can be written as $58 \leq x_i[k] - x_i[k-1] \leq 62$ for $1 \leq i \leq 24$ or equivalently $58 \leq x_i[k] - x_{i+24}[k] \leq 62$ for $1 \leq i \leq 24$.

Further, suppose that it is necessary to induce a connection

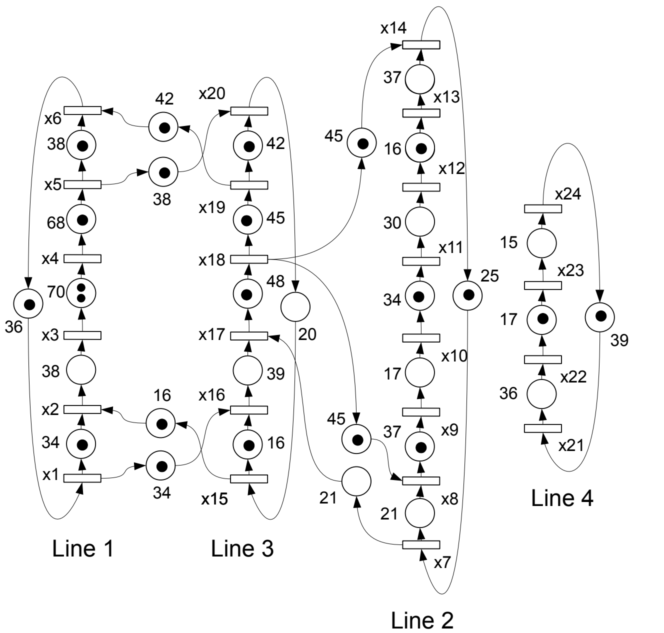

Fig. 2. A TEG representing the sub-system of the Dutch railway system. The units are described in minutes.

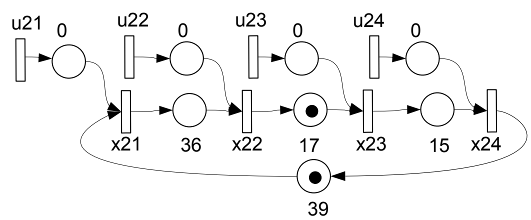

Fig. 3. Line 4 with all the transitions directly controllable.

between line Line 3 and 4 to passengers arriving from Amf in Ut desiring to go to Ah. To this, the date of *departure* to the route Ut $\Rightarrow$ Ah cannot be 3 minutes more than the date of *arrival* of the train from the route Amf $\Rightarrow$ Ut. The date of the k^{th} departure to the route Ut $\Rightarrow$ Ah is $x_{21}[k]$. The date of the k^{th} departure to the route Amf $\Rightarrow$ Ut is $x_{20}[k]$. The date of arrival of the k^{th} train in Ut is 15 minutes after its k^{th} departure from Amf. Note that the total time, as seen in Figure 2, is the sum of the run time - 15 minutes - plus the dwell - 5 minutes. Thus, there must exist two k_1, k_2 such that $|x_{21}[k_1] - 15x_{20}[k_2]| \leq 3$. This means that the passenger arriving from the k_1^{th} from Amf will pick the k_2^{th} train to Ah. In order to deal with this constraint, one assumes that $k_1 = k_2$. Thus $|x_{21}[k]-15x_{20}[k]| \leq 3 \iff -3 \leq x_{21}[k]-15x_{20}[k] \leq 3$. In order to ensure that the problem is coupled, an innocuous set of constraints $-250 \leq x_i[k]-x_j[k] \leq 250$ for all i, j, will be posed to the system , see the discussion in Section III. Also, see Figure 2 to check that the constraint is, indeed, innocuous.

Note that the resulting constraint set $\mathcal{X}_{\texttt{cons}}$ is max-plus multiplicatively invariant. These constraints, in special, can be written in the form $Ex = Dx$. Indeed, all of them can be written as $x_i[k] \geq Mx_j[k]$ for an i, j and M. All of these can be written matricially as $x[k] \succeq Qx[k]$ for a matrix Q or, equivalently, $x[k] = Q^*x[k]$. The matrix Q has no $\perp$ entries and hence the resulting problem is coupled. See the discussion in Section III. Therefore, $E = I$ and $D = Q^*$ can be chosen which results in 48 constraints ($E, D \in \mathbb{Q}_{\max}^{48\times 48}$) in the system.

Note also that one could try to use the approach for generating timetables proposed in [18] and mentioned in Subsection VI-B. However, it turns out that $B \neq I$ and hence the methodology cannot be applied.

Since A, B, E, D has only integers or $\perp$ entries, the algorithm described in Subsection IV-B can be used. The spectral function $s(\lambda)$ can be found to be the composition of three pieces

$$s(\lambda) = \begin{cases} \lambda - 58 \text{ for } \rho(A) = 54.25 \leq \lambda < 58; \\ 0 \text{ for } 58 \leq \lambda \leq 62; \\ \frac{62-\lambda}{3} \text{ for } \lambda > 62; \end{cases}$$

which implies that any $\lambda \in [58, 62]$ can be used. Using $\lambda = 60$ it is possible to solve Equation (13) and then obtain the pair $\{q, w\}$.

B. Comparison of the approaches

Now, three approaches will be compared: Algorithm 6.1 (Periodic Synchronizer - PS, with h such that Equation (17) holds), Algorithm 6.1 with the improvement proposed in Algorithm 7.1 (Feedback Accelerator - FA) and Algorithm 8.1 (Chebyshev Optimized Feedback - CO). For this purpose, 200 random initial conditions $x_{\texttt{ic}}$ were generated, and each one of the three algorithms was applied with these initial conditions. In the Chebyshev Optimized Feedback $N = q$, with q as in Equation (10), was used.

The convergence number of the system is $\kappa(\lambda^{-1}A) = 12$. With the acceleration proposed in Algorithm 7.1 this number is reduced to $\kappa(\lambda^{-1}A_{\texttt{acc}}) = 4$. Two indexes will be used for comparison. The first one is the convergency step $k_{\texttt{cons}}$, which is the smallest step in which convergence is achieved. The second one is the mean elapsed time, $T_{\texttt{mean}}$ (in minutes), as discussed in Subsection VII-B, which is the mean (in the entries) of the elapsed time vector $T \equiv x[k_{\texttt{cons}}] - x[0]$ (as defined in Subsection VII-B), which measures the time between the first firing and the earliest firing in which convergence is achieved, averaged in all entries. To each one of those indexes, the mean value and standard deviation is computed. For Algorithm 8.1, the mean and standard deviation of the computation time, $t_{\texttt{comp}}$ (in seconds), needed to compute the spectral radius is also computed. The experiments were done in an Intel Core I5 with 2.50 GHz and 4GB of RAM, coded in ScicosLab 4.4.1 . The results are shown in Table I.

The computation of the value of the mean payoff game were done using the algorithm proposed in [14]. For the solution of max-plus linear equations, the iterative procedure given by Equation (9) was used.

From Table I, the following can be concluded:

1) The Feedback Accelerator successfully improved not only the convergence number of the system, but also greatly reduced the elapsed time;

TABLE I
MEAN (μ) AND STANDARD DEVIATION (σ) OF DIFFERENT INDEXES OVER 200 SAMPLES

	PS	FA	CO
$\mu(k_{\text{cons}})$	12.94	4	3
$\sigma(k_{\text{cons}})$	0.75	0	0
$\mu(T_{\text{mean}})$	739.51	229.28	169.28
$\sigma(T_{\text{mean}})$	23.73	13.24	13.12
$\mu(t_{\text{comp}})$	X	X	0.0271
$\sigma(t_{\text{comp}})$	X	X	0.0091

2) In the Periodic Synchronizer, of the 200 samples only 12 did not converge exactly in $13 = \kappa(\lambda^{-1}A)+1$ steps, which is the bound in Proposition 6. The other ones converged in 12 steps. This substantiate the claim in Subsection I-B that the proposed bound is not conservative. Further, all the experiments for the Feedback Accelerator converged in less than this bound ($\kappa(\lambda^{-1}A_{\text{acc}})+1 = 5$), but still very close to it;
3) Computing the spectral radius was done extremely quickly, which corroborates with the claim that the Chebyshev Optimization approach can be used efficiently even in relatively large problems.

Finally, for the sake of illustration, Figure 4 shows the simulation of the system, with the Periodic Synchronizer without feedback acceleration (blue) and the Chebyshev-Optimized Feedback (red), under perturbations. The initial condition to both simulations is the same. The perturbation is done at each step with 40% of chance, in which independent and uniform distributed integer delays between 0 and 4 minutes are added do each entry of the unperturbed $x[k]$. The Periodic Synchronizer with Feedback Accelerator was not tested because in this case $\rho(A_{\text{acc}}) = \lambda$, and thus a rescheduling of the parameter h would be necessary. This discussion will be made in a future work.

By Figure 4, it is possible to conclude that the Periodic Synchronizer - even with its open loop nature - is capable of rejecting perturbations. However, it was again outperformed by the Chebyshev-Optimized Feedback, which operates in closed-loop, because it provides a smaller error overall.

X. CONCLUSION

This paper proposes two algorithms (and an improvement for one of them) for solving, in steady state, a specific control problem in the context of TEGs. As mentioned in Section I, by the author's knowledge no previous published work handles the exact form of the proposed problem. Simulations show that the approaches are capable of handling a problem of moderate complexity in a very reasonable amount of time.

The major direction for a future work is proving asymptotic stability for Algorithm 8.1. The authors would like to formalize this idea in a future work. Another important step is how to consider perturbations in the Periodic Synchronizer in the case $\lambda = \rho(A)$, in which a rescheduling of h, depending on the current state, may be necessary. The authors also believe the methodology can be modified and applied to the problem of tracking of signals. There is also the fundamental discussion of how to properly implement those controllers in practice when state feedback is needed, since the controllers are in the event domain but must be implemented in the time domain. In this case, important questions of causality arises. The authors are also interested in developing more applications of the Chebyshev distance optimization approach for other problems in control of TEGs. Finally, the authors believe, and are currently working, that there is still room for generalizations of the proposed algorithms on many different perspectives.

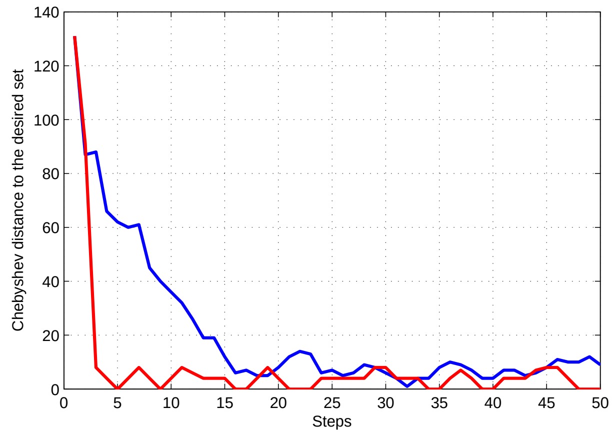

Fig. 4. Performance of the Periodic Synchronizer (blue) and Chebyshev-Optimized Feedback (red) under perturbations. The distance to the desired set at each step is computed as $\|x[k] - (q \backslash x[k])q\|_\infty$.

XI. ACKNOWLEDGEMENTS

The authors are grateful to CAPES, CNPq and FAPEMIG for the financial support. Also, they are very grateful to the anonymous reviewers who provided many corrections, suggestions and insights.

REFERENCES

[1] M. Akian, S. Gaubert, and A. E. Guterman. Tropical polyhedra are equivalent to mean payoff games. *International Journal of Algebra and Computation*, 22(1), 2012.

[2] X. Allamigeon, S. Gaubert, and E. Goubault. The tropical double description method. In *27th International Symposium on Theoretical Aspects of Computer Science (STACS 2010)*, volume 5, pages 47–58, 2010.

[3] S. Amari, I. Demongodin, and J. Loiseau. Control of linear min-plus systems under temporal constraints. *44th IEEE Conference on Decision and Control and European Control Conference*, 2005.

[4] S. Amari, I. Demongodin, J. Loiseau, J. Jean Jacques, and C. Martinez. Max-plus control design for temporal constraints meeting in timed event graphs. *IEEE Transactions on Automatic Control*, 57(2):462 – 467, 2012.

[5] S. Amari, I. Demongodin, and J. J. Loiseau. Sizing and cycle time of an industrial plant using dioid algebra. *Chapter 6 in Supply Chain Optimization, A. Dolgui, J. Soldek and O. Zaikin (Eds.), Series Applied Optimization, Springer-Verlag,*, pages 71–85, 2004.

[6] R. Attia, S. Amari, and C. Martinez. Control of time-constrained dual-armed cluster tools using (max, +) algebra. In *2010 Conference on Control and Fault Tolerant Systems*, pages 412–417, 2010.

[7] A. M. Atto, C. Martinez, and S. Amari. Control of discrete event systems with respect to strict duration: Supervision of an industrial manufacturing plant. *Computers and Industrial Engineering*, 61(4):1149–1159, 2011.

[8] F. Baccelli, G. Cohen, G. J. Olsder, and J.-P. Quadrat. *Synchronization and Linearity*. Wiley, New York, 1992.

[9] C. A. Brackley, D. Broomhead, M. C. Romano, and M. Thiel. A max-plus model of ribosome dynamics during mRNA translation. *Journal of Theoretical Biology*, 2011.

[10] P. Butkovic and G. Hegedus. An elimination method for finding all solutions of the system of linear equations over an extremal algebra. *Ekonom. mat. Obzor*, 20(2):203215, 1984.

[11] P. Butkovic and K. Zimmermann. A strongly polynomial algorithm for solving two-sided linear systems in max-algebra. *Theoretical Computer Science.*, 154(3):437–446, 2006.

[12] B. Cottenceau, L. Hardouin, J.-L. Boimond, and J.-L. Ferrier. Synthesis of greatest linear feedback for timed-event graphs in dioid. *IEEE Transactions on Automatic Control*, 44(6):1258 – 1262, 1999.

[13] R. A. Cuninghame-Green and P. Butkovic. The equation $A \otimes x = B \otimes y$ over (max; +). *Theoretical Computer Science*, 293(1):3 – 12, 2003.

[14] V. Dhingra and S. Gaubert. How to solve large scale deterministic games with mean payoff by policy iteration. *In Proceedings of the 1st international conference on Performance evaluation methodologies and tools*, 2006.

[15] S. Gaubert and S. Sergeev. The level set method for the two-sided eigenproblem. *Discrete Event Dynamic Systems*, 23(2):105–134, 2013.

[16] V. M. Gonçalves, A. Maia, and L. Hardouin. On the solution of max-plus linear equations with application on the control of timed event graphs. *Workshop on Discrete Event Systems (WODES2012)*, pages 91–97, 2012.

[17] J. Gunawardena and M. Keane. On the existence of cycle times for some non-expansive maps. Hewlett-Packard Technical Report HLP-BRIMS-95-03, 1995.

[18] B. Heidergott, G. J. Olsder, and J. van der Woude. *Max Plus At Work*. Princeton University Press, 2006.

[19] S. Gaubert J. Cochet-Terrasson and J. Gunawardena. A constructive fixed-point theorem for minmax functions. *Dynamics and Stability of Systems*, page 407433, 1999.

[20] R. D. Katz. Max-plus (A; B)-invariant spaces and control of timed discrete-event systems. *IEEE Transactions on Automatic Control*, 52(2):229 – 241, 2007.

[21] E. Lorenzo and M. J. de la Puente. An algorithm to describe the solution set of any tropical linear system $Ax = Bx$. *Linear Algebra and its Applications*, 435(4):884–901, 2011.

[22] M. D. Loreto, S. Gaubert, R. D. Katz, and J.-J. Loiseau. Duality between invariant spaces for max-plus linear discrete event systems. *SIAM J. Control Optim*, 48:5606–5628, 2010.

[23] C. A. Maia, C. R. Andrade, and L. Hardouin. On the control of maxplus linear system subject to state restriction. *Automatica*, 47(5):988–992, 2011.

[24] C. A. Maia, L. Hardouin, R. Santos-Mendes, and J. J. Loiseau. A super-eigenvector approach to control constrained max-plus linear systems. *IEEE CDC-ECC, Orlando, Florida*, 2011.

[25] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. The MIT Press, 1994.

[26] W. F. Robert. Algorithm 97: Shortest path. *Communications of the ACM*, 5(6):345, 1962.

[27] B. De Schutter and B. De Moor. A method to find all solutions of a system of multivariate polynomial equalities and inequalities in the max algebra. *Discrete Event Dynamic Systems*, 6(2), 1996.

[28] W. Stephen. A theorem on boolean matrices. *Journal of the ACM*, 9(1):11–12, 1962.

[29] L. Truffet. A decomposition formula of idempotent polyhedral cones based on idempotent superharmonic spaces. *Contributions to algebra and geometry*, 51(2):313–336, 2010.

[30] U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158:343–359, 1996.