Optimal Control Synthesis for Max-Plus Linear Dynamical Systems

Performing the Open-Loop and Feedback Control Policies in Just-in-Time Context

Guilherme Gomes da Silva
Advisor: Prof. Dr. Carlos Andrey Maia

Universidade Federal de Minas Gerais
Programa de Pós-Graduação em Engenharia Elétrica

Optimal Control Synthesis for Max-Plus Linear Dynamical Systems

Performing the Open-Loop and Feedback Control Policies in Just-in-Time Context

Guilherme Gomes da Silva

Thesis submitted to the Programa de Pós-Graduação em Engenharia Elétrica (PPGEE) from Universidade Federal de Minas Gerais (UFMG), as a partial condition for obtaining the PhD Degree in Electrical Engineering.

Thesis Evaluation Committee:
Prof. Dr. Carlos Andrey Maia - Advisor (DEE (UFMG))
Prof. Dr. João Carlos dos Santos Basílio (COPPE (UFRJ))
Prof. Dr. Valter Júnior de Souza Leite (Mecatrônica (CEFET-MG))
Prof. Dr. Ricardo Hiroshi Caldeira Takahashi (DMAT (UFMG))
Prof. Dr. Leonardo Antônio Borges Tôrres (DELT (UFMG))
Prof. Dr. Vinícius Mariano Gonçalves (DEE (UFMG))

Belo Horizonte - Brazil
October, 2016

"Optimal Control Synthesis for Max-plus Linear Dynamical Systems: Performing the Open-loop and Feedback Control Policies in Just-in-time Context"

Guilherme Gomes da Silva

Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Doutor em Engenharia Elétrica.

Aprovada em 27 de outubro de 2016.

Por:

Prof. Dr. Carlos Andrey Maia
DEE (UFMG) - Orientador

Prof. Dr. João Carlos dos Santos Basílio
COPPE (UFRJ)

Prof. Dr. Valter Júnior de Souza Leite
Mecatrônica (CEFET -MG)

Prof. Dr. Ricardo Hiroshi Caldeira Takahashi
DMAT (UFMG)

Prof. Dr. Leonardo Antônio Borges Tôrres
DELT (UFMG)

Prof. Dr. Vinícius Mariano Gonçalves
DEE (UFMG)

À memória dos meus sempre amados pais:
Maria Raquel Silva e Silva e José Reinaldo da Silva
****
To memory of my always beloved parents:
Maria Raquel Silva e Silva and José Reinaldo da Silva

Agradecimentos

Primeiramente agradeço à Deus pela força, proteção e bençãos em cada dia de minha vida.

Agradeço à minha esposa Juliana, pelo amor, paciência e apoio ao longo de toda essa jornada. Ao meu filho, Luís Miguel, que no término da jornada, com seu jeito de ser e olhar, chegou trazendo mais força e incentivo para conclusão deste trabalho.

Aos meus queridos pais, pelo amor, ensinamentos e proteção enviados, com toda certeza, de algum lugar bem próximo à Deus. Minha família (não citarei nomes devido ao grande número de pessoas, mas todos foram muito importantes), a qual foi fundamental para esta conquista, sendo sem dúvidas a base fundamental de minha vida. Agradeço à Tia Regina pela guarda e proteção, sempre com palavras de incentivo em momentos oportunos. Aos meus avós, exemplos de vida e superação, na casa dos quais, por grande parte de minha vida, sempre tive lar digno e acolhedor. À minha irmã Fernanda, obrigado por tudo.

Agradeço especialmente ao meu orientador, Prof. Dr. Carlos Andrey Maia, pela orientação e ensinamentos que vão além do conteúdo desta tese. Agradeço à confiança desprendida desde o início dos trabalhos ainda no mestrado. Serei sempre grato.

Aos membros do Programa de Pós-Graduação em Engenharia Elétrica da UFMG, técnicos e docentes, em especial ao coordenador do programa Prof. Dr. Rodney Rezende Saldanha, que sempre me ajudaram em várias esferas.

À CAPES e a UFMG pelo apoio financeiro.

Agradeço também à todos os amigos da Universidade Federal de São João Del Rei, em especial aos professores Cláudio Alexandre Pinto Tavares, Edgar

Campos Furtado, Leonardo Adolpho Rodrigues da Silva e Cássia Regina Santos Nunes Almeida pelo apoio incondicional ao longo desses anos.

Agradeço aos meus amigos do movimento escoteiro, Tarcisio, Rainer, Marlon, Marcos, Artur (em memória) e Ronaldo. Amigos que escolhi para irmãos. Agradeço também ao grande amigo Geraldo Antero.

À aqueles que não citei mas que de algum modo fizeram parte desta conquista. A importânica de todos será sempre lembrada.

Finalmente, sou grato à todas as pessoas que de alguma forma fizeram parte desta importante conquista em minha vida.

Mais uma vez, Obrigado à Todos.

Acknowledgment

Firstly I thank God for the strength, protection and blessing in every day of my life.

I'd like to thank my wife Juliana for her love, patience and support throughout this journey. My son, Luís Miguel, who at the end of the journey with his way of being and gentle eyes, came into the world bringing me the strength and encouragement to fulfill complete this task.

I'd like to thank my dear parents for their love, teaching and protection.They sent me from somewhere very close to God. The whole family (I will not write any names due to the great number of people, but everyone was really important), which was crucial for this achievement. Thanks to Aunt Regina for her custody, protection, always with encouraging words at appropriate times. To my grandparents, my role model, in whose house, for a long time of my life, I always had a dignified and welcoming home. I'd like to thank my sister Fernanda, for all she has done to me.

Special thanks to my advisor, Prof. Dr. Carlos Andrey Maia, for the guidance and teachings that go far beyond the content of this thesis. I thank you for believing in me from the start of this project, which is being developed since the master degree course. I will always be grateful.

To the members of Programa de Pós-Graduação em Engenharia Elétrica (PPGEE) from Universidade Federal de Minas Gerais (UFMG), technicians and professors, especially the program coordinator Prof. Dr. Rodney Rezende Saldanha, who always helped me in various spheres.

To CAPES and UFMG for financial support.

I also thank all the friends from Universidade Federal de São João del Rei, especially the professors Claudio Alexandre Pinto Tavares, Edgar Campos Furtado, Leonardo Adolpho Rodrigues da Silva and Cassia Regina Santos Nunes Almeida for the unconditional support over the years.

I thank my friends from Boy Scouting, Tarcisio, Rainer, Marlon, Marcos, Artur (in memory) and Ronaldo. Friends whom chose as brothers. I also thank my great friend Geraldo Antero.

I thank all those ones that I haven't mention by name but somehow were part of this achievement.

Again, Thanks to all.

Resumo

Sistemas Dinâmicos Max-Plus Lineares são sistemas modelados por Grafos de Eventos Temporizados (GET) cuja dinâmica pode ser descrita pela álgebra max-plus. Esta tese trata de políticas de controle aplicadas à Sistemas Dinâmicos Max-Plus Lineares. Uma nova formulação multi-objetivo para problemas de controle é proposta, tal formulação é baseada em problemas de otimização e, por meio desta, é possivel considerar restrições não convexas (na álgebra convencional) no problema. Duas políticas de controle são obtidas a partir do problema geral. A primeira política de controle é o Controle "Just-in-Time" em malha aberta, que pode ser desenvolvida tanto em horizonte finito quanto em horizonte infinito visando a economia de recursos e o controle ótimo. As condições necessárias e suficientes para a solução dos problemas são apresentadas, bem como a discussão sobre a complexidade computacional dos métodos propostos. Visando solucionar os problemas de controle, alguns conceitos da álgebra max-plus, como espaços (A,B)-invariantes, Teoria da Residuação e a Teoria dos Semimódulos, são utilizados. Devido à complexidade computacional do método geral de solução, propriedades algébricas são utilizadas para solucionar uma classe importante de problemas de interesse prático. A segunda política de controle é o Controle por Realimentação de Estados no contexto "Just-in-Time". As condições para a existência de uma matriz de realimentação são apresentadas e, se esta matriz existe, um meio para encontrar a maior matriz de realimentação é proposto, a fim de atender à um calendário de demanda para a saída do sistema. Ao final do desenvolvimento de cada política de controle, exemplos numéricos são apresentados para ilustratar as metodologias propostas e a importância dos sistemas tratados

neste trabalho.

Palavras-chave: Sistemas a Eventos Discretos, Álgebra Max-Plus, Sistemas Dinâmicos Max-Plus Lineares, Controle "Just-in-Time", Controle por Realimentação de Estados.

Abstract

Max-Plus Linear Dynamical Systems are systems modeled by Timed Event Graphs (TEG) whose dynamic can be described by Max-Plus Algebra. This thesis deals with control policies applied to the Max-Plus Linear Dynamical Systems. A new multi-objective formulation to control this class of systems is proposed. This formulation is based on optimization problems and it is possible to consider non-convex constraints (in conventional algebra) in the formulation. Two control policies are obtained from the general problem. The first one is the open-loop Just-in-Time Control, which can be developed either in finite horizon or in infinite horizon aiming to saving resources and the optimal control. The necessary and sufficient conditions to solve the problems are presented, as well as the discussion about the computational complexity of the proposed methods in order to solve them. Some concepts on max-plus algebra are used, such as (A,B)-invariant sets, Residuation Theory and the Theory of Semimodules. Due to computational complexity of general method of solution, algebraic properties are used to solve an important class of problems of practical interest. The second control policy is the Feedback control in Just-in-Time context. The conditions for the existence of a feedback matrix are presented. It is also presented a way to find the greatest feedback matrix in order to comply with deadline dates for the system output. At the end of each control problem, numerical examples are developed to illustrate the applicability of the proposed methodologies and the relevance of systems here addressed.

Keywords: Discrete Event Systems, Max-Plus Algebra, Max-Plus Linear Dy-

namic Systems, Just-in-Time Control, Feedback Control.

Contents

List of Symbols

Symbol	**Description**
$i.e.$	That is ("id est")
$e.g.$	For example ("exempli gratia")
$\mathbb{Z}$	Relative integer numbers
$\mathbb{Z}_{max}$	Max-Plus Algebra (Dioid)
$\succeq$	Greater or equal than (natural order)
$\preceq$	Less or equal than (natural order)
$\oplus$	Max-Plus addition
$\otimes$	Max-Plus multiplication
$+$	Traditional sum
$-$	Traditional subtraction
A^T	Transpose of A
I	Max-Plus Identity Matrix
A^n	n^{th} power of A in Max-Plus Algebra
$\backslash\!\!\circ$	Left residuation
$/\!\!\circ$	Right residuation
$\mathsf{Im}A$	Image of matrix A
$\wedge$	Minimum operator (Ex: $2 \wedge 5 = 2$)
$\bigwedge$	Matrix greatest lower bound (Ex: $[1\ 5] \bigwedge [3\ 2] = [1\ 2]$)
$\top$	Top (largest) element in Max-Plus Algebra
ε	Zero element in Max-Plus Algebra
e	Identity element in Max-Plus Algebra
$\lvert\rho\rvert_l$	Length of path ρ
$\lvert\rho\rvert_w$	Weight of path ρ

Chapter 1

Introduction

1.1 Brief Contextualization of the Thesis

Technological advances have increasingly required new techniques for the synthesis and control of complex systems. Some of these systems have man-made operational rules and the rules are related to events, observable or not observable. Events are such as the beginning of a machine operation, a resource enters a system and a server starts the customer service. The events are discrete by system definition, *i.e.*, the events do not have a time duration. This class of systems is classified as Discrete Event Systems (DES).

The behavior of a DES cannot be described by the classical theory of systems, which is based on differential and difference equations. However, there are some tools to deal with DES, for example, Petri Nets, Automata, Markov Process and Dioid Algebra (Cassandras and Lafortune, 2008). This thesis deals with Petri nets, more precisely a subclass of Petri Nets, called Timed Event Graphs (TEG), in which the dynamic behavior can be described by dioid algebra. The dioid algebra used in this work is known as Max-Plus Algebra (also called in some works as Tropical Algebra), because the algebra uses the maximization and addition operations from conventional algebra. So, the systems modeled by a TEG and described by max-plus algebra are called Max-Plus Linear Dynamic Systems (MPLS).

The nonlinear systems based on events cannot use the classical theory of control because the classical theory deals with continuous systems in time. The main importance of max-plus algebra is the fact it describes complex nonlinear systems. That is done in a linear way by using space state equations.

In order to illustrate the statements described in the previous paragraph, consider the queuing system illustrated in Figure 1.1. There are different systems that can be viewed as queuing system, for example, production systems, communications and computer systems, transportation systems, garage systems, airports, and so forth.

The elements of this system are the queue, the server and the customers. The term customer can refer to people, tasks, trucks, pieces, patients, airplanes, e-mail, cases, orders, and so on. The term server might refer to something able to do a service like a receptionist, a machine, a medical personnel, an attendant, a CPU in a computer, any resource that provides service. The queue is the place where the customers must wait by the service because, normally, the capacity of the server is bound, and, in some systems, there is a limit to the number of customers that may be in the queue.

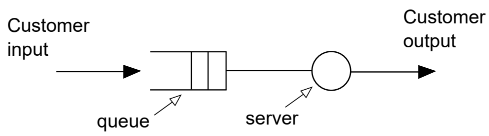


Figure 1.1: Queuing System

It is possible to consider three events driving the queuing system:

- event a: the customer enters the system.
- event s: the service starts.
- event c: the service is completed and the customer leaves the system.

The system can be modeled by Timed Event Graphs (TEG), as showed in Figure 1.2, representing the events a, s and c as transitions (bars associated with a, s and c). The queue and the server are represented by places (circles) Q and B, respectively. The place I represents when the server is idle or busy, this condition is shown by the token (black circle in place), *i.e.*, when the server is idle there is a token in place I.

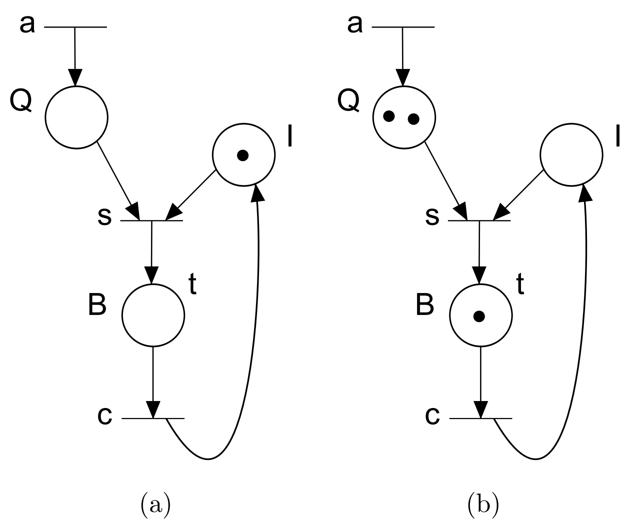


Figure 1.2: Timed Event Graphs. (a) Queuing System A. (b) Queuing System B.

In Figure 1.2a no tokens are placed in Q and B, indicating that there are no customers in the queue and the server is idle, respectively. In Figure 1.2b there are two tokens in place Q, indicating that there are two customers in queue, and one token in B, indicating that the server is busy.

The event a is spontaneous and requires no condition to happen, occurring at instant t_a. On the other hand, the event s depends on two conditions to happen: the presence of customers in the queue and the server being idle, *i.e.*, the event s can happen at the instant $t_s = max(t_a, t_{c^-})$, in which the instant t_{c^-} represents the date when the server is idle by the output of a previous customer of system. Lastly, the transition c requires that the server to be busy and a time t of service, so the event c will happen at date $t_c = t_s + t$. So a customer leaves the system at date $t_c = \max(t_a,t_{c^-}) + t_s + t$.

Considering that the system can serve a lot of customers, although just one at a time, the system can be represented, in a general way, by the TEG of Figure 1.3.

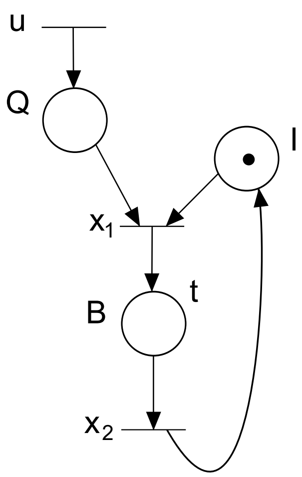


Figure 1.3: Queuing System modeled as Timed Event Graph

The same notation will be used to denominate a transition of a TEG and the variable associated with it. In Figure 1.3, the transition u is the input transition to which the arriving instants (dates) of customers in the system will be associated. The x_1 and x_2 are the internal transitions, also called state transitions. The transition x_2 can also be called output transition y, that represents the date when a customer leaves the system. In this way, considering the previous statements, the dynamic behavior of the system is described by the following equations:

$$\begin{aligned} x_1(k) &= max(u(k), x_2(k-1)), \\ x_2(k) &= t + x_1(k), \\ y(k) &= x_2(k), \end{aligned}$$

being that the integer variable k numerates the transitions firing dates, for example, $x_1(k)$ indicates the instant (date) in which the k^{th} customer is accepted by the server.

The system dynamic behavior in this example could be completely described using the operator maximization (max) and operator addition ($+$). The operator max is related to the synchronization phenomena and the operator $+$ to the processing time of the process. Then, the behavior of a TEG can be completely described by Max-Plus Algebra, in which the operator max is represented by the symbol $\oplus$ and the operator $+$ is represented by the symbol $\otimes$. Therefore, the previous equations can be rewritten as:

$$\begin{aligned} x_1(k) &= x_2(k-1) \oplus u(k) \\ x_2(k) &= t \otimes x_1(k) \\ y(k) &= x_2(k) \end{aligned}$$

This queuing systems can be used to compose a complex queuing net and, consequently, it is possible to get equations in max-plus algebra to describe the system behavior.

Based on the previous equations, the max-plus algebra is very relevant and it simplifies the representation of several complex systems because, as previously mentioned, it is able to write non-linear equations (endowed with operator *max* and + in conventional algebra) in a linear way.

Besides queuing systems, several systems can be classified as MPLS and use the max-plus algebra to describe the temporal dynamic, as examples, it is possible to mention the manufacturing systems; logistics, transportation and distribution systems; chemical systems; communication and computational systems; military and health applications, and so on (Cassandras and Lafortune, 2008)(Banks et al., 2005)(Katz, 2007). Therefore, tools and theories used for modeling and controlling these complex systems, in a simple way, are very valuable given the importance of systems.

Efficient ways to model and control the mentioned systems are useful to generate the optimization of a chosen criterion. Namely, in a manufacturing system it is possible to optimize the stock of resources and save money. In a queuing system it is possible to optimize the number of servers and decrease the time for the customer service. In a transportation system it is possible to optimize the number of vehicles and do not make passengers waiting time go beyond a certain limit.

1.1.1 Thesis Justification

Based on the context previously described, this thesis presents results related to Max-Plus Linear Dynamic Systems. The main results are useful to synthesize optimal controllers in order to make the system respect some constraints. The main objective of the controller proposed in this work is making the system evolve in accordance with the Just-in-Time (JIT) policy, *i.e.*, finding the maximum system input dates in order to comply with deadline dates aiming to develop the minimum cost policy for the inventory. The inventory can be time, money, pieces, and so fourth.

This policy is a management strategy in which the production rate is decided by the demand requires. The main advantage is controlling the inventory costs while still serving customers demand. The control must satisfy some initial condition, state variables are subject to some constraints and the control is optimal to the chosen criterion (Houssin et al., 2007).

The objective of JIT control is to find the maximum input dates from a given date $k = k'$ so that the output dates respect a desirable viable trajectory, *i.e.*, the output dates are less than or equal to the desirable output dates.

This control policy can be applied in a finite horizon or in an infinite horizon. The difference between finite and infinite horizon is determinate by the desirable trajectory for output dates. When the desirable trajectory is bound, the JIT control is applied in finite horizon, otherwise, if the desirable trajectory is not bound, the control problem is applied in infinite horizon. It is possible to understand the control problem in finite horizon as a control problem in infinite horizon but, in this case, the horizon must be large enough (greater than the transient interval of a system).

Initially in this thesis, a general control problem formulation is proposed. The formulation is developed as a multiobjective optimization problem . Two control policies are derived from this general control problem: The Open-Loop Just-in-Time Control and the Feedback Just-in-Time Control policies.

It is important to remark that the control policy choice depends on practical interest and the system features. The distinction between an open-loop control system and a feedback control is important and fundamental. In the open-loop control, the inputs are fixed and independent of output effects (variables of system), the variables of the system do not have any *a posteriori* influence in control action. On the other hand, the feedback control uses any available information about the system behavior to adjust the control input.

From the classical control theory, in closed-loop control policy, the feedback makes the system output relatively more insensible to external disturbances and the internal parameter variations of the system, in comparison with open-loop control policy. In this way, it is possible to use some inaccurate components in order to get a precise control of the system. However, considering stability, open-loop control policies are better than closed-loop control policies since the closed-loop control policies can cause oscillations in the output of a stable system (If a system is unstable it will remain unstable for feedback control in classical control theory). Therefore, the feedback can make a stable system becomes unstable.

On the other hand, considering DES modeled as a TEG, one of stability concepts is related to the number of tokens in places, *i.e.*, a TEG is stable if the number of tokens in places is bound for any input applied. Therefore, in order to obtain a stable system, the feedback control is better than open-loop policies, although the feedback policy is bound in the sense of satisfing some constraints. The open-loop control policies can guarantee optimal performance for any kind of DES system, but it cannot guarantee stability (for more details see Maia (2003)).

To systems in which the inputs are known in advance and there are no disturbances, the open-loop control systems are more indicated. To systems endowed with unknown parameters and subjected to disturbances the closed-loop control policy is more indicated.

From the general control problem, this thesis presents initially an approach to the open-loop JIT control in finite horizon which is useful to solve some problems. A second approach of open-loop JIT control in infinite horizon is also presented.

In order to solve these problems, two issues can be exposed. The first issue is the time computational complexity. As the horizon grows, the time complexity can grow double exponential with the horizon using some methodologies. The second issue is the computational memory. The computational memory to find the solution can be impracticable for some applications depending on the size of the horizon.

In order to deal with the issues previously mentioned, algebraic properties to solve an important class of problems of practical interest are studied. The necessary and sufficient conditions to solve these classes of problems are presented. Thanks to these properties, it is possible to find the solution to problems of practical interest.

The second control policy obtained from the general control problem is the Feedback Control Policy in Just-in-Time context. In this case, the problem objective is to find the greatest feedback matrix so that the system will evolve in accordance with a desired trajectory. In some cases, the feedback matrix found can be non causal, *i.e.*, there are entries in the feedback matrix less than zero, but a causal feedback matrix can be found from the non causal feedback matrix.

The control problem formulation presented in this work is original, however, other works have being developing the JIT control and the feedback control. Concerning the JIT control in infinite horizon, the approach presented in Menguy et al. (2000) does not allow to take general constraints into the system dynamic, but to consider past values of the input dates. Houssin et al. (2007) also deal with a constrained Just-in-Time control in infinite horizon, the authors presented a sufficient condition to find the optimal solution based on an iterative approach using transfer function and the Residuation Theory. On the other hand, finite horizon control is present in De Schutter and van den Boom (2001), in which a state space formulation based on daters is presented, but general constraints can be taken into account by conventional algebra (non general convex constraints). Since conventional algebra was used, the optimization problem is mono-objective and formulated using a complex nonlinear model. The paper does not deal with the algebraic properties of the problem, like the necessary and sufficient conditions for the existence of solutions. A simple formulation of a complex multi-objective optimization problem in finite horizon is presented in Gomes da Silva and Maia (2014), general constraints on the inputs are presented in a convex way using the max-plus algebra (non-convex in conventional algebra) and two methods to solve the problem was developed. The first one is based on the Theory of Semimodule and the optimal solution is given deterministically by an equation, the second one is based on the Modified Alternating Algorithm, originally presented by Cuninghame-Green and Butkovic (2003), and the optimal solution, if it exists, is a fixed point of that algorithm.

Another control policy of interest is the Feedback Control. Regarding constrained feedback control problem, several results were obtained for some class of problems. The paper Maia et al. (2011b) aims to find a feedback controller that ensures the system evolution in accordance with semimodule constraints. The methodology to achieve the goal is based on the super-eigenvector of a matrix. The paper Amari and Isabel Demongodin (2012) develops the constrained feedback controller and the solution is addressed looking for the constrained state equations. The supervisor feedback controllers are calculated and classified according to their performance, and there is no guarantee for the optimal feedback control. In Maia et al. (2011a) the feedback controller is calculated using an equation that involves the system, the feedback and the constraint matrices. The sufficient conditions to calculate a causal feedback matrix, using the Alternating Algorithm Cuninghame-Green and Butkovic (2003), are presented.

Houssin et al. (2013) deals with feedback control using dioid series (idempotent semiring) $\mathcal{M}_{in}^{ax}[\![\gamma,\sigma]\!]$ in an infinite horizon, however, the control objective is the opposite of Just-in-Time policy, *i.e.*, the transitions will fire as soon as possible. All works mentioned about feedback control aim to find the smallest causal feedback matrix.

Based on the previously literature review, this thesis proposes a new general formulation that covers two important policies of control largely studied in literature of control systems. It is based on optimization problems because the work aims to find the optimal control, with multiple objectives and some constraints, including non-convex constraints in conventional algebra. These constraints will be applied to the control problem using the semimodule equation. Recall that the semimodule (in max-plus algebra) is similar to the notion of linear space state in conventional algebra. The main importance of semimodules is the fact that all solutions to equations like $Dx = Ex$ belong to a space finitely generated characterized by an image of a matrix (Butkovic and Hegedus, 1984)(Gondran and Minoux, 2010).

The two policies are important in control theory and largely used in many practical applications. In this sense, the formulation deals with the direct realization for the problem, *i.e.*, dioid series are not applied because the applicability of direct realization is simpler and easier to manipulate in practice. Unlike some previous papers on the subject, this thesis presents the discussion about the necessary and sufficient conditions to find a solution to the problems.

Lastly, the author hopes that the results published in this thesis will be useful to increase the applicability and the interest in the DES theory.

1.2 Publications

The publications related to this PhD thesis are:

- Gomes da Silva, G. and Maia, C. A. (2012). *Controle "just-in-time" em horizonte finito de sistemas max-plus lineares.* In Congresso Brasileiro de Automatica (CBA2012). Campina Grande, Paraiba - Brazil.

- Gomes da Silva, G. and Maia, C. A. (2014). *On Just-in-Time Control of Timed Event Graphs with input constraints: a semimodule approach.* In Discrete Event Dynamic Systems Journal, (DOI: 10.1007/s10626-014-0200-z).

- Gomes da Silva, G. and Maia, C. A. (2014). *Controle "Just-in-Time" em Horizonte Infinito de Grafos de Eventos Temporizados com Restrições.* In

Congresso Brasileiro de Automatica (CBA2014). Belo Horizonte, Minas Gerais - Brazil.

- Gomes da Silva, G. and Maia, C. A. (2015). *Controle "Just-in-Time" Aplicado à um Sistema de Transporte Urbano Max-Plus Linear*. In Simpósio Brasileiro de Automação Inteligente (SBAI2015). Natal, Rio Grande do Norte - Brazil.

- Gomes da Silva, G. and Maia, C. A. (2015). *A Multiobjective Formulation for Just-in-Time Control of Constrained Max-Plus Linear Systems in Infinite Horizon*. In Conference on Decision and Control (CDC2015). Osaka, Japan.

- Gomes da Silva, G. and Maia, C. A. (2016). *Multi-objective Optimization of Max-Plus Linear Systems in Infinite Horizon: Performing the Open-Loop and Feedback Control Policies*. **Submitted for Publication**.

1.3 Organization

The thesis is organized as follows:

- Chapter 1 is the introduction.

- Chapter 2 presents preliminary concepts useful for the comprehension of the thesis, for example, the Residuation Theory, the Theory of Semimodules and the Modified Alternating Algorithm. These concepts were obtained from Baccelli et al. (1992), Cassandras and Lafortune (2008) and Gondran and Minoux (2010).

- Chapter 3 presents the contributions of this thesis. To the general control problem formulation, the open-loop JIT control problem, in finite and infinite horizon, and the feedback control problem in JIT context are developed. The solutions to the control problems as well as the necessary and sufficient conditions to solve the problems are present. In this chapter, for each control problem proposed, numerical examples are developed to illustrate the methodology proposed.

- Chapter 4 presents the final discussion about the thesis. Proposals of future works and the conclusions about the contributions of this thesis are presented.

Chapter 2

Preliminary Concepts

The necessary concepts on Discrete Event Systems for the comprehension of this thesis will be presented in this chapter. The objective of the chapter is just introduce the used concepts such as, for example, Timed Event Graphs (TEG), Max-Plus Algebra and Residuation Theory.

2.1 Discrete Event Dynamic Systems

As mentioned in the previous chapter, many man-made systems evolve according to some rules related to observable or not, deterministic or stochastic events. The events should be considered as occurring instantaneously and causing transitions from one state value to another in a system.

An event can be identified as a specific taken action, a spontaneous occurrence dictated by nature or it may be the result of several conditions which are suddenly all met. Examples of events are the beginning and the end of a task in a manufacturing system, the input of a customer in a queuing system and the act of sending a message in a communication system.

Discrete Event Dynamic Systems (DEDS), or simply Discrete Event Systems, are systems in which the state changes by the occurrence of events (in general asynchronous events), the set of reachable states is discrete and the transition between states occurs only in some discrete points in time.

In other words, in Discrete Event Systems the space state is described by a discrete set and the state transitions are only observed at discrete points in time, these state transitions are associated with events.

Definition 2.1.1 (Discrete Event System) *(Cassandras and Lafortune, 2008) A Discrete Event System is a discrete-state event-driven system, that is, its state evolution depends entirely on the occurrence of asynchronous discrete events over time.*

Important complex systems such as manufacturing systems; logistics, transportation and distribution systems; chemical systems; communication and computational systems; military and health systems, are all examples of DEDS. If these systems can be described by max-plus algebra (this algebra will be introduced further), the DEDS can be classified as a Max-Plus Linear Dynamic System.

Example: 2.1.1 (Discrete Event Dynamic System - Warehouse System) *Consider the warehouse system illustrated in Figure 2.1. The system presents one input area, one output area and a robotic manipulator. When a box enters the system, this fact is related to event "a" and the event "d" occurs when a box leaves the system. Therefore, the system evolution will be driven by the occurrence of two events. The system state will be given by the number of boxes in the warehouse and this number can be given by the sequence of occurrence of events "a"* *and "d". For example, considering the warehouse initially empty, if the sequence of events which occurred in the system is:*

$$aaaaadddaaadda$$

the number of boxes in the warehouse will be equal to four.

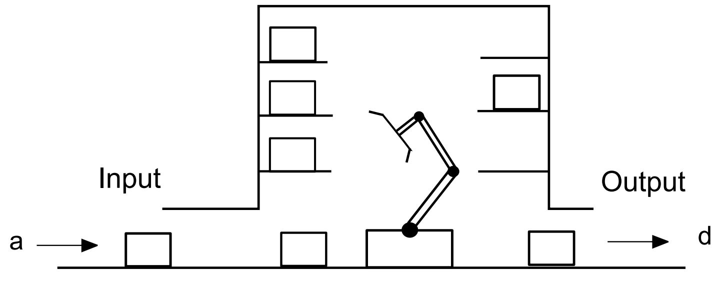


Figure 2.1: Warehouse System

Another example of system evolution is presented in Figure 2.2. By this figure, it is possible to see that five events "a" occur , at dates t_1 to t_5, and in sequence one event "d" occurs at date t_6 and so on.

Therefore, this system is an event-driven system and the set of possible reached states is discrete.

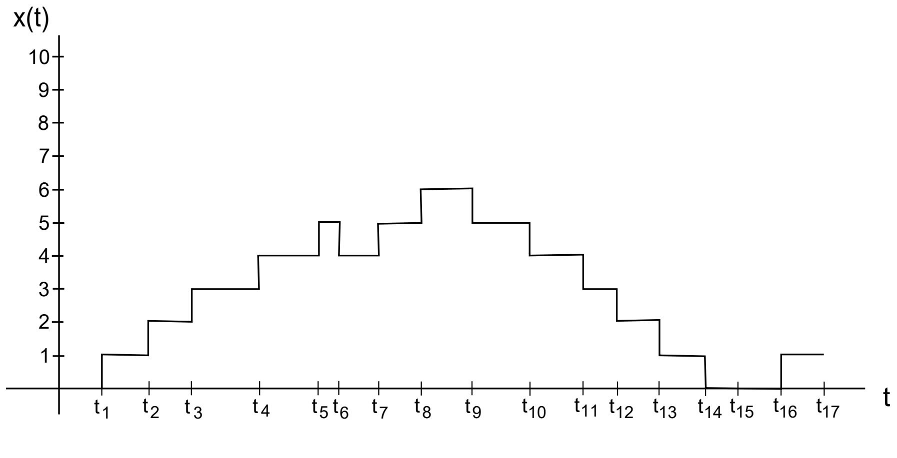


Figure 2.2: Warehouse System State (x(t)) Evolution

There are some tools to deal with DEDS, for example, Graphs, Automata, Petri Nets, Timed Event Graphs, Markov Process and Dioid Algebra. In this thesis, the methodologies will be develop based on Graphs, Petri Nets, Timed Event Graphs and systems described by using Dioid Algebra.

2.2 Graphs

In this section, some necessary graph concepts for the comprehension of this thesis are presented. Concepts such as the definition of a graph and connected graphs are useful to understand the Petri nets and some tools in max-plus algebra. The definitions were obtained in Baccelli et al. (1992). Firstly, the directed graphs are defined.

Definition 2.2.1 (Directed Graph) *A directed graph G is a pair (V, ϵ), being V a set of elements called nodes and ϵ a set of elements which are ordered pairs of nodes, called arcs.*

A directed graph is illustrated in Figure 2.3.

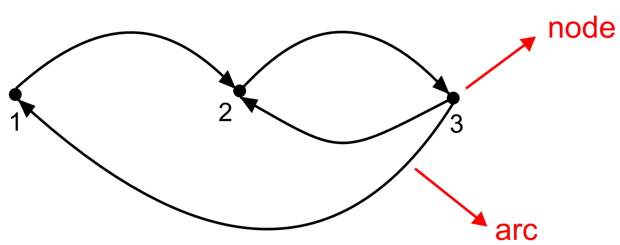


Figure 2.3: Directed Graph

Definition 2.2.2 *If in a graph $(i,j) \in \epsilon$, then node i is called a predecessor of node j and node j is called a successor of node i.*

Definition 2.2.3 (Path, Circuit, Loop, Length) *A* ***path*** ρ *is a sequence of nodes* $i_1, i_2, ..., i_p$, $p > 1$, *so that* i_j *belongs to the set of nodes* $\pi(i_{j+1})$, $j = 1,...,p-1$, *in which the set* $\pi(i_{j+1})$ *indicates the predecessor nodes of node* i_{j+1}. *Node* i_1 *is the initial node and node* i_p *is the final node of the path. In other words, a path is a sequence of arcs which connects a sequence of nodes. When the initial node and the final node are the same, it is called the path as a* ***circuit****, by definition a circuit is defined as a sequence of nodes* $(i_1, i_2, ..., i_p, i_1)$. *A* ***loop*** *is a circuit* (i,i) *composed by a single node which is the initial and the final node. The* ***length*** *of a path or a circuit is equal to the sum of the lengths of the arcs which compose this path. The lengths of the arcs are* 1 *unless otherwise specified. By this definition, the length of a loop is* 1.

Definition 2.2.4 (Subgraphs) *Given a graph* $G = (V, \epsilon)$, *a graph* $G^{'} = (V^{'}, \epsilon^{'})$ *is said to be a subgraph of* G *if* $V^{'} \subset V$ *and if* $\epsilon^{'}$ *consists of a set of arcs of* G *which have their origins and destinations in* $V^{'}$.

Definition 2.2.5 (Connected Graphs) *A graph is called connected when there exists a chain joining* i *and* j *for every pair of nodes* i *and* j. *A chain is a sequence of nodes* $(i_1, i_2, ..., i_p)$ *so that between each pair of successive nodes either the arc* (i_j, i_{j+1}) *or the arc* (i_{j+1}, i_j) *exists. If one disregards the directions of the arcs in the definition of a path, one obtains a chain.*

Definition 2.2.6 (Strongly Connected Graphs) *A graph is called strongly connected when there is at least a path from* i *to* j *for any two different nodes* i *and* j. *According to this definition, a graph which is comprised by an isolated node, with or without a loop, is a strongly connected graph.*

Example: 2.2.1 (Graphs) *Consider the graph in Figure 2.4 to exemplify the* *concepts introduced in this section. It is a directed graph since the arcs are directed. The graph has nine nodes. Node* 2 *is a predecessor node of node* 3*, therefore* $2 \in \pi(3)$*. The sequence of nodes* 1, 2, 3, 7, 2, 4 *is a path. The arc* (9,9) *is a loop. The sequence of nodes* 1, 2, 3, 9, 8, 1 *is a circuit of length equal to* 5*. The graph of Figure 2.4 is connected.*

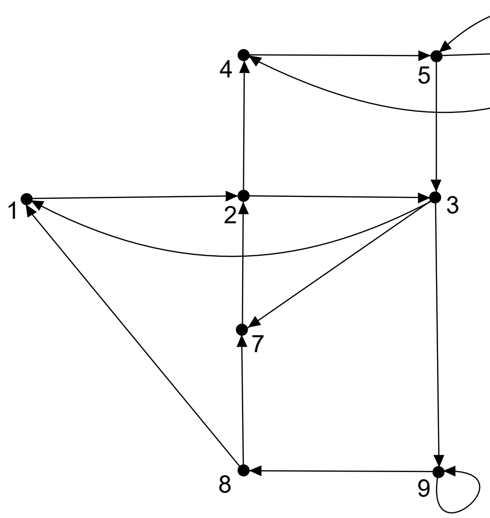


Figure 2.4: Example of Directed Graph

Definition 2.2.7 (Bipartite Graph) *If the set of nodes* V *of a graph* G *can be partitioned into two disjoint subsets* V_1 *and* V_2 *so that, every arc of* G *connects an element of* V_1 *to one of* V_2 *or the other way around, then* G *is called bipartite.*

Definition 2.2.8 (Equivalence Relation $\mathfrak{R}$) *Let* $i,\ j \in V$ *be two nodes of a graph. The equivalence* $i \mathfrak{R} j$ *holds, if either* $i = j$ *or there exist paths from* i *to* j *and from* j *to* i*.*

Definition 2.2.9 (Maximal Strongly Connected Subgraphs) *The subgraphs* $G_i = (V_i, \epsilon_i)$ *corresponding to the equivalence classes determined by* $\mathfrak{R}$ *are the maximal strongly connected subgraphs of* G*.*

Definition 2.2.10 (Cycle Mean) *(Baccelli et al., 1992) The mean weight of a path is defined as the sum of the weights of the individual arcs of this path, divided by the length of this path. If the path is denoted* ρ*, then the mean weight is equal to* $|\rho|_w/|\rho|_l$ *(where* $|\rho|_w$ *is the weight of path* ρ *and* $|\rho|_l$ *is the length of path* ρ*). If such path is a circuit, one talks about the mean weight of circuit, or simple cycle mean.*

Definition 2.2.11 (Maximum Cycle Mean) *The maximum cycle mean is taken over all circuits in the graph, i.e., the maximum over all the cycle mean.*

These definitions are useful to understand and work with Petri nets theory since these nets are directed graphs.

2.3 Petri Nets

Petri nets are directed bipartite graphs, more precisely, a Petri net is a weighted graph endowed with a finite set of places, transitions and arcs. Besides that, there are arc weight functions.

Definition 2.3.1 (Petri Nets) *(Baccelli et al., 1992) A Petri net is a directed bipartite graph* (P, T, A, w)*, being* P *a finite set of places,* T *a finite set of transitions,* A *a finite set of arcs, and* w *are arc weight functions.*

In this thesis, only connected graphs are stated and treated, *i.e.*, there are no isolated places in the graph. Consider the graph in Figure 2.5 to illustrate Petri nets.

The net can be defined by the set $P = \{P_1, P_2\}$, the set $T = \{t_1\}$, the set $A = \{(P_1,t_1),(t_1,P_2)\}$ and by the functions $w(P_1,t_1) = 2$ and $w(t_1,P_2) = 1$.

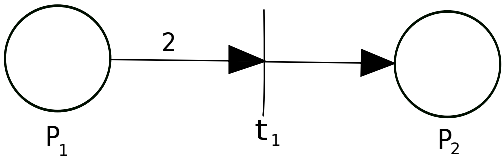


Figure 2.5: Petri Net

Definition 2.3.2 (Input and Output Places) *Considering an arc* (P_i, t_i)*, the place* P_i *is called input place for transition* t_i *and* t_i *is called output transition for place* P_i*, represented by* $I(t_i)$ *and* $O(P_i)$*, respectively. In the same way, considering an arc* (t_j, P_j)*, the place* P_j *is called output place for transition* t_j *and* t_j *is called input place for place* P_j*, represented by* $O(t_j)$ *and* $I(P_j)$*.*

Definition 2.3.3 (Marked Petri Nets) *(Cassandras and Lafortune, 2008) A marked Petri net is a quintuple* (P, T, A, w, x)*, in which* (P, T, A, w) *is a Petri net and* x *is a marking of the set of places. The vector*

$$x = \begin{bmatrix} x(p_1) & x(p_2) & \cdots & x(p_n) \end{bmatrix} \in \mathbb{N}^n$$

is a row vector associated with x*.*

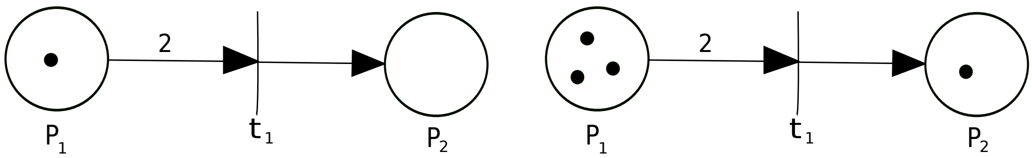
Considering the Petri net in Figure 2.5, there are a lot of possible markings for this net, one of them is the vector

$$x_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

and another vector is

$$x_2 = \begin{bmatrix} 3 & 1 \end{bmatrix}.$$

These markings are represented, respectively, in Figure 2.6a and in Figure 2.6b.



(a) Marked Petri Net 1

(b) Marked Petri Net 2

Figure 2.6: Marked Petri Nets. (a) Marked Petri Net with vector x_1. (b) Marked Petri Net with vector x_2.

The markings in each place (black circles) are called tokens. The way how a net is marked represents the net state because there is only one marking for each state reached by the net. In order to simplify the nomenclature, the marked Petri nets will be called just Petri nets, since all Petri nets are marked, even if the marking is a null marking.

Definition 2.3.4 (Enabled Transition) *(Cassandras and Lafortune, 2008) A transition $t_j \in T$ in a Petri net is called enabled if*

$$x(p_i) \succeq w(p_i,t_j) \text{ for all } p_i \in I(t_j)$$

being $I(t_j)$ the set of input places of transition t_j.

A transition in a Petri net can be enabled to fire and indeed fires. A transition is enabled to fire when the number of tokens in input places of the transition is greater than the weight of the arcs which connect the respective place to the transition. The transition can fire as soon as it is enabled. The transition fires when the event associated with the transition happens. When a transition fires, tokens are removed from input places and tokens are placed in the output places in accordance with the weight of the respective arcs.

Definition 2.3.5 (Dynamic of Petri Net) *(Cassandras and Lafortune, 2008)* *The state transition function* $f : \mathbb{N}^n \times T \rightarrow \mathbb{N}^n$, *of a Petri net* (P,T,A,w,x) *is defined for the transition* $t_j \in T$ *if and only if,*

$$x(p_i) \succeq w(p_i,t_j), \text{ for all } p_i \in I(t_j)$$

being $I(t_j)$ *the set of input places of a transition* t_j*. If* $f(x,t_j)$ *is defined, it is defined* $x^+ = f(x,t_j)$*, in which*

$$x^+(p_i) = x(p_i) - w(p_i,t_j) + w(t_j,p_i), i = 1,...,n. \tag{2.1}$$

For example, in Figure 2.6a, the transition t_1 is not enabled and in Figure 2.6b the transition t_1 is enabled. When the transition t_1 fires, from Figure 2.6b, two tokens are removed from place P_1 and one token is placed in P_2, in accordance with the weights of the arcs. The obtained Petri net when the transition t_1 fires, from marking x_2, is shown in Figure 2.7. Therefore, it possible to conclude that the number of tokens in a net is not conserved for some models.

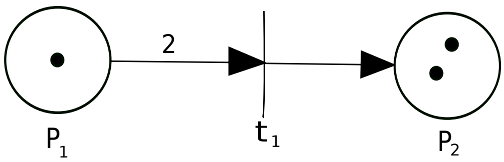


Figure 2.7: Petri net when transition t_1 fire from marking x_2

The Equation 2.1 can be generalized by the following matrix equation:

$$X^+ = X + uA \tag{2.2}$$

in which X is a row vector with entries the number of tokens in each place, u is a vector representing which transition will fire (if a transition will be fired it is represented by number 1 in vector u, otherwise it is represented by 0. Only one transition fires each time) and matrix A is the incidence matrix (the matrix A represents the number of tokens removed and placed in places by the transitions). A transition in a net can fire only if

$$X + uA^{-} \succeq 0 \tag{2.3}$$

in which matrix A^{-} represents the number of tokens taken off by transitions of places. In order to illustrate these equations, consider the following example.

Example: 2.3.1 *Consider the following graph in Figure 2.8. The incidence matrix is given by:*

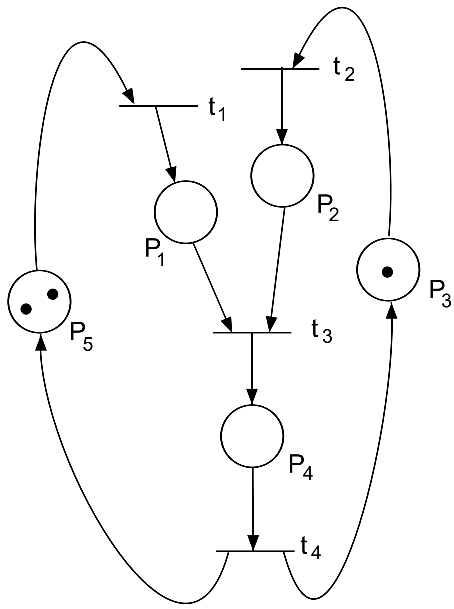


Figure 2.8: Petri Net.

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 & 0 \\ -1 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & -1 & 1 \end{bmatrix}$$

The incidence matrix represents the graph structure and it can be obtained by looking for the weights between places and transitions. For example, consider the entry a_{ij} from matrix A, the variable i is related to transitions and the variable j is related to places. Therefore the entry a_{11} is the weight between the transition t_1 and place P_1 and it is equal to 1 because t_1 puts one token in place P_1. The entry a_{15} is equal to -1 because t_1 removes one token from place P_5, and so on.

The matrix A^- is obtained similarly to A. The matrix A^- is obtained from the number of tokens that each transition removes from each place. In this example, the matrix A^- is given by:

$$A^- = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \end{bmatrix}.$$

The initial marking of the net in this example is:

$$X_0 = \begin{bmatrix} P_1 & P_2 & P_3 & P_4 & P_5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 2 \end{bmatrix}.$$

and it gives the number of tokens initially in each place.

If the transition t_2 will be fired, the vector u can be given by:

$$u = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix},$$

but, it is necessary that

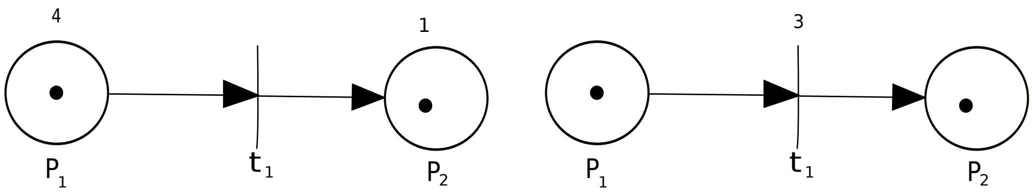
$$X_0 + uA^- \succeq 0.$$

This inequality holds so the transition t_2 is enabled to fire. If t_2 fires the reached state is given by:

$$X_1 = X_0 + uA,$$

$$X_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 2 \end{bmatrix}.$$

The Petri nets are useful to model Discrete Event Systems and the model can use processing time associated with its structure, called holding time. There are some ways to do the timing, being the time associated with places (Net P-timed) or the time associated with transitions (Net T-timed) the most common. The nets P-timed and T-timed are showed in Figure 2.9a and Figure 2.9b, respectively.



(a) P-timed Petri Net

(b) T-timed Petri Net

Figure 2.9: Timed Petri Nets. (a) P-timed Petri Net. (b) T-timed Petri Net.

When the time is associated with transitions, after the transition is enabled, it must wait the associated time and, then, the transition can fire. In cases when the time is associated with places, a token must spend the time in the place before contributing to the enabling of output transitions. In this thesis, the holding time is associated with places.

The Petri nets of interest for this thesis are a subclass called Timed Event Graphs (TEG), these nets are introduced in the next section.

2.3.1 Timed Event Graphs

Definition 2.3.6 (Event Graph) *A Petri net is called an event graph if each place has at most one input transition $I(P_i)$ and at most one output transition $O(P_i)$.*

An event graph is a Petri net in which each place has at most one input transition and at most one output transition. The event graph are able to model discrete event systems endowed with time delay and synchronization phenomena, *i.e.* the event graphs are not able to model systems where there is a competition for resources.

Definition 2.3.7 (Timed Event Graph (TEG)) *(Baccelli et al., 1992) A timed event graph is an event graph in which each place has a holding time associated with it.*

Assumption: 2.3.1 *In this thesis, the time associated with places is assumed non-varying.*

Figure 2.10 shows a TEG with three places (P_1, P_2 and P_3) and three transitions (u_1, x_1 and x_2). In this graph, the transition x_1 can fire by k^{th} date after the k^{th} firing date of transition u_1 and the $(k-1)^{th}$ firing date of transition x_2. The k^{th} firing date of transition x_1 is related to the $(k-1)^{th}$ firing date of transition x_2 by the net initial condition, *i.e.*, there is a token in P_3 enabling the transition

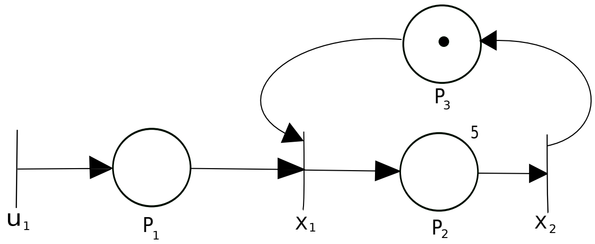


Figure 2.10: Timed-Event Graph.

x_1 and this token was placed in P_3 when the transition x_2 fired at the $(k-1)^{th}$ date. Then, the transition x_1 will be enabled to fire at the k^{th} date after the greatest date between the firing date $u(k)$ and $x_2(k-1)$. The place P_2 has 5 time units associated with it, therefore when a token arrives at this place it must wait, at least, five time units before contributing to enabling transition x_2. It is possible to describe the k^{th} firing date of transition x_1 and x_2 by the following equations:

$$x_1(k) = max\{u_1(k), x_2(k-1)\}, \tag{2.4}$$

$$x_2(k) = x_1(k) + 5. \tag{2.5}$$

The Equations 2.4 and 2.5 use the operator addition and the operator maximization. The operator addition can be related with time delay linked with places. The operator maximization can be related to the synchronization phenomena. With these operators the TEG dynamics can be completely described.

Those equations can be described using a dioid, which is an algebraic structure endowed with all properties of a ring, except the inverse additive element, so the dioids are characterized algebraically as an idempotent semiring (Baccelli et al., 1992). The dioids will be defined in the next section.

2.4 Dioids and Max-Plus Algebra

A ring is defined algebraically as $(\mathcal{A},\oplus,\otimes)$, with the set $\mathcal{A}$ endowed with two internal operators. The operator $\oplus$ (addition) is associative, invertible, commutative and it has the neutral element ε. The operator $\otimes$ (multiplication) is associative, commutative, it admits the neutral element e and, besides that, it is distributive with relation to $\oplus$.

In this way, dioids are algebraic structures characterized as an idempotent semiring since the dioids have two operators and all properties of a ring, except the additive inverse element.

Definition 2.4.1 (Dioids) *(Baccelli et al., 1992) A dioid is defined as a set $\mathcal{D}$ endowed with two internal operators, $\oplus$ (addition) and $\otimes$ (multiplication), obeying the following axioms:*

- *The addition is associative and commutative: $\forall a,b,c \in \mathcal{D}$, $(a \oplus b) \oplus c = a \oplus (b \oplus c) = c \oplus (b \oplus a)$.*
- *The multiplication is associative and distributive on left and on right with relation to addition: $\forall a,b,c \in \mathcal{D}$, $(a \otimes b) \otimes c = a \otimes (b \otimes c)$ and $(a \oplus b) \otimes c = (a \otimes c)(b \otimes c)$.*
- *Existence and absorbing by neutral element of addition (ε): $\forall a \in \mathcal{D}$, $a \oplus \varepsilon = a$ and $a \otimes \varepsilon = \varepsilon$.*
- *Existence of identity element of multiplication (e): $\forall a \in \mathcal{D}$, $a \otimes e = e \otimes a = a$.*
- *Idempotency of addition: $\forall a \in \mathbb{D}$, $a \oplus a = a$.*

A dioid is said to be complete if it is a complete ordered set in accordance with the partial canonical relation $\preceq$ ($a \preceq b \Leftrightarrow b = a \oplus b$ and $c \succeq b$ and $b \succeq c \Leftrightarrow b = c$), and if, besides that, it is closed in relation to infinite sums and the multiplication is distributive with relation to infinite sums.

2.4.1 Lattice Properties of Dioids

The properties and definitions presented in this subsection are used in this thesis and they were obtained in Baccelli et al. (1992).

Order Relation: a binary relation (denoted by $\succeq$) which is reflexive, transitive and anti-symmetric.

Total (Partial) Order: the order is total if for each pair of elements (a,b), the order relation holds true either for (a,b) or for (b,a), or otherwise stated, if a and b are always comparable; otherwise, the order is partial.

Ordered Set: a set endowed with an order relation; it is sometimes useful to represent an ordered set by an undirected graph the nodes of which are the element of the set; two nodes are connected by an arc if the corresponding elements are comparable, the greater one being higher in the diagram; the minimal number of arcs is represented, the other possible comparisons being derived by transitivity.

Top Element (of an ordered set): an element which is greater than any other element of the set.

Bottom Element (of an ordered set): an element which is smaller than any other element of the set.

Maximum Element: an element of the subset which is greater than any other element of the subset; if it exists, it is unique; it coincides with the top element if the subset is equal to the whole set.

Minimum Element: an element of the subset which is smaller than any other element of the subset; if it exists, it is unique; it coincides with the bottom element if the subset is equal to the whole set.

Maximal Element: an element of the subset which is not smaller than any other element of the subset; if a subset has a maximum element, it is the unique maximal element.

Majorant: an element not necessarily belonging to the subset which is greater than any other element of the subset; if a majorant belongs to the subset, it is the maximum element.

Minorant: an element not necessarily belonging to the subset which is smaller than any other element of the subset; if a minorant belongs to the subset, it is the minimum element.

Upper Bound: the least majorant, that is, the minimum element of the subset of majorants.

Lower Bound: the greatest minorant, that is, the maximum element of the subset of minorants.

2.4.2 Max-Plus Algebra

The max-plus algebra is defined as a complete dioid endowed with the structure $(\mathbb{Z} \cup \{-\infty\}, max, +)$, being denoted by $\mathbb{Z}_{max}$.

Definition 2.4.2 (Algebraic Structure of $\mathbb{Z}_{max}$) *The symbol* $\mathbb{Z}_{max}$ *denotes the set* $(\mathbb{Z} \cup \{-\infty\}$ *endowed with the maximization operation and the addition operation represented, respectively, as* $\oplus$ *and* $\otimes$*, and the convention* $(-\infty)+\infty = -\infty$.

The Kleene star operation is another operation defined on any dioid, denoted by the symbol *. This operator is algebraically defined as:

$$a^* = \bigoplus_{i\in\mathbb{N}} a^i = a \otimes a^{i-1},$$

being $a^0 = e$.

In subsection 2.3.1, the dynamic behavior of a TEG could be completely described using only two operators: the addition operator and the maximization operator. Then, the TEG can be completely described, in a linear way, using the max-plus algebra, *i.e.*, it is possible to rewrite Equations 2.4 and 2.5 using the max-plus algebra as:

$$x_1(k) = u_1(k) \oplus x_2(k-1) \tag{2.6}$$

and

$$x_2(k) = 5 \otimes x_1(k). \tag{2.7}$$

In this thesis, in order to simplify the notation, the symbol $\otimes$ will be omitted in equations when convenient, without any information loss. From TEG of Figure 2.10, the output dates will be given by:

$$y(k) = x_2(k). \tag{2.8}$$

By Equations 2.6, 2.7 and 2.8, it is possible to rewrite these equations in matrix notation as state space equations in max-plus algebra:

$$\begin{cases} x(k) = Ax(k-1) \oplus Bu(k) \\ y(k) = Cx(k). \end{cases} \tag{2.9}$$

in which A, B and C are matrix of appropriate dimensions with the characteristics of the system, $x(k)$ the state vector (also called internal) at k^{th} date, $u(k)$ the k^{th} input date and $y(k)$ the k^{th} output date.

2.4.3 Graphs and Matrices

An important tool to deal with TEG described by max-plus algebra are the matrices in $\mathbb{Z}_{max}$. All weighted graphs (graphs in which the arcs are associated with weights) are related with matrices (Baccelli et al., 1992), *i.e.*, every weighted graph has a representative matrix, besides that, every matrix whose entries are integers has a representative graph.

The matrix operations in max-plus algebra are similar to the matrix operations in conventional algebra. Let $A, B \in \bar{\mathbb{Z}}_{max}^{n\times m}$, being n,$m \in \mathbb{N}$, the addition operation is defined as:

$$[A \oplus B] = a_{ij} \oplus b_{ij} \tag{2.10}$$

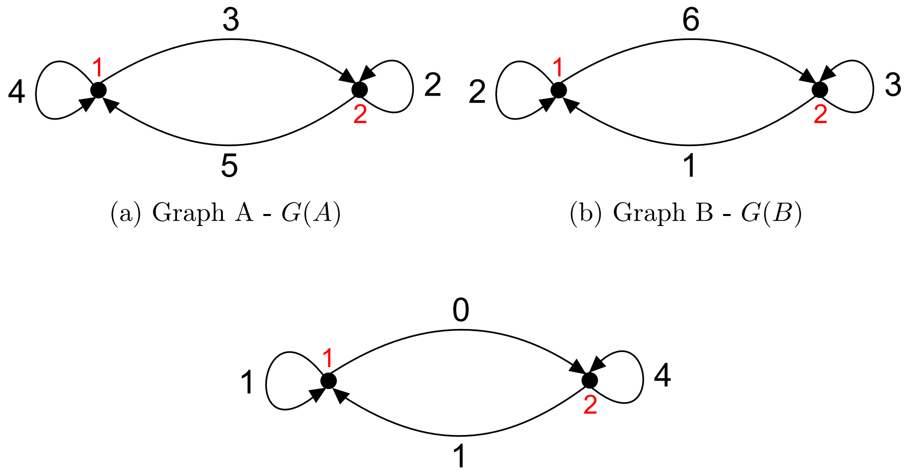
in which i and j are, respectively, the rows and columns of matrices A and B. Let $C \in \bar{\mathbb{Z}}_{max}^{m\times p}$, the matrix multiplication between matrix A and matrix C in $\mathbb{Z}_{max}$ is defined as:

$$[A \otimes C]_{ik} = \bigoplus_{j=1}^{m} a_{ij} \otimes c_{jk} \tag{2.11}$$

in which i and k are, respectively, the row and column indexes of the elements in the resulting matrix.

Example: 2.4.1 (Representation and Operations with Matrices) *Consider the graphs of Figure 2.11. The graph A, graph B and graph C can be related to the matrices A, B and C, respectively given by:*

$$A = \begin{bmatrix} 4 & 3 \\ 5 & 2 \end{bmatrix}, B = \begin{bmatrix} 2 & 6 \\ 1 & 3 \end{bmatrix}$$



(a) Graph A - $G(A)$ (b) Graph B - $G(B)$

(c) Graph C - $G(C)$

Figure 2.11: Weighted Graphs. (a) Graph A. (b) Graph B. (c) Graph C.

and

$$C = \begin{bmatrix} 1 & 0 \\ 1 & 4 \end{bmatrix}.$$

Then

$$A \oplus B = \begin{bmatrix} 4 \oplus 2 & 3 \oplus 6 \\ 5 \oplus 1 & 2 \oplus 3 \end{bmatrix} = \begin{bmatrix} 4 & 6 \\ 5 & 3 \end{bmatrix}$$

and

$$(A \oplus B) \otimes C = \begin{bmatrix} 4 \otimes 1 \oplus 6 \otimes 1 & 4 \otimes 0 \oplus 6 \otimes 4 \\ 5 \otimes 1 \oplus 3 \otimes 1 & 5 \otimes 0 \oplus 3 \otimes 4 \end{bmatrix} = \begin{bmatrix} 5 \oplus 7 & 4 \oplus 10 \\ 6 \oplus 4 & 5 \oplus 7 \end{bmatrix} = \begin{bmatrix} 7 & 10 \\ 6 & 7 \end{bmatrix}.$$

In $\mathbb{Z}_{max}$ the identity matrix is denoted by I with $i_{mn} = e$ for $m = n$ and $i_{mn} = \varepsilon$ for $m \neq n$. Let $A \in \bar{\mathbb{R}}^{m \times m}_{max}$, the Kleene star operator is defined for matrices as:

$$A^* = \bigoplus_{m \in \mathbb{N}} A^m,$$

in which $A^m = A \otimes A^{m-1}$ and $A^0 = I$.

In order to facilitate the representation of the neutral element of addition, it will be denoted by a dot in matrix notation.

Example: 2.4.2 (Timed Event Graph and Max-Plus Algebra) *In order to illustrate the linearity features of the dynamic behavior of the systems by max-plus algebra, consider the following timed event graph in Figure 2.12:*

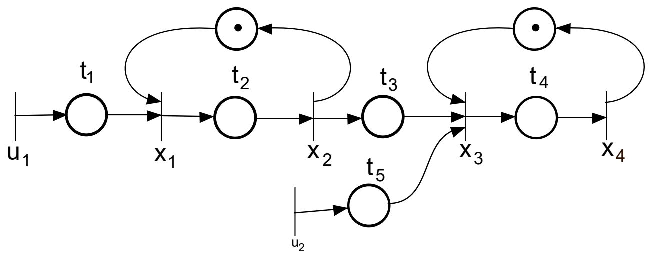


Figure 2.12: Timed Event Graph

The TEG in Figure 2.12 can model a queuing system, a manufacturing system, and so forth. As explained in Chapter 1, the dynamic behavior of a TEG can be described using only the maximization operator (max) and the addition operator (+) in conventional algebra, by the following equations:

$$x_1(k) = max(t_1 + u_1(k), x_2(k-1)) \tag{2.12}$$

$$x_2(k) = t_2 + max(t_1 + u_1(k), x_2(k-1)) \tag{2.13}$$

$$x_3(k) = max(t_3 + t_2 + max(t_1 + u_1, x_2(k-1)), t_5 + u_2(k), x_4(k-1)) \tag{2.14}$$

$$x_4(k) = t_4 + max(t_3 + t_2 + max(t_1 + u_1(k), x_2(k-1)), t_5 + u_2(k), x_4(k-1)) \tag{2.15}$$

The Equations 2.12 to 2.15 are complex nonlinear equations in conventional algebra that can be used to describe the dynamic behavior of the system. In addition, these equations are obscure from the point of view of conventional algebra.

However, the behavior of the system can be described in a simpler way by *using the max-plus algebra, in which the maximization is denoted by* $\oplus$ *and the addition denoted by* $\otimes$ *(the symbol* $\otimes$ *will be omitted by convenience), by the following equations:*

$$x_1(k) = x_2(k-1) \oplus t_1 u_1(k) \tag{2.16}$$

$$x_2(k) = t_2 x_2(k-1) \oplus t_2 t_1 u_1(k) \tag{2.17}$$

$$x_3(k) = t_2 t_3 x_2(k-1) \oplus x_4(k-1) \oplus t_3 t_2 t_1 u_1(k) \oplus t_5 u_2(k) \tag{2.18}$$

$$x_4(k) = t_4 t_3 t_2 x_2(k-1) \oplus t_4 x_4(k-1) \oplus t_4 t_3 t_2 t_1 u_1(k) \oplus t_5 t_4 u_2(k) \tag{2.19}$$

The Equations 2.16 to 2.19 are linear in max-plus algebra and they can be written in matrix notation as:

$$\begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} = \begin{bmatrix} . & e & . & . \\ . & t_2 & . & . \\ . & t_3 t_2 & . & e \\ . & t_4 t_3 t_2 & . & t_4 \end{bmatrix} \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \\ x_3(k-1) \\ x_4(k-1) \end{bmatrix} \oplus \begin{bmatrix} t_1 & . \\ t_2 t_1 & . \\ t_3 t_2 t_1 & t_5 \\ t_4 t_3 t_2 t_1 & t_5 t_4 \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix}$$

in which the dot in matrices is the neutral element of addition ε*.*

Therefore, the max-plus algebra is able to describe the behavior of important complex nonlinear systems in a linear way. Another advantage is that concepts such as eigenvalues, eigenvectors and linear vector space can be inherited from conventional algebra by max-plus algebra. These concepts will be presented further.

2.4.4 Systems of Linear Equations

In this subsection, some systems of linear equations are addressed, mainly in matrix notation. Dealing with max-plus algebra, the general system of equations is

$$Ax \oplus c = Cx \oplus d \tag{2.20}$$

in which A and B are matrices and c and d are vectors of appropriate dimensions.

Definition 2.4.3 (Canonical form of a System of Affine Equations) *(Baccelli et al., 1992) The system $Ax \oplus c = Bx \oplus d$ is said to be in canonical form if A, B, c and d satisfy:*

- $B_{ij} = \varepsilon$ *if* $A_{ij} \succ B_{ij}$, *and* $A_{ij} = \varepsilon$ *if* $A_{ij} \prec B_{ij}$;
- $d_i = \varepsilon$ *if* $b_i \succ d_i$, *and* $b_i = \varepsilon$ *and* $b_i \prec d_i$.

Cuninghame-Green and Butkovic (2003) developed a methodology to find the greatest solution, smaller than the initial condition for Equation 2.20. Therefore, considering the initial condition equal to $\top$ (the greatest element in max-plus algebra), the method finds the greatest solution to that equation. The solution to Equation 2.20 will be better discussed in Subsection 2.7.1.

For instance, there are two classes of linear systems of interest for which there exists a satisfactory theory. The first one is $x = Ax \oplus b$.

Theorem 2.4.1 *(Baccelli et al., 1992) If there are only circuits of non positive weight in a graph $\mathcal{G}(A)$, there is a solution to $x = Ax \oplus b$ which is given by $x = A^*b$. Moreover, if the circuit weight is negative, the solution is unique.*

The $(A^*)_{ij}$ represents the maximum weight of all paths of any length from j to i in a graph. Thus, the necessary and sufficient condition for the existence of $(A^*)_{ij}$ is the non existence of circuits with positive weight.

Theorem 2.4.2 *(Baccelli et al., 1992) If a graph has no circuit with positive weight, then*

$$A^* = e \oplus A \oplus \ldots \oplus A^{n-1} \tag{2.21}$$

where n is the dimension of matrix A.

The second class of linear systems is $Ax = b$. In this case, however, the notion of subsolution of $Ax = b$ must be considered, *i.e.*, the values of x which satisfy $Ax \preceq b$, where the order relation on the vectors is defined by $x \preceq y$ if $x \oplus y = y$.

Theorem 2.4.3 *(Baccelli et al., 1992) Given an $n \times n$ matrix A and an n-vector b in $\mathbb{Z}_{max}$, the greatest solution of $Ax \preceq b$ exists and it is given by*

$$-x_j = \max_i(-b_i + A_{ij}) \tag{2.22}$$

or

$$x_j = \min_i(b_i - A_{ij}) \tag{2.23}$$

The solution to equation $Ax = b$ and the notion of subsolution will be discussed in Section 2.5.

2.4.5 Spectral Theory of Matrices

The main objective is to find the maximum cycle mean, where the maximum cycle is obtained from all circuits in a graph. Considering a graph $G(A)$ related to a $n \times n$ matrix A, the maximum weight of all circuits of length j which pass through node i of G can be written as $(A^j)_{ii}$. The maximum of these weights over all nodes is $\bigoplus_{i=1}^{n}(A^j)_{ii}$, that can be written as the trace of matrix A. Then, the maximum cycle mean (ν) of a graph can be given, in max-plus algebra notation, by:

$$\nu = \bigoplus_{j=1}^{n}(\text{trace}(A^j))^{1/j} \tag{2.24}$$

Definition 2.4.4 *(Baccelli et al., 1992) Let $A \in \mathbb{Z}_{max}$ a square matrix. If there exists a scalar $\lambda \in \mathbb{Z}_{max}$ and a vector $v \in \mathbb{Z}_{max}$ that has at least one finite entry so that*

$$A \otimes v = \lambda \otimes v, \tag{2.25}$$

then λ is called an eigenvalue of A and v an eigenvector associated with eigenvalue λ.

Theorem 2.4.4 *(Baccelli et al., 1992) The necessary and sufficient condition for a square matrix A to be irreducible is the graph $G(A)$ associated with matrix A be strongly connected.*

Theorem 2.4.5 *(Baccelli et al., 1992) If A is irreducible, or equivalently if $G(A)$ is strongly connected, there exists one and only one eigenvalue (but possible several eigenvectors). This eigenvalue is equal to the maximum cycle mean of the graph:*

$$\lambda = \max_{\zeta} \frac{|\zeta|_w}{|\zeta|_l} \tag{2.26}$$

where ζ ranges over the set of circuits of $G(A)$, in which $|\zeta|_w$ is the weight of path ζ and $|\zeta|_l$ is the length of path ζ.

2.4.6 Asymptotic Behavior of A^k

Definition 2.4.5 (Critical Circuits) *(Baccelli et al., 1992) A circuit ζ of the graph $G(A)$ is called critical if it has maximum weight, that is, $|\zeta|_w = e$.*

Definition 2.4.6 (Critical Graph) *(Baccelli et al., 1992) The critical graph $G^c(A)$ consists of those nodes and arcs of $G(A)$ which belong to a critical circuit of $G(A)$. Its nodes constitute the set V^c.*

Example: 2.4.3 Baccelli et al. (1992) *Consider the matrix*

$$A = \begin{bmatrix} e & e & \varepsilon & \varepsilon \\ -1 & -2 & \varepsilon & \varepsilon \\ \varepsilon & -1 & -1 & \varepsilon \\ \varepsilon & \varepsilon & e & e \end{bmatrix}.$$

Its precedence graph $G(A)$ has three critical circuits, namely: the circuit from node 1 to node 1, the circuit from node 3 to node 4 and to node 3 and the circuit from node 4 to node 4.

Its critical graph is the precedence graph of matrix

$$C = \begin{bmatrix} e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & e \end{bmatrix}.$$

Finally, the matrix A has the eigenvector

$$\begin{bmatrix} e & -1 & -2 & -2 \end{bmatrix}^T$$

associated with eigenvalue e.

Definition 2.4.7 *(Baccelli et al., 1992) The cyclicity of a maximal strongly connected subgraph is the greatest common divisor of the lengths of all its circuits. The cyclicity $\varsigma(G)$ of a graph $G(A)$ is the least common multiple of the cyclicities of all its maximal strongly connected subgraphs.*

Definition 2.4.8 *(Baccelli et al., 1992) Let $A \in \mathbb{Z}_{max}$ such that the corresponding graph has at least one circuit. The cyclicity of A, denoted by $\varsigma(A)$, is the cyclicity of the critical graph of A.*

Theorem 2.4.6 *(Baccelli et al., 1992) Let $A \in \mathbb{Z}_{max}$ an irreducible matrix, then $\exists k_0 \in \mathbb{N}$ such that $\forall k \succeq k_0 : A^{k+\varsigma} = \lambda^{\varsigma} \otimes A^k$, in which λ is the eigenvalue of matrix A and ς is the cyclicity of A.*

Definition 2.4.9 *(Baccelli et al., 1992) A matrix A is said to be cyclic if there exist d and M such that $\forall m \succeq M$, $A^{m+d} = A^m$. The least such d is called the cyclicity of matrix A and A is said to be $d-$cyclic.*

Lemma 2.4.1 *(Baccelli et al., 1992) Let $A \in \mathbb{Z}_{max}$ be an irreducible matrix endowed with cyclicity $\varsigma(A)$. Then, the cyclicity of matrix A^{ς} is equal to 1.*

The cyclicity equal to 1 defines a periodic behavior in steady state. Consider the initial state $x(0)$, from the state $x(k)$ at the k^{th} date, the behavior will be periodic at $x(k+\varsigma)$, by the following equation:

$$x(k+\varsigma) = A^{(k+\varsigma)} \otimes x(0) \tag{2.27}$$

that can be rewritten as

$$x(k+\varsigma) = \lambda^{\varsigma} A^k \otimes x(0) \tag{2.28}$$

and considering, thanks to the periodicity, $x(k) = A^k x(0)$,

$$x(k+\varsigma) = \lambda^{\varsigma} \otimes x(k). \tag{2.29}$$

For a graph endowed with cyclicity equal to 1, it is possible to show that

$$x(k+1) = A \otimes x(k) = \lambda x(k). \tag{2.30}$$

Theorem 2.4.7 *(Baccelli et al., 1992) A necessary and sufficient condition to have $lim_{k\to\infty} A^k = Q$ is that the cyclicity of each maximal strongly connected subgraph of $G(A)$ is equal to 1.*

Theorem 2.4.8 *(Baccelli et al., 1992) Suppose that $G(A)$ is strongly connected graph. Then there exists a $k^{'}$ such that*

$$\forall k \succeq k^{'}, A^k = Q, \tag{2.31}$$

if and only if the cyclicity of each maximal strongly connected subgraph of $G(A)$ is equal to 1.

2.4.7 Max-Plus Linear Systems Theory

Firstly, the max-plus linear systems are defined.

Definition 2.4.10 (Max-Plus Linear Dynamic Systems) *The systems modeled by timed event graphs whose dynamics are described by max-plus algebra by state space equations are called Max-Plus Linear Dynamic Systems.*

The Equations in 2.9 are used to describe max-plus linear systems and in these systems the firing dates of transitions are non-decreasing, *i.e.*, $A \succeq I$, since

$x(k) \succeq x(k-1)$.

By using the max-plus algebra to describe max-plus linear systems it is common to find equations such as

$$x(k) = A_0 x(k) \oplus A_1 x(k-1) \oplus B_0 u(k) \tag{2.32}$$

in which A_0, A_1 and B_0 are system matrices, but this equation can be rewritten as:

$$x(k) = Ax(k-1) \oplus Bu(k) \tag{2.33}$$

considering what was presented in Theorem 2.4.1.

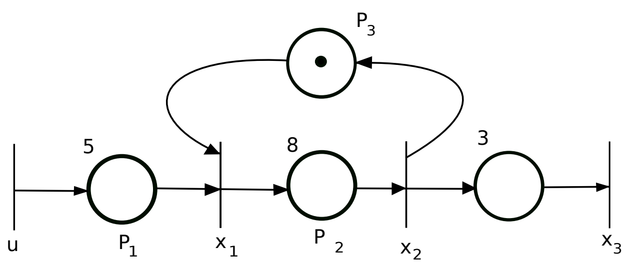


Figure 2.13: Timed Event Graph

This important result can be better understood by considering the TEG in Figure 2.13, that has the firing dates described by an equation as Equation 2.32, which can be rewritten as

$$x(k) = A_0 x(k) \oplus W, \tag{2.34}$$

in which $W = A_1x(k-1) \oplus B_0u(k)$. Suppose that $x(k)$ is a solution, consequently, $x(k)$ must satisfy the Equation 2.34, so

$$x(k) = A_0x(k) \oplus W \quad (2.35)$$

$$x(k) = A_0(A_0x(k) \oplus W) \oplus W \quad (2.36)$$

$$x(k) = A_0^2x(k) \oplus A_0W \oplus W \quad (2.37)$$

$$\vdots$$

$$x(k) = A_0^lx(k) \oplus A_0^{l-1}W \oplus A_0^{l-2}W \oplus \cdots \oplus W \quad (2.38)$$

and then $x(k) \succeq A_0^*W$. Equation 2.38 can be rewritten as $x = Ax \oplus b$. Using Theorem 2.4.1 and considering all graph circuits with non positive weights, the solution to Equation 2.38 is given by:

$$x(k) = A_0^*W, \quad (2.39)$$

since the entries of A_0^l are the maximum weights of circuits with weight l. For l great enough, the entries of A_0^l are weights of the paths of length k. Those paths necessarily traverse some circuits of A_0 a number of times going to ∞ with l. Since the weights of these circuits are all negative, $A_0^l \to [\varepsilon]$ when $l \to \infty$.

Replacing W in Equation 2.39, the equation

$$x(k) = A_0^*A_1x(k-1) \oplus A_0^*B_0u(k) \quad (2.40)$$

is obtained, resulting in

$$x(k) = Ax(k-1) \oplus Bu(k) \quad (2.41)$$

with $A = A_0^* A_1$ and $B = A_0^* B_0$.

Example: 2.4.4 (Max-Plus Linear System) *Consider a system modeled as the TEG in Figure 2.13. The dinamic behavior of the TEG can be described by the following equations in max-plus algebra:*

$$\begin{aligned} x_1(k) &= x_2(k-1) \oplus 5 \otimes u_1(k) \\ x_2(k) &= 8 \otimes x_1(k) \\ x_3(k) &= 3 \otimes x_2(k) \\ y(k) &= x_3(k) \end{aligned}$$

This equations can be rewritten as $x(k) = A_0 x(k) \oplus A_1 x(k-1) \oplus B_0 u(k)$, in matrix notation, as:

$$\begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} = \begin{bmatrix} . & . & . \\ 8 & . & . \\ . & 3 & . \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} \oplus \begin{bmatrix} . & e & . \\ . & . & . \\ . & . & . \end{bmatrix} \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \\ x_3(k-1) \end{bmatrix} \oplus \begin{bmatrix} 5 \\ . \\ . \end{bmatrix} u(k) \tag{2.42}$$

$$y(k) = \begin{bmatrix} . & . & e \end{bmatrix} \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \\ x_3(k-1) \end{bmatrix}. \tag{2.43}$$

Using the previous result, the Equation 2.42 can be rewritten as the Equation 2.41:

$$\begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} = \begin{bmatrix} . & e & . \\ . & 8 & . \\ . & 11 & . \end{bmatrix} \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \\ x_3(k-1) \end{bmatrix} \oplus \begin{bmatrix} 5 \\ 13 \\ 16 \end{bmatrix} u(k). \tag{2.44}$$

2.5 Residuation Theory

As previously mentioned, the max-plus algebra is an idempotent semiring (dioid) which does not have the inverse element for the $\oplus$ operation, therefore the operation $\oplus$ is not particularly invertible for matrix applications such as finding a solution to matrix equations such as $Ax \preceq b$ or $Ax = b$.

Definition 2.5.1 (Isotone Mappings) *(Baccelli et al., 1992) A mapping f defined on a dioid $(\mathcal{D},\otimes,\oplus)$ in a dioid $(\mathcal{C},\otimes,\oplus)$ is called isotone mapping if, for all a,$b \in \mathcal{D}$, the following order relation is preserved:*

$$a \preceq b \Leftrightarrow f(a) \preceq f(b)$$

The Residuation Theory, applied to dioids, deals with the inversion of isotone mappings and with the solutions to equations in partially ordered sets. Let f be the isotone mapping of a dioid $\mathcal{D}$ on a dioid $\mathcal{C}$, if an equation $f(x) = b$ is not surjective, the equation cannot have a solution to some values of b, and if $f(x) = b$ is not injective, the equation has non unique solutions, *i.e.*, equations like $f(x) = b$ can have innumerable or no solutions. The solution to this problem can be obtained considering a subset of solutions, *i.e.*, values to x that satisfy $f(x) \preceq b$. The Residuation Theory is particularly useful to find the maximal sub-solution to the inequality of the form $f(x) \preceq b$. The maximal sub-solution x^{sub} is equal to the maximal value of x so that the $f(x^{sub})$ is smaller than or equal to b. Dually, the Dual Residuation Theory finds the smallest super-solutions to equations such as $f(x) = b$ in dioid algebra. The smallest super-solution x^{sup} is the smallest solution to x such that $f(x^{sup})$ is greater than or equal to b (Maia, 2003) (Baccelli et al., 1992). To ensure the existence of a lower bound and an upper bound, the dioids $\mathcal{D}$ and $\mathcal{C}$ are assumed as complete dioids.

The definitions and theorems presented below were obtained from Baccelli et al. (1992) and Maia (2003) and applications of Residuation Theory on dioids are shown in Baccelli et al. (1992).

Definition 2.5.2 (Residual and Residuated Mapping) *Let $\mathcal{D}$ and $\mathcal{C}$ be partially ordered sets. The isotone mapping $f : \mathcal{D} \mapsto \mathcal{C}$ is a residuated mapping if, for all $y \in \mathcal{C}$, there exists the greatest subsolution for the inequality $f(x) \preceq y$. The mapping $f^{\sharp}$ is called residual of mapping f and the greatest subsolution is denoted by $f^{\sharp}(y)$.*

Theorem 2.5.1 (Residuation) *Let $\mathcal{D}$ and $\mathcal{C}$ be ordered sets. The isotone mapping $f : \mathcal{D} \mapsto \mathcal{C}$ is residuated, if and only if $f^{\sharp}$ is the unique isotone mapping such that*

$$(f \circ f^{\sharp})(y) \preceq y \text{ and } (f^{\sharp} \circ f)(x) \succeq x$$

$\forall x \in \mathcal{D}$ and $\forall y \in \mathcal{C}$.

The residuated mappings to complete dioids are characterized by the following theorem.

Theorem 2.5.2 (Residuation for Complete Dioids) *Consider the complete dioids $\mathcal{D}$ and $\mathcal{C}$, the mapping $f : \mathcal{D} \mapsto \mathcal{C}$ is residuated, if and only if, for all subset*

X of $\mathcal{D}$,

$$
\begin{aligned}
f\left(\bigoplus_{x\in X} x\right) &= \bigoplus_{x\in X} f(x), \\
f(\varepsilon) &= \varepsilon.
\end{aligned}
$$

To dually residuated mappings, analogous statements from residuated mappings can be demonstrated.

Definition 2.5.3 (Dual Residue and Dually Residuated Mapping) *Let $\mathcal{D}$ and $\mathcal{C}$ be ordered sets. The isotone mapping $f : \mathcal{D} \mapsto \mathcal{C}$ is dually residuated, if for all $y \in \mathcal{C}$, there exists the smallest super-solution for the inequality $f(x) \succeq y$. This smallest super-solution is denoted by $f^{\flat}(y)$ and the mapping $f^{\flat}$ is called dual residue of f.*

Theorem 2.5.3 (Dual Residuation) *Let $\mathcal{D}$ and $\mathcal{C}$ be ordered sets. The isotone mapping $f : \mathcal{D} \mapsto \mathcal{C}$ is dually residuated, if and only if, $f^{\flat}$ is the unique isotone mapping such that,*

$$
f \circ f^{\flat}(y) \succeq y \text{ and } f^{\flat} \circ f(x) \preceq x
$$

$\forall x \in \mathcal{D}$ and $\forall y \in \mathcal{C}$.

Theorem 2.5.4 (Dual Residuation for Complete Dioids) *Let $\mathcal{D}$ and $\mathcal{C}$ be complete dioids. The mapping $f : \mathcal{D} \mapsto \mathcal{C}$ is dually residuated, if and only if, for all subsets X of $\mathcal{D}$*

$$
\begin{aligned}
f\left(\bigwedge_{x\in X} x\right) &= \bigwedge_{x\in X} f(x) \\
f(\top) &= \top
\end{aligned}
$$

in which $\bigwedge$ is the lower bound operator and $\top$ is the upper bound element (infinity element in conventional algebra).

Residuated Mappings and Dually Residuated Mappings in Complete Dioids

Let the mappings L_a, R_a and T_a defined on a complete dioid $\mathcal{D}$ as:

$$L_a : x \mapsto a \otimes x \tag{2.45}$$

$$R_a : x \mapsto x \otimes a \tag{2.46}$$

$$T_a : x \mapsto a \oplus x \tag{2.47}$$

The Theorem 2.5.2 ensures that these mappings are residuated. From mappings presented in this section, it is straightforward to see that $R_a(\varepsilon) = L_a(\varepsilon) = \varepsilon$. It is also possible to verify that the multiplication is distributive with respect to infinite sums on the right and on the left for the mappings L_a and R_a since, as mentioned in Section 2.4, a dioid is said to be complete and distributive if it is closed in relation to infinite sums and if the multiplication is distributive over infinite sums. Besides that, if $\mathcal{D}$ is commutative, $L_a = R_a$ and, consequently, $L_a^\sharp = R_a^\sharp$.

Considering a complete and distributive dioid, $T_a(\varepsilon) \neq \varepsilon$, so the mapping T_a is not always residuated. However, $T_a(\top) = \top$, then this mapping is, in accordance to the Theorem 2.5.4, dually residuated.

From the theorems and definitions presented in Section 2.5, the linear mappings $f(x) = a \otimes x$ and $f(x) = x \otimes a$ and the affine mapping $f(x) = x \oplus a$ are, respectively, residuated and dually residuated in any dioid.

The notations presented in the following subsection for the mappings L_a, R_a and T_a are shown in Baccelli et al. (1992).

Notation: 2.5.1 (Residues of L_a, R_a and T_a)

$$\begin{aligned} L_a^\sharp(x) &= a \lr x \\ R_a^\sharp(x) &= x \rl a \\ T_a^\flat(x) &= x \ominus a \end{aligned}$$

in which $\lr$ *is the symbol for left residuation,* $\rl$ *is the symbol for right residuation and* $\ominus$ *is the symbol for dual residuation.*

For the particular case of $\mathbb{Z}_{max}$, it is possible to show that $L_a^\sharp = R_a^\sharp = x - a$ (subtraction operation in conventional algebra). The residue of mapping $T_a^\flat$ implies that:

$$T_a^\flat = x \ominus a = \begin{cases} x, & if \quad x \succ a \\ ;\varepsilon, & if \quad x \preceq a. \end{cases}$$

Generally, if a dioid $\mathcal{D}$ is a complete dioid, $\mathcal{D}^{n\times n}$ is also a complete dioid. Then the operations $L_A(X) = A \otimes X$ and $R_A(X) = X \otimes A$ are also residuated, being A and X matrices with coefficients in $\mathcal{D}$. It is possible to show that the residuals of matrices $L_A^\sharp = A \lr X$ and $R_a^\sharp = X \rl A$ are given by:

$$(L_A^\sharp)_{ij} = (A \lr X)_{ij} = \bigwedge_{l=1}^{n} a_{li} \lr x_{lj} \tag{2.48}$$

$$(R_A^\sharp)_{ij} = (X \rl A)_{ij} = \bigwedge_{l=1}^{n} x_{li} \lr a_{lj} \tag{2.49}$$

in which i,j and n are, respectively, the rows, columns and the dimensions of matrices $R_A^\sharp$ or $L_A^\sharp$. Analogously, the mapping $T_A(X) = X \oplus A$ is dually residuated and the dual residue is given by:

$$(T_A^{\flat})_{ij} = (X \ominus A)_{ij} = \begin{cases} x_{ij}, & if \quad x_{ij} \succ a_{ij}; \\ \varepsilon, & if \quad x_{ij} \preceq a_{ij}. \end{cases} \tag{2.50}$$

Example: 2.5.1 (Residuation and Dual Residuation) *Let the matrices* A,$B \in \bar{\mathbb{R}}^{2\times2}_{max}$, *be given by*

$$A = \begin{bmatrix} 2 & 3 \\ 1 & 0 \end{bmatrix} \ and \ B = \begin{bmatrix} 0 & 4 \\ 1 & 2 \end{bmatrix}.$$

By the Residuation Theory, the greatest solution to X *such that* $A \otimes X \preceq B$ *is given by:*

$$X = A \backslash B = \begin{bmatrix} -2 & 1 \\ -3 & 1 \end{bmatrix}$$

and the greatest solution to $X \otimes A \preceq B$, *is given by:*

$$X = B / A = \begin{bmatrix} 0 & -1 \\ -2 & -2 \end{bmatrix}.$$

By the Dual Residuation, the lowest solution to X *such that* $X \oplus A \succeq B$ *is given by:*

$$X = B \ominus A = \begin{bmatrix} \varepsilon & 4 \\ \varepsilon & 2 \end{bmatrix}.$$

2.6 Theory of Semimodules

In the same way that the operator in dioid algebra can be related to conventional algebra, the concept of semimodule can be related to the classical theory of systems. By definition, the concept of semimodule is equivalent in semirings to the notion of linear state space in classical theory of systems.

In order to introduce the definition of semimodule, first consider the definition of monoids.

Definition 2.6.1 (Monoids) *Let $\mathcal{M}$ be a set, $\hat{\otimes}$ an operation in $\mathcal{M}$ and $\varepsilon_{\mathcal{M}} \in \mathcal{M}$ is the neutral element when $\forall x \in \mathcal{M} : \varepsilon_{\mathcal{M}} \hat{\otimes} x = x \wedge x \hat{\otimes} \varepsilon_{\mathcal{M}} = x$. Then, if there exists a neutral element to the pair $(\mathcal{M}, \hat{\otimes})$, this pair is called a monoid. If the operator $\hat{\otimes}$ is commutative, the monoid is also commutative.*

The definition of monoids is useful to define the semimodules.

Definition 2.6.2 (Semimodules) *(Cohen et al., 2004) A semimodule is defined on a semiring $(\mathcal{D}, \oplus, \otimes, \varepsilon_s, e)$ as a commutative monoid $(\mathcal{M},\hat{\otimes})$, with neutral element $\varepsilon_{\mathcal{M}}$ and equipped with the map $(\mathcal{D} \times \mathcal{M}) \mapsto \mathcal{M}$, that is $(\lambda,v) \mapsto \lambda.v$ (left action), in which:*

1. $(\lambda \otimes \mu).v = \lambda.(\mu.v)$
2. $\lambda(u \hat{\oplus} v) = \lambda u \hat{\oplus} \lambda v$
3. $(\lambda \hat{\oplus} \mu).v = \lambda.v \hat{\oplus} \mu v$
4. $\varepsilon_s.v = \varepsilon_{\mathcal{M}}$
5. $\lambda.\varepsilon_{\mathcal{M}} = \varepsilon_{\mathcal{M}}$
6. $e.v = v$

for all $u,v \in \mathcal{M}$ *and* $\lambda, \mu \in \mathcal{D}$.

Definition 2.6.3 (Subsemimodule) *(Katz, 2007) A subsemimodule of* $\mathcal{M}$ *is a subset* $\mathcal{S} \subset \mathcal{M}$ *for which if* $u,v \in \mathcal{S}$ *and* $\lambda, \mu \in \mathcal{D}$ *then* $\lambda.v \hat{\oplus} \mu.v \in \mathcal{S}$.

In this work, the subsemimodule is considered the $n-$dimensional vector with entries in $\mathcal{D}$ equipped with operators $(u\hat{\oplus}v)_i = u_i \oplus v_i$ and $\lambda.v = \lambda \otimes v$.

In this context, it is possible to show that the set of all solutions to equations like $Ax = Bx$[1], for which A,B and x have entries in $\mathbb{Z}_{max}$, can be characterized as a semimodule finitely generated, *i.e.*, the set of all solutions to this kind of equation can be expressed as an image of a matrix with entries in $\mathbb{Z}_{max}$ (Butkovic and Hegedus, 1984) (Gaubert, 1992) (Maia et al., 2011a).

2.6.1 Finding All Solutions to Equation $Ax = Bx$

The methodology to generate all solutions to the equation $Ax = Bx$ was presented first in Butkovic and Hegedus (1984). Complexity issues are discussed in Allamigeon et al. (2008) and a simplified version of the method, as well as a mathematical proof of its effectiveness, is presented in Maia et al. (2011b). The objective of this subsection is to introduce this methodology.

The interest is in equations based on the dioid $\mathbb{Z}_{max}$. As previously mentioned, all solutions to equation $Ax = Bx$, for which A and $B \in \mathbb{Z}_{max}^{m\times n}$ and $x \in \mathbb{Z}_{max}^{n}$, belong to a finitely generated semimodule given by the columns of a matrix. The semimodule can be computed by the algorithm presented in Butkovic and Hegedus (1984), this algorithm was improved in Allamigeon et al. (2008).

In order to find a solution to equation $Ax = Bx$, it is possible to consider solving the equation row by row of the matrices A and B. The solution to this

[1] Inequalities like $Ax \preceq Bx$ can be easily rewritten as an equation since they are equivalent to $(A \oplus B)x = Bx$, considering $Ax \preceq Bx \Rightarrow Ax \oplus Bx = Bx \Rightarrow (A \oplus B)x = Bx$.

equation can be straightforwardly obtained by solving the first row of the matrices A and B and then using the result to solve the second row of the matrices and so on.

Therefore, the equation to be solved in each step is:

$$a_1 \otimes x_1 \oplus \ldots \oplus a_n \otimes x_n = b_1 \otimes x_1 \oplus \ldots b_n \otimes x_n. \quad (2.51)$$

Hereafter, without loss of generality, these vectors are assumed so that $a_k \oplus b_k \neq \varepsilon$, $\forall k \in \{1, \ldots, n\}$. In this sense, if there exists a non null solution to the problem, then:

$$\exists (i,j) | a_i \otimes x_i = b_j \otimes x_j, \quad (2.52)$$

for which

$$(a_k \otimes x_k \preceq a_i \otimes x_i) \text{ and } (b_k \otimes x_k \preceq x_j), \forall k. \quad (2.53)$$

Since the solution is non null, $\exists k$ such that $x_k \neq \varepsilon$. Then $a_k \otimes x_k \neq \varepsilon$ or $b_k \otimes x_k \neq \varepsilon$. As a result of Inequality 2.53, $a_i \otimes x_i = b_j \otimes x_j \neq \varepsilon$ is ensured. Therefore

$$a_i \otimes b_j \neq \varepsilon, \quad (2.54)$$

and it can be seen $[x_i \ x_j]^T \in \mathsf{Im} \ [b_j \ a_i]^T$, $a_i \succeq b_i$ and $b_j \succeq a_j$. Moreover, it is possible to show that all vectors $v^{(l,p)} \in \mathbb{Z}^n_{max}$, such that $v^{(l,p)}(l) = b_p$, $v^{(l,p)}(p) = a_l$ and $v^{(l,p)}(k) = \varepsilon$ for $k \notin \{l,p\}$, for which $a_l \succeq b_l$ and $b_p \succeq a_p$, generate a solution to the Equation 2.51. Based on this result, the following set is defined:

$$\Upsilon = \{(l,p) | (a_l \succeq b_l) \text{ and } (b_p \succeq a_p)\} \quad (2.55)$$

Therefore, it is easy to see that all vectors $v^{(l,p)}$, $(l,p) \in \Upsilon$ are solution to Equation 2.51. Then all vectors in the image of the matrix, denoted by $\mathcal{M}$, in which the columns are vectors $v^{(l,p)}$, $(l,p) \in \Upsilon$, are solutions to that equation.

If there exists a non null solution $x = [x_1 \dots x_n]^T$ for the problem

$$\exists (i,j) \in \Upsilon | a_i \otimes x_i = b_j \otimes x_j \tag{2.56}$$

then Inequalities 2.53 and 2.54 hold true. Therefore, x_i and x_j are generated by the vector $\beta v^{(i,j)}$ by taking β such that $x_i = \beta b_j$. It remains to show that all other non null entries of x_k such that $k \in \{i,j\}$ can be generated by a linear combination of columns of $\mathcal{M}$. In this sense, it is possible to have both possibilities presented below, obtained from Maia et al. (2011b).

1. $(a_k \succeq b_k)$: since $b_j \succeq a_j$ then $(k,j) \in \Upsilon$. x_k can be generated by the image of $v^{(k,j)}$. In this sense α_k is chosen such that $x_k = \alpha_k \otimes b_j$. It remains to show that $\alpha_k \otimes a_k \preceq x_j$, since x_j is already generated by the image of $v^{(i,j)}$. From Inequality 2.54 $a_k \otimes x_k \preceq a_i x_i$, since $a_i x_i = b_j x_j$. By Inequality 2.55, b_j is a non null scalar number, then x_i, x_j and x_k are generated by $\alpha_k v^{c(i,j,k)} \oplus \beta v^{(i,j)}$, in which $c(i,j,k) = (k,j)$.

2. $(b_k \succeq a_k)$: since $a_i \succeq b_i$, then $(i,k) \in \Upsilon$. The proof follows the same reasoning of the item (1), that is, x_k can be generated by the image of $v^{(i,k)}$. To this end α_k is chosen such that $x_k = \alpha_k a_i$ and $\alpha_k \otimes b_k \preceq x_i$ must be ensured, since x_i is already generated by the image of $v^{(i,j)}$. By Inequality 2.54, $b_k \otimes x_k \preceq b_j x_j$, since $b_j x_j = a_i x_i$ then $b_k \otimes \alpha_k \otimes a_i \preceq a_i x_i$. By Inequality 2.55, a_i is a non null scalar number, then x_i, x_j and x_k are generated by

$$\alpha_k v^{c(i,j,k)} \oplus \beta v^{(i,j)},$$

in which $c(i,j,k) = (i,k)$.

Due to the idempotency of dioid $\mathbb{Z}_{max}$, the non null solution x is described as a linear combination of the columns of $\mathcal{M}$, that is, all solutions to Equation 2.51 belong to the image of $\mathcal{M}$, *i.e.*, the semimodule is finitely generated. Explicitly:

$$x = \bigoplus_{\forall k \notin \{i,j\} \text{ and } (x_k \neq \varepsilon)} (\alpha_k v^{c(i,j,k)}) \oplus \beta v^{(i,j)} \in \mathsf{Im}\mathcal{M}, \tag{2.57}$$

in which $c(i,j,k)$ is taken as (k,j) if $(a_k \succeq b_k)$ or (i,k) otherwise.

Finally, if $a_k \oplus b_k = \varepsilon$ is taken into account by adding a column in the matrix $\mathcal{M}$ in which the k^{th} entry is equal to e and the others are null.

Example: 2.6.1 (Semimodule) *Consider the equation $Ax = Bx$ in which the matrices A and $B \in \mathbb{Z}_{max}^{2\times 2}$ and $x \in \mathbb{Z}_{max}^{2\times 1}$, being*

$$A = \begin{bmatrix} e & 3 \\ 5 & 1 \end{bmatrix} \text{ and } B = \begin{bmatrix} 2 & 3 \\ \varepsilon & 2 \end{bmatrix}$$

and

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

Using the previously mentioned methodology it is possible to write the equation of the rows of matrices A and B, i.e.,

$$e \otimes x_1 \oplus 3 \otimes x_2 = 2 \otimes x_1 \oplus 3 \otimes x_2 \tag{2.58}$$

Then, it is possible to enunciate the following set:

$$\Upsilon_1 = \{(l,p) | (a_l \succeq b_l) \text{ and } (b_p \succeq a_p)\} = \{(2,1),(2,2)\}$$

and find the following vector $v^{(l,p)} \in \mathbb{Z}_{max}^{2\times 1}$*, such that* $v^{(l,p)}(l) = b_p$*,* $v^{(l,p)}(p) = a_l$ *and* $v^{(l,p)}(k) = \varepsilon$ *for all* $k \in \{(l,p)\}$*, i.e.,*

$$v^{(2,1)} = \begin{bmatrix} 3 \\ 2 \end{bmatrix} \text{ and } v^{(2,2)} = \begin{bmatrix} \varepsilon \\ 3 \end{bmatrix}.$$

These linearly independent vectors are used as columns of the matrix M_1*, i.e.,*

$$M_1 = \begin{bmatrix} 3 & \varepsilon \\ 2 & 3 \end{bmatrix}.$$

The columns of M_1 *generate a set of solutions in which the solution to Equation 2.58 belongs. This space was generated by the fist row of Equation* $Ax = Bx$*. The matrix* M_1 *must be used to solve the equation generated from the second row of matrices* A *and* B*, so that:*

$$\begin{bmatrix} 5 & 1 \end{bmatrix} \begin{bmatrix} 3 & \varepsilon \\ 2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \varepsilon & 2 \end{bmatrix} \begin{bmatrix} 3 & \varepsilon \\ 2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Simplifying the matrix equation, there exists

$$8 \otimes x_1 \oplus 4 \otimes x_2 = 4 \otimes x_1 \oplus 5 \otimes x_2$$

so that,

$$\Upsilon_2 = \{(1{,}2)\},$$

and

$$v^{(1,2)} = \begin{bmatrix} 5 \\ 8 \end{bmatrix}.$$

Therefore,

$$M_2 = \begin{bmatrix} 5 \\ 8 \end{bmatrix},$$

with the matrices M_1 and M_2 it is possible to find a matrix M so that the matrix columns generate a finite semimodule in which all solutions to equation $Ax = Bx$ belong to this semimodule. The matrix M is given by:

$$M = M_1 \otimes M_2 = \begin{bmatrix} 8 \\ 11 \end{bmatrix}$$

In this way, the solution $x \in \mathsf{Im}\ M$.

Remark: 2.6.1 *The equation $Ax = Bx$ is equivalent to:*

$$\begin{bmatrix} A \\ B \end{bmatrix} x = \begin{bmatrix} I \\ I \end{bmatrix} y$$

in which I is the identity matrix in max-plus algebra and y is a vector of appropriate dimensions.

Remark: 2.6.2 (Inexistence of Solution) *The equation $Ax = Bx$ has no solution if any rows of matrix A are strictly greater than the respective row of matrix B. Likewise, the assertion is also true if any rows of B are strictly greater than the respective row of matrix A.*

2.7 Modified Alternating Algorithm

This section presents a function useful to find a solution to equations like $\bar{A}x = \bar{B}y$. If a solution exists, it is a fixed point of the algorithm and the following function can provide a solution in a finite number of steps. This is possible because the methodologies deal with the integer case. A finite-time procedure can be ensured if the matrices are doubly G-astic[2], as showed in Cuninghame-Green and Butkovic (2003). In this thesis, the following contribution is presented as a result.

function x=altern($\bar{A}$,$\bar{B}$, x_0)

$r = 0; \quad x(0) := x_0;$

$y(r) := \bar{B} \diagdown\!\!\circ (\bar{A}x(r));$

$x(r+1) := \bar{A} \diagdown\!\!\circ (\bar{B}y(r)) \wedge x(0)$;

while $\bar{A}x(r+1) \neq \bar{B}y(r)$

$$r := r+1; \tag{2.59}$$

$$y(r) := \bar{B} \diagdown\!\!\circ (\bar{A}x(r)); \tag{2.60}$$

$$x(r+1) := \bar{A} \diagdown\!\!\circ (\bar{B}y(r)); \tag{2.61}$$

end

end

Lemma 2.7.1 *If the set*

$$S_{AB}(A, B, \bar{x}) = \{x \in \mathbb{Z}_{max} | \bar{A}x \in \mathsf{Im}\ \bar{B} \text{ and } x \preceq \overline{x}\}, \tag{2.62}$$

is not empty then it has a maximal element, in which $\mathsf{Im}\ \bar{B}$ *are the elements generated by columns of* $\bar{B}$.

Proof: Define $x^{up} = \bigoplus_{x \in S_{AB}} x$*, therefore* $x^{up} \succeq x$, $\forall x \in S_{AB}$*. If* $\{x_1, x_2\} \in S_{AB}$, $\exists \{y_1, y_2 | Ax_1 = By_1 \text{ and } Ax_2 = By_2\}$*, due to linearity,* $Ax_1 \oplus Ax_2 = By_1 \oplus By_2$*, therefore it is straightforward to see that* x^{up} *satisfies the equation* $\bar{A}x = \bar{B}y$ *and,*

[2]The matrices $\bar{A}$ and $\bar{B}$ have at least one finite element on each row and on each column. Such matrices are called doubly G-astic

due to idempotency, if $x^{up} \preceq \overline{x}$*, since* $\bigoplus_{\forall i}(x_i \oplus \overline{x}) = \bigoplus \overline{x} = \overline{x}$*, so* $x^{up} \in S_{AB}$*.*

As a consequence of the presented function, the following modified version of the Theorem proposed by Cuninghame-Green and Butkovic (2003) is presented.

Theorem 2.7.1 *The sequences* $\{x(r)\}$ $(r = 0{,}1{,}...)$ *are non-increasing.*

Proof: The modified method ensures that $x(1) \preceq x(0)$ *. The proof for* $r \succeq 1$ *follows the same reasoning of Cuninghame-Green and Butkovic (2003), that is: Residuation Theory ensures that* $\bar{A}x(r+1) \preceq \bar{B}y(r)$ *and* $\bar{B}y(r) \preceq \bar{A}x(r)$*, so* $\bar{A}x(r+1) \preceq \bar{A}x(r)$*. Since* $y(r) = \bar{B} \backslash\!\!\backslash (\bar{A}x(r))$*, isotony guarantees that* $y(r+1) \preceq y(r)$ *and consequently* $x(r+1) \preceq x(r)$ *.*

Corollary: 2.7.1 *The sequences* $\{y(r)\}$ $(r = 0{,}1{,}...)$ *are non-increasing.*

Property: 2.7.1 *If* $\bar{A}x(r) = \bar{B}y(r-1)$ $(r \succ 0)$ *then* $x(r+1) = x(r)$*, that is* $x(r) = f(x(r))$*, in which* $f(.)$ *is the isotone function implemented by the Alternating Method, that is* $f(x) = (\bar{A} \backslash\!\!\backslash (\bar{B}y)) \wedge x(0)$ *for* $y = \bar{B} \backslash\!\!\backslash (\bar{A}x)$*.*

Proof: If $\bar{A}x(r+1) = \bar{B}y(r)$ *then* $y(r+1) = \bar{B} \backslash\!\!\backslash (\bar{B}y(r))$*. So Residuation Theory ensures that* $y(r+1) \succeq y(r)$*, since* $x = y(r)$ *satisfies the inequality* $\bar{B}x \preceq \bar{B}y(r)$*, being* $x = \bar{B} \backslash\!\!\backslash (\bar{B}y(r))$ *its greatest element. By Corollary 2.7.1,* $y(r+1) \preceq y(r))$*, so* $y(r+1) = y(r)$*. By Theorem 2.7.1,* $x(r+1) \preceq x(r)$*, so it is possible to write* $x(r+1) = (\bar{A} \backslash\!\!\backslash (\bar{B}y(r))) \wedge x(0)$ *and then* $x(r+2) = x(r+1)$*.*

Recall Lemma 2.7.1 to ensure that the greatest solution of Equation $\bar{A}x = \bar{B}y$ that is smaller than the initial condition $x(0)$ exists.

Proposition 2.7.1 *Denote* x^{up} *as the greatest solution of Equation* $\bar{A}x = \bar{B}y$ *that is smaller than the initial condition* $x(0)$*.* x^{up} *is a fixed point of the isotone function* $f(.)$ *implemented by the Alternating Method.*

Proof: *The method implements* $f(x^{up}) = (\bar{A} \diagdown\!\!\circ (\bar{B}y)) \wedge x(0)$ *for* $y = \bar{B} \diagdown\!\!\circ (\bar{A}x^{up})$. *Since* x^{up} *is a solution* $\exists \tilde{y} \in \mathbb{Z}_{max} | \bar{A}x^{up} = \bar{B}\tilde{y}$. *Therefore* $y = \bar{B} \diagdown\!\!\circ (\bar{A}x^{up})$ *can be rewritten as* $y = \bar{B} \diagdown\!\!\circ (\bar{B}\tilde{y})$. *Residuation Theory is used to show that* $\bar{B}y = \bar{B}\tilde{y} = \bar{A}x^{up}$. *As a result,* $f(x^{up}) = (\bar{A} \diagdown\!\!\circ (\bar{A}x^{up})) \wedge x(0)$ *and Residuation Theory ensures that* $\bar{A} \diagdown\!\!\circ (\bar{A}x^{up}) \succeq x^{up}$. *Since* $x^{up} \preceq x(0)$,

$$f(x^{up}) \succeq x^{up}. \tag{2.63}$$

Isotony of multiplication ensures that:

$$\bar{A}f(x^{up}) \succeq \bar{A}x^{up} = \bar{B}\tilde{y}. \tag{2.64}$$

On the other hand, since $f(x^{up}) = (\bar{A} \diagdown\!\!\circ (\bar{A}x^{up})) \wedge x(0)$ *then* $f(x^{up}) \preceq (\bar{A} \diagdown\!\!\circ (\bar{A}x^{up}))$. *As a consequence, isotony of multiplication and Residuation Theory ensures that:*

$$\bar{A}f(x^{up}) \preceq \bar{A}(\bar{A} \diagdown\!\!\circ (\bar{A}x^{up})) = \bar{A}x^{up} = \bar{B}\tilde{y}. \tag{2.65}$$

Inequalities 2.64 and 2.65 ensure that $\bar{A}f(x^{up}) = \bar{B}\tilde{y}$. *By definition* $f(x^{up}) \preceq x(0)$ *and therefore* $f(x^{up})$ *is a solution of Equation* $\bar{A}x = \bar{B}y$ *that is smaller than the initial condition* $x(0)$. *Since* x^{up} *is the greatest solution in this situation,*

$$f(x^{up}) \preceq x^{up}. \tag{2.66}$$

Inequalities 2.63 and 2.66 ensure that x^{up} *is a fixed point of* f, *that is* $f(x^{up}) = x^{up}$.

Property: 2.7.2 *The solution found by the modified method is the greatest one that is smaller than the initial condition* $x(0)$.

Proof: *Recall by Property 2.7.1 that every solution* x *is so that* $x = f(x)$*, in which* f *is the isotone function implemented by the method. For a given initial condition* $x(0)$*, consider* $x = f(x)$ *a solution found by the method. By Lemma 2.7.1 the equation has a greatest solution* x^{up} *such that* $x \preceq x^{up} \preceq x(0)$*. Since* x *is a solution,* $\exists m \in \mathbb{N}^+ | x = f^{(m)}(x(0))$*, in which* $f^{(m)} = f^{(m-1)} \circ f$ *is an isotone function that results from* m *compositions of the function* f*. So* $x \preceq x^{up} \preceq x(0) \Rightarrow f^{(m)}(x) \preceq f^{(m)}(x^{up}) \preceq f^{(m)}(x(0))$*. Since* x *and* x^{up} *(by Property 2.7.1) are solutions (fixed points of* f*), then* $f^{(m)}(x) = x$ *and* $f^{(m)}(x^{up}) = x^{up}$*. Therefore* $f^{(m)}(x) \preceq f^{(m)}(x^{up}) \preceq f^{(m)}(x(0)) \Rightarrow x \preceq x^{up} \preceq x$*. Consequently* $x = x^{up}$*.*

Remark: 2.7.1 *The modified algorithm is a contribution of this thesis: the term* $x(r) := \bar{A} \backslash\!\!\circ (\bar{B} y(r)) \wedge x(0)$ *was added. Due to this modification, using Lemma 2.7.1, Properties 2.7.1 to 2.7.2, Proposition 2.7.1 and Theorem 2.7.1, it is possible to prove that this modified algorithm converge to the greatest solution smaller than the initial condition. This important result for the proposed problem does not appear in Cuninghame-Green and Butkovic (2003). Without this new term, the generated solution cannot be ensured smaller than the initial condition.*

2.7.1 Dealing with Equations such as $Ax \oplus c = Bx \oplus d$

An important equation for this work is the semimodule equation (two-sided linear equations)

$$Dx = Ex.$$

This equation can be solved by the modified alternating algorithm considering an appropriate initial condition and that

$$Dx = y \text{ and } Ex = y, \tag{2.67}$$

that implies

$$\begin{bmatrix} D \\ E \end{bmatrix} \otimes x = \begin{bmatrix} I \\ I \end{bmatrix} \otimes y,$$

in which I is the identity matrix in $\mathbb{Z}_{max}$.

The modified alternating algorithm can also solve equations such as $Ax \oplus c = Bx \oplus d$ using the property presented in Equation 2.67, where A and B are matrices in $\mathbb{Z}_{max}$ and x, c and d are vectors in $\mathbb{Z}_{max}$ of appropriate dimensions, by introducing an auxiliary scalar variable t (see Cuninghame-Green and Butkovic (2003)), resulting in:

$$Ax \oplus ct = Bx \oplus dt. \tag{2.68}$$

So, the Equation 2.68 can be rewritten as:

$$\begin{bmatrix} A & c \end{bmatrix} \otimes \begin{bmatrix} x \\ t \end{bmatrix} = \begin{bmatrix} B & d \end{bmatrix} \otimes \begin{bmatrix} x \\ t \end{bmatrix},$$

once this equation is linear in the extended vector $\bar{x} = [x \ t]^T$.

Considering the initial condition for the modified alternating algorithm as $\bar{x}_0 = [x_0 \ 0]^T$, if an upper bound solution x for $Ax \oplus c = Bx \oplus d$ exists, such that $x \preceq x_0$, the vector $\bar{x}$ will converge to the solution and t will remain equal to 0 (Cuninghame-Green and Butkovic, 2003).

As previously mentioned, the modified alternating algorithm will converge to the greatest solution smaller than the initial condition. Therefore the greatest solution to the equation $Ax \oplus c = Bx \oplus d$ can be found by using the greatest possible initial condition, *i.e.*, $x_0 = \top$.

Recently, a new methodology was proposed for finding the smallest solution to Equation 2.68 considering a special semimodule. The method introduces the concept of weak residuation and strong residuation for an element (for more details see Gonçalves (2015)).

2.8 Conclusion

This chapter presented useful preliminary concepts for comprehension of this work. The Petri nets, more precisely the timed-event graphs are used to model the systems and the max-plus algebra is used to mathematically describe the dynamic behavior of these systems. These tools, as well as, the Residuation Theory and the Theory of Semimodules, are fundamental to develop the control strategies presented in the next chapter.

Chapter 3

Control Problem Formulation and Optimal Synthesis

3.1 Introduction

The control problem formulations and the optimal control synthesis will be presented in this chapter. The formulation is original and some important classes of problems can be obtained from it, for example, the open-loop Just-in-Time control problem and the Feedback control problem. The formulation is based on optimization problems formulations, so it is desired to maximize a merit function $g(Z)$ of interest subject to some constraints. The main constraint is given by a semimodule equation (non convex constraint in conventional algebra).

For instance, in this thesis, the main objective is maximize the input dates for the system for the desired horizon, *i.e.*, it is desired to delay as much as possible the input dates, in order to avoid inventory generation and respect some constraints. The constraints are: a reference demand, a desired system dynamic and the system characteristics. For example, a desired dynamic can be when a piece of a product enters the system after other piece, *i.e.*, the entry order of different pieces must be respected. Other dynamic constraint can be when a machine finishes a certain service before another machine can do it. The system dynamic constraints can be described by using linear equations in max-plus algebra.

From a proposed general formulation, the open-loop Just-in-Time control with a finite horizon is initially presented. Two ways to find a solution to the control problem are also proposed. Then, the open-loop Just-in-Time formulation for infinite horizon is presented. In order to perform the control in infinite horizon, some max-plus algebraic tools are used to simplify the search for solutions. The control in infinite horizon can be understood as a finite horizon large enough, which can have a huge computational cost, making the finding a solution not feasible. It will be discussed in an appropriate moment in the following sections.

In the end, the feedback control policy in Just-in-Time context is developed. The conditions for the existence of a feedback matrix F are discussed and the maximal feedback matrix that guarantees that the system will respect the deadline dates is found. If the maximal feedback matrix is non causal, a way to find a causal feedback matrix from the non causal matrix is presented.

Numerical examples to illustrate the applicability of the proposed methodologies is presented in each section. The examples will also be used to illustrate the relevance and importance of systems, which can use the proposed methodologies.

3.2 General Optimization Control Problem Formulation

To formulate the general control problem it is considered a Max-Plus Linear System described by the following state space equations:

$$\begin{cases} x(k) = Ax(k-1) \oplus Bu(k), \\ y(k) = Cx(k). \end{cases} \tag{3.1}$$

In order to constrain the system evolution, the system output dates evolution will be bounded by deadline dates generated using a reference model. The reference model is also a Max-Plus Linear System described by:

$$\begin{cases} x_m(k) = A_m x_m(k-1) \oplus B_m u_m(k), \\ y_m(k) = C_m x_m(k), \end{cases} \tag{3.2}$$

where the vectors $x(k) \in \mathbb{Z}^n_{max}$ and $x_m(k) \in \mathbb{Z}^p_{max}$ are the state vectors (internal), $u(k) \in \mathbb{Z}^n_{max}$ and $u_m(k) \in \mathbb{Z}^l_{max}$ are the k^{th} firing date of input transitions and $y(k)$ and $y_m(k)$ are the k^{th} output dates. A, A_m, B, B_m, C and C_m are the system matrices of appropriate dimensions. The following constraint

$$Cx(k) \preceq C_m x_m(k), \forall k \geq k', \tag{3.3}$$

must hold, *i.e.*, the system output date must be smaller than or equal to the reference model output dates $\forall k \succeq k'$, for a given k', and, in addition, the system state must respect semimodule constraints such as:

$$\mathcal{R} = \{x(k) | Dx(k) = Ex(k)\} \tag{3.4}$$

in which D and E are matrices of state constraints of appropriate dimensions. It is important to remark that the vector state $x(k)$ can be the system state or an extended vector state $\bar{x}(k)$ defined as

$$\bar{x}(k) = \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}, \tag{3.5}$$

i.e., the constraint can be applied to the state vector and also to the inputs.

Remark: 3.2.1 *Constraints like $Tx(k) \preceq Rx(k)$ can be easily written as $Dx(k) = Ex(k)$, since $Tx(k) \preceq Rx(k) \Rightarrow (T \oplus R)x(k) = Rx(k) \Rightarrow Dx(k) = Ex(k)$.*

Remark: 3.2.2 *Constraint like $Tx(k) \preceq x(k) \Rightarrow T^*x(k) = x(k) \Rightarrow x(k) \in \mathsf{Im}T^*$, in which the operator $(.)^*$ is the Kleene star operator. (For more details see Subsection 2.4.3 and Baccelli et al. (1992)).*

Considering what was presented before, the general control problem can be defined in the following way.

Definition 3.2.1 (General Control Problem Formulation) *The general multiobjective control problem can be defined as:*

$$\max \ (g_i(Z))$$

subjected to:

$$x(k) = Ax(k-1) \oplus Bu(k) \tag{3.6}$$

$$y(k) = Cx(k) \tag{3.7}$$

$$x_m(k) = A_m x_m(k-1) \oplus B_m u_m(k) \tag{3.8}$$

$$y_m = C_m x_m(k) \tag{3.9}$$

$$y(k) \preceq y_m(k) \tag{3.10}$$

$$Dx(k) = Ex(k) \tag{3.11}$$

$$x_m(0) = x_m^0 \tag{3.12}$$

where x_m^0 *is the initial condition for the the reference model for the system.* $g_i(Z)$ *is a objective function of interest (a parameter or variable of system),* $i = 1$ *to* N_{obj}*, being* N_{obj} *the number of objective functions.*

Assumption: 3.2.1 *The operator* max *in the objective function of control problem formulation means to find the maximal value to objective functions* $g_i(Z)$*, for* $i = 1$ *to* N_{obj}*, being* N_{obj} *the number of objective functions.*

Remark: 3.2.3 *For instance, the control problem objective in this thesis is to find the maximal input vector* $u(k)$ *for the system computing the maximum entries* $u_i, \forall i,$ *of* $u(k)$*.*

Remark: 3.2.4 *The constraint of Equation 3.11 is very general because it can include constraints on the system inputs. To illustrate this assertion, it is possible to show that the equation*

$$x(k) = Ax(k-1) \oplus Bu(k), \tag{3.13}$$

is equivalent to

$$\begin{cases} x(k) = Ax(k-1) \oplus Bu(k), \\ u(k) = u(k-1) \oplus s(k), \end{cases}$$

considering the expanded state

$$\bar{x}(k) = \begin{bmatrix} x(k) \\ u(k) \end{bmatrix},$$

and $s(k)$ as the system input. Therefore the expanded state equation can be rewritten as

$$\begin{bmatrix} x(k) \\ u(k) \end{bmatrix} = \begin{bmatrix} A & B \\ \varepsilon & I \end{bmatrix} \begin{bmatrix} x(k-1) \\ u(k-1) \end{bmatrix} \oplus \begin{bmatrix} B \\ I \end{bmatrix} s(k)$$

and the Equation 3.11 can be rewritten as

$$D\bar{x}(k) = E\bar{x}(k). \tag{3.14}$$

From the general formulation, some classes of problems can be obtained as presented in the following sections.

3.3 The Open-Loop Just-in-Time Control Problem in a Finite Horizon

3.3.1 Introduction

The open-loop just-in-time control problem can be understood as finding the maximum firing dates of the input transition u, so that the firing dates of the output transition y occurs before given deadline dates. For a fixed horizon, from firing k up to firing $k + N_p - 1$, in which N_p is the prediction horizon, it is necessary to delay as much as possible the input firing dates in order to respect the output deadline dates for the firings $k + 1$ until $k + N_p$.

More precisely, let k be the firing number from which it is required to make the predictions and $\hat{y}(k + l)$ the output firing dates predicted for a given step l.

Given the information on the state of the system at the firing number k and the input firing dates firing from k up to firing $k + N_p - 1$, the future firing date outputs $\hat{x}(k+j|k)$ can be predicted for the firings $k+1$ until $k+N_p$, *i.e.*:

$$\hat{x}(k+j|k) = A^j x(k) \oplus \sum_{i=0}^{j-1} A^{j-i-1} B u(k+i),$$

$$\hat{y}(k+j|k) = CA^j x(k) \oplus \sum_{i=0}^{j-1} CA^{j-i-1} B u(k+i).$$

This is a well-known result from classical control theory of linear systems (See for instance Garcia et al. (1989); De Schutter and van den Boom (2001)). In matrix notation:

$$\hat{x}(k) = H_1 \hat{u}(k) \oplus G_1 x(k) \tag{3.15}$$

$$\hat{y}(k) = H_2 \hat{u}(k) \oplus G_2 x(k). \tag{3.16}$$

in which

$$\hat{x}(k) = \begin{bmatrix} x(k+1|k) \\ \vdots \\ x(k+N_p|k) \end{bmatrix}, \tag{3.17}$$

$$H_1 = \begin{bmatrix} B & \varepsilon & \cdots & \varepsilon \\ AB & B & \cdots & \varepsilon \\ \vdots & \vdots & \ddots & \vdots \\ A^{N_p-1}B & A^{N_p-2}B & \cdots & B \end{bmatrix}, \tag{3.18}$$

$$G_1 = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{N_p} \end{bmatrix}, \tag{3.19}$$

$$\hat{y}(k) = \begin{bmatrix} y(k+1|k) \\ \vdots \\ y(k+N_p|k) \end{bmatrix}, \tag{3.20}$$

$$H_2 = \begin{bmatrix} CB & \varepsilon & \cdots & \varepsilon \\ CAB & CB & \cdots & \varepsilon \\ \vdots & \vdots & \ddots & \vdots \\ CA^{N_p-1}B & CA^{N_p-2}B & \cdots & CB \end{bmatrix}, \tag{3.21}$$

$$G_2 = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{N_p} \end{bmatrix} \tag{3.22}$$

and

$$\hat{u}(k) = \begin{bmatrix} u(k) \\ \vdots \\ u(k+N_p-1) \end{bmatrix}. \tag{3.23}$$

Then, the Equation 3.6 can be replaced by Equation 3.15 without losing the generality.

Therefore it is desired to delay as much as possible the admission dates of material into the system, given by $\hat{u}(k)$, in order to deliver the products in accordance with a known *a priori* reference demand $\hat{r}(k)$ (deadline dates), that is $\hat{y}(k) \preceq \hat{r}(k)$.

For the control in a finite horizon, the reference demand will be given by *a priory* fixed vector $\hat{r}(k)$, generated by a reference model. So $A_m = [\varepsilon]$, $B_m = I$ and $C_m = I$, in which $[\varepsilon]$ is a matrix of appropriate dimensions with entries equal to ε and I is the identity matrix in max-plus algebra. The reference demand will be given by:

$$\hat{r}(k) = \left[\begin{array}{ccccc} y_m(k) & y_m(k+1) & y_m(k+2) & \cdots & y_m(k+N_p-1) \end{array}\right]. \tag{3.24}$$

In this section it is also considered time constraints on the dynamics of the state, so the semimodule constraint can be replaced by an equation such as $\hat{D}\hat{x}(k) = \hat{E}\hat{x}(k)$. In this sense it is defined the following multi-objective control problem.

Definition 3.3.1 (Open-loop Just-in-Time Control Problem in a Finite Horizon) *The multi-objective control problem can be formulated as follows:*

$$\max\left(\hat{u}(l)_{(k \leq l \leq k+N_p-1)}\right) \tag{3.25}$$

subjected to:

$$\hat{x}(k) = H_1\hat{u}(l) \oplus G_1 x(k), \tag{3.26}$$

$$\hat{D}\hat{x}(k) = \hat{E}\hat{x}(k), \tag{3.27}$$

$$\hat{r}(k) = y_m(k) = C_m[A_m x_m(k-1) \oplus B_m u_m(k)], \tag{3.28}$$

$$\hat{y}(k) \preceq y_m(k), \tag{3.29}$$

$$u(l) \succeq u_{\min}(k) \tag{3.30}$$

where $\hat{D}$ and $\hat{E}$ are matrices of appropriate dimensions, $x(k)$ is the system state at date k, i.e., the state when the control begins. $A_m = [\varepsilon]$, $B_m = I$, $C_m = I$ and $u_m(k+1) = \top$ for $k \succeq N_p$, being $\top$ the greatest element in max-plus algebra.

$u_{\min}(k) = [u(k)\ u(k)\ \cdots\ u(k)]^T$.

Remark: 3.3.1 *The control problem in Definition 3.3.1, requires finding the maximal system input dates* $\hat{u}(l)$, $k \preceq l \preceq k + N_p - 1$, *computing the maximum entries* $\hat{u}_i$, $\forall i$, *of* $\hat{u}(l)$.

The set of solutions of Equation 3.27 can be expressed as a semimodule whose generator can be computed by several methods, one of them was presented in Section 2.6.1 (Butkovic and Hegedus, 1984)(Allamigeon et al., 2008). If it is necessary only one solution, it can be computed by the Modified Alternating Method (Gomes da Silva and Maia, 2014) (Cuninghame-Green and Butkovic, 2003).

Definition 3.3.2 *The reference* $\hat{r}(k)$ *is viable if* $\hat{r}(k) \succeq y_{\min}(k)$, *in which* $y_{\min}(k) = H_2 u_{\min}(k) \oplus G_2 x(k)$, *i.e., the reference* $\hat{r}(k)$ *is viable if it is possible for the system to produce outputs at the desirable dates.*

Proposition 3.3.1 *A necessary and sufficient condition for the existence of a solution is the set* $\Omega_u = \{\hat{u} \quad | \hat{D}(H_1\hat{u} \oplus G_1 x(k)) = \hat{E}(H_1\hat{u} \oplus G_1 x(k))\}$ *to be non-empty and the reference* $\hat{r}(k)$ *be viable.*

Proof: Necessity is straightforward. Sufficiency: Residuation Theory ensures that the greatest solution of Inequality 3.29 exists. The Equation $\hat{D}(H_1\hat{u} \oplus G_1 x(k)) = \hat{E}(H_1\hat{u} \oplus G_1 x(k))$ *can be rewritten as* $Ax = By$ *and Lemma 2.7.1 completes the proof.*

3.3.2 Determination of Minimum Viable Reference

Unlike conventional systems, the output firing dates of a TEG can only occur after a certain amount of time, since they depend on the token sojourn times. As a result the reference demand $\hat{r}(k)$ (deadline dates) could be not viable. However this issue can be overcome by computing the minimum modification, in an additive sense, in order to make $\hat{r}(k)$ viable.

Fastest output firing dates

Up to the k^{th} firing date of the system output, it is assumed that it is known the $(k-1)^{th}$ firing date of the input of the system $u(k-1)$. Since $\hat{u}(k)$ is non-decreasing, in order to obtain the fastest output response for a given system from firings k up to $k+N_p$, all future inputs are set to $u(k-1)$, *i.e.*, $\hat{u}(k+j) = u(k-1)$ for all $0 \preceq j \preceq (N_p - 1)$. In this sense $u_{\min}(k)$ is defined as the vector $N_P \times 1$ in which all entries are equal to $u(k-1)$. As a result the smallest output dates for the horizon $[k+1, \ldots, k+N_p]$ are:

$$y_{\min}(k) = H_2 u_{\min}(k) \oplus G_2 x(k) \tag{3.31}$$

in which,

$$u_{\min}(k) = \begin{bmatrix} u(k) \\ \vdots \\ u(k) \end{bmatrix}.$$

The smallest viable projection

The $y_{\min}(k)$ is the fastest firing dates of the system output for the horizon $[k+1, \ldots, k+N_p]$. Therefore a reference demand $\hat{r}(k)$ cannot be viable if $\hat{r}(k) \not\succeq y_{\min}(k)$. However, based on the ideas of Menguy et al. (2000), it is possible to obtain the smallest viable projection in $\oplus$ sense.

Definition 3.3.3 (The smallest viable projection) *The smallest viable pro-*

jection of x (on y) is denoted by $P_y(x) = \triangle_x \oplus x$, in which $\triangle_x$ is the smallest element of the set:

$$\Psi_z = \{z | z \oplus x \succeq y\}.$$

Property: 3.3.1 *The smallest viable projection can be computed by $P_y(x) = x \oplus y$.*

Proof: It follows from Definition 3.3.3 that $x \oplus y \in \Psi_z$. As a consequence $\triangle_x \preceq x \oplus y$ and $\triangle_x \oplus x \preceq x \oplus y$, thanks to the idempotency of $\oplus$. On the other hand $\triangle_x \oplus x \succeq y$, so $\triangle_x \oplus x \succeq y \oplus x$. Therefore $\triangle_x \oplus x = x \oplus y$.

As a result the minimum viable reference closest to the desired one, in $\oplus$ sense, is

$$P_{y_{\min}}(\hat{r}(k)) = y_{\min}(k) \oplus \hat{r}(k). \tag{3.32}$$

Remark: 3.3.2 *It is important to notice that if $\hat{r}(k)$ is viable, then $P_{y_{\min}}(r(k)) = \hat{r}(k)$.*

3.3.3 Performing the Open-loop in Just-in-Time Control in a Finite Horizon

Two methods for computing the solution to the control problem in Definition 3.3.1 are presented in the following. The first one is based on the semimodule generation and the second one uses the Alternating Method.

Solution through semimodule generation

Remark: 3.3.3 *As presented in Section 2.7.1, to solve the equation $Ax \oplus c = Bx \oplus d$ it is necessary to consider an auxiliary single scalar variable t. It is clear* *that the equation has a solution with t finite if and only if it has a solution with* $t = 0$*. This result allows to state the following assumption. (for more details see Cuninghame-Green and Butkovic (2003))*

Assumption: 3.3.1 *It is assumed hereafter that the set* $\Omega_x = \{\hat{x} \quad | \hat{D}\hat{x} = \hat{E}\hat{x}\}$ *is non-empty, in which* $\hat{x}(k) = H_1\hat{u}(k) \oplus G_1x(k)$ *and* $x(k)$ *is the system state at the beginning of the prediction horizon.*

Remark: 3.3.4 *Considering the Property 3.3.1, Remark 3.3.3 and the Assumption 3.3.1, the control problem has a solution and, consequently, the auxiliary scalar variable t can be set equal to zero.*

It is possible to show that all solutions to Equation 3.27, in which $\hat{D}$ and $\hat{E}$ are matrices representing the constraints and $\hat{x}(k)$ is a vector, can be expressed as a finitely generated semimodule, that is $\hat{x}(k) \in \mathsf{Im}\ M$, in which M is a matrix (Butkovic and Hegedus, 1984) (Gaubert, 1992). In this sense, the following proposition can be stated.

Proposition 3.3.2 *Consider a semimodule* $\hat{D}\hat{x}(k) = \hat{E}\hat{x}(k)$ *generated by* $\Omega_x = \mathsf{Im}\ M$ *and the smallest viable projection of the reference demand* $P_{y_{\min}}(\hat{r}(k))$*. If the control problem of Definition 3.3.1 is solvable, the optimal solution to the control problem can be computed by*

$$\hat{u}(l) = \bar{M}[(H_2\bar{M})\diagdown\!\!\!\circ P_{y_{\min}}(\hat{r}(k)) \wedge \bar{Q}\diagdown\!\!\!\circ 0] \tag{3.33}$$

where $\bar{M}$ *is a matrix such that* $\hat{u}(k) \in \mathsf{Im}\bar{M}$ *and* $\bar{M}$ *generated by methodology of Section 2.6.1,* $P_{y_{\min}}(\hat{r}(k))$ *is the smallest viable projection for the reference demand and* H_2 *is defined in Equation 3.21.*

Proof: Replacing Equation 3.26 in Equation 3.27, it is possible to write:

$$\hat{D}H_1\hat{u}(l) \oplus \hat{D}G_1x(k) = \hat{E}H_1\hat{u}(l) \oplus \hat{E}G_1x(k) \tag{3.34}$$

which is equivalent to:

$$\begin{bmatrix} \hat{D}H_1 & \hat{D}G_1x(k) \end{bmatrix} \begin{bmatrix} \hat{u}(k) \\ t \end{bmatrix} = \begin{bmatrix} \hat{E}H_1 & \hat{E}G_1x(k) \end{bmatrix} \begin{bmatrix} \hat{u}(l) \\ t \end{bmatrix}$$

being t *an auxiliary variable and, considering Remark 3.3.1,* $t = 0$*. Consequently, by theory of semimodule*

$$\begin{bmatrix} \hat{u}(l) \\ t \end{bmatrix} \in \mathsf{Im} \begin{bmatrix} \bar{M} \\ \bar{Q} \end{bmatrix}$$

where $\bar{M}$ *and* $\bar{Q}$ *are matrices of appropriate dimensions found by using the technique presented in Section 2.6.1. Then*

$$\begin{bmatrix} \hat{u}(l) \\ t \end{bmatrix} = \begin{bmatrix} \bar{M} \\ \bar{Q} \end{bmatrix} \otimes v$$

where v *is an appropriate vector.*

As the control problem is solvable $t = 0 \Rightarrow \bar{Q}v = 0$*. Therefore*

$$\hat{u}(l) = \bar{M}v, \tag{3.35}$$

provided that v *is such that* $Qv = 0$*. Looking for the deadline dates, it is considered*

$$H_2\hat{u}(l) \oplus G_2x(k) \preceq P_{y_{\min}}(\hat{r}(k)). \tag{3.36}$$

This inequality is equivalent to

$$H_2\hat{u}(l) \preceq P_{y_{\min}}(\hat{r}(k)), \tag{3.37}$$

$$G_2x(k) \preceq P_{y_{\min}}(\hat{r}(k)). \tag{3.38}$$

Recall by Equation 3.31 that $y_{\min}$ *is such that* $Gx(k) \preceq y_{\min}$ *and by Equation 3.32 that* $y_{\min} \preceq P_{y_{\min}}(\hat{r}(k))$*. Hence, Inequality 3.38 is always satisfied. Replacing Equation 3.35 in Inequality 3.37, it is possible to show that:*

$$H_2\bar{M}v \preceq P_{y_{\min}}(\hat{r}(k)). \tag{3.39}$$

The solution of Inequality 3.39 can be obtained by using the Residuation Theory:

$$v = (H_2\bar{M}) \backslash\!\!\circ P_{y_{\min}}(\hat{r}(k)) \wedge \bar{Q} \backslash\!\!\circ 0, \tag{3.40}$$

provided that v *is such that* $\bar{Q}v = 0$*.*

Remember that the Residuation Theory ensures that this v *is the greatest one such that Inequality 3.39 holds*[1] *and the control problem has a solution only if* $\bar{Q}v = 0$*. Therefore, by replacing it into Equation 3.35, it is found the greatest input dates that satisfy the smallest viable projection of the reference demand, i.e.,*

$$\hat{u}(l) = \bar{M}[(H_2\bar{M}) \backslash\!\!\circ P_{y_{\min}}(\hat{r}(k)) \wedge \bar{Q} \backslash\!\!\circ 0]. \tag{3.41}$$

[1] If a system of the form $Ax = B$ and $x \preceq C$ has a solution, then $x^{up} = A \backslash\!\!\circ B \wedge C$ is its biggest solution. Indeed, a solution x must be such that $x \preceq x^{up}$. So isotony of multiplication and residuation theory ensure $B = Ax \preceq Ax^{up} \preceq B$, which means that $Ax^{up} = B$. Moreover, $x^{up} \preceq C$.

Provided that it is possible to compute a generator for semimodule Ω_x, expressed as an image of a matrix M, Proposition 3.3.4 ensures that the greatest firing dates of the input can be computed by Equation 3.41. However, in general situations, the computation of such generator demands a huge computational effort. In order to deal with this issue, it is proposed a method based on the Alternating Method, which will be presented in the following section.

Solution through Alternating Method

The restriction presented in Inequality 3.29 is equivalent to

$$H_2\hat{u}(l) \preceq P_{y_{\min}}(\hat{r}(k))), \tag{3.42}$$

$$G_2 x(k) \preceq P_{y_{\min}}(\hat{r}(k))). \tag{3.43}$$

where H_2 and G_2 are defined in Equations 3.21 and 3.22 and $P_{y_{\min}}(\hat{r}(k)))$ is the smallest viable projection for the reference demand.

As shown in the previous subsection, Inequality 3.43 is always satisfied for the smallest viable projection. As a consequence, the greatest vector $\hat{u}(l)$ that satisfies the Inequality 3.42 always exists and it is given by the following equation:

$$\hat{u}^{max} = H_2 \backslash\!\!\circ P_{y_{\min}}(\hat{r}(k)). \tag{3.44}$$

The vector $\hat{u}^{max}$ obtained in Equation 3.44 is an upper bound for the solutions and it is used as the initial condition for the Alternating Method. Recall that this method can find a solution $\hat{x}$ for Equation $\hat{D}\hat{x} = \hat{E}\hat{x}$.

Remark: 3.3.5 *If the solution exists, it is a fixed point of the Modified Alternating algorithm.*

Proposition 3.3.3 *Consider the semimodule of constraint* $\Omega_x = \{\hat{x} \ |\hat{D}\hat{x} = \hat{E}\hat{x}\}$ *non-empty and the smallest viable projection of the reference demand, that is* $P_{y_{\min}}(\hat{r}(k)) = \hat{r}(k) \oplus y_{\min}(k)$. *By taking* $\hat{u}(0) = H_2 \diagdown\!\!\!\!\diagup P_{y_{\min}}(\hat{r}(k))$ *as the initial condition, the Alternating Method converges to the optimal solution of the control problem solving the affine equation*

$$\hat{D}H_1\hat{u}(l) \oplus \hat{D}G_1x(k) = \hat{E}H_1\hat{u}(l) \oplus \hat{E}G_1x(k), \tag{3.45}$$

in which $x(k)$ *is the state at the beginning of the prediction horizon.*

Proof: The proof follows from Property 2.7.2, which ensures that the Alternating Method will lead to the largest $\hat{u}$ *such that* $\hat{x}(k) \in \Omega_x$ *and* $\hat{u} \preceq H_2 \diagdown\!\!\!\!\diagup P_{y_{\min}}(\hat{r}(k))$.

Remark: 3.3.6 *Proposition 3.3.3 gives another important method to compute, in general in a more efficient way, the solution to the control problem. It is important to remark that this solution also leads to the greatest output smaller than the smallest viable projection of the reference demand.*

A Particular Case of Open-loop Control in a Finite Horizon

In some applications it is possible or necessary to consider constraints only on the inputs, therefore, the control problem in a finite horizon can be simplified. Based on this fact the following optimization control problem is defined.

Definition 3.3.4 (Particular Open-loop Just-in-Time Control Problem in Finite Horiz

The multi-objective control problem can be formulated as follows:

$$\max(\hat{u}(l)_{k \leq l \leq k+N_p-1}) \tag{3.46}$$

subjected to:

$$\hat{y}(k) = H_2\hat{u}(l) \oplus G_2x(k), \tag{3.47}$$

$$\hat{D}\hat{u}(l) = \hat{E}\hat{u}(l), \tag{3.48}$$

$$\hat{r}(k) = y_m(k) = C_m[A_mx_m(k-1) \oplus B_mu_m(k)], \tag{3.49}$$

$$\hat{y}(k) \preceq y_m(k), \tag{3.50}$$

$$u(l) \succeq u_{\min}(k) \tag{3.51}$$

where $\hat{D}$ *and* $\hat{E}$ *are matrices of appropriate dimensions representing the constraints on* $\hat{u}(l)$*,* $x(k)$ *is the system state at date* k*, i.e., the state when the control begins.* $A_m = [\varepsilon]$*,* $B_m = I$*,* $C_m = I$ *and* $u_m(k+1) = \top$ *for* $k \succeq N_p$*, being* $\top$ *the greatest element in max-plus algebra.* $u_{\min}(k) = [u(k)\ u(k)\ \cdots\ u(k)]$*.*

In order to solve this control problem, two methods for computing the solution are presented. The first one is based on the semimodule generation and the second one uses the Alternating Method.

Solution through semimodule generation

In the same way as previously presented, it is possible to show that all solutions to Equation 3.48, in which $\hat{D}$ and $\hat{E}$ are matrices and $\hat{u}$ is a vector, can be expressed as a finitely generated semimodule , that is $\hat{u} \in \mathsf{Im}\ M$, in which M is a matrix (Butkovic and Hegedus, 1984) (Gaubert, 1992). Based on this result the following proposition is stated.

Proposition 3.3.4 *Consider a semimodule* $\Omega_u = \{\hat{u}(l)|\hat{D}\hat{u}(l) = \hat{E}\hat{u}(l)\}$ *generated by* $\Omega_u = \mathsf{Im}\ M$ *and the smallest viable projection of the reference demand*

$P_{y_{\min}}(\hat{r}(k))$. *The optimal solution to the control problem can be computed by*

$$\hat{u}(l) = M((H_2M) \diagdown\!\!\circ P_{y_{\min}}(\hat{r}(k))). \tag{3.52}$$

Proof: If $\hat{u} \in \mathsf{Im}\ M$*, it is possible to write:*

$$\hat{u} = Mv, \tag{3.53}$$

for a given vector v*. By replacing Equations 3.47, 3.49 and 3.53 into Inequality 3.50, it is possible to write*

$$H_2Mv \oplus G_2x(k) \preceq P_{y_{\min}}(\hat{r}(k)). \tag{3.54}$$

Inequality 3.54 is equivalent to

$$H_2Mv \preceq P_{y_{\min}}(\hat{r}(k)) \tag{3.55}$$

$$G_2x(k) \preceq P_{y_{\min}}(\hat{r}(k)) \tag{3.56}$$

Recall from Equation 3.31 that $y_{\min}$ *is such that* $G_2x(k) \preceq y_{\min}$ *and from Equation 3.32 that* $y_{\min} \preceq P_{y_{\min}}(\hat{r}(k))$*. Hence, Inequality 3.56 is always satisfied. The solution of Inequality 3.55 can be obtained using the Residuation Theory:*

$$v = (H_2M) \diagdown\!\!\circ P_{y_{\min}}(\hat{r}(k)). \tag{3.57}$$

Remember that the Residuation Theory ensures that this v *is the greatest one so that Inequality 3.55 holds. Therefore, by replacing it into Equation 3.53, it is found the greatest input dates that satisfy the smallest viable projection of the* *reference demand, i.e.,*

$$\hat{u}(l) = M((H_2 M) \diamond\!\!\!\setminus P_{y_{\min}}(\hat{r}(k))). \tag{3.58}$$

Corollary: 3.3.1 *If the constraint can be expressed as* $T\hat{u}(l) \preceq \hat{u}(l)$, *that is* $T\hat{u}(l) \oplus \hat{u}(l) = \hat{u}(l)$, *the optimal solution to the control problem can be computed by*

$$\hat{u}(l) = T^*((H_2 T^*) \diamond\!\!\!\setminus P_{y_{\min}}(\hat{r}(k))) \tag{3.59}$$

Proof: The equivalence: $Tx \preceq x \Leftrightarrow T^*x = x \Leftrightarrow x \in \mathsf{Im}\ T^*$, $T^* = \bigoplus_{i \in \mathbb{N}} T^i$, *in which* $(.)^*$ *is the Kleene star operator. Therefore the generator of the semimodule* $T\hat{u}(l) \preceq \hat{u}(l)$ *is* $\mathsf{Im}\ T^*$.

Solution through Alternating Method

The restriction presented in 3.50 is equivalent to

$$H_2 \hat{u}(l) \preceq P_{y_{\min}}(\hat{r}(k)), \tag{3.60}$$

$$G_2 x(k) \preceq P_{y_{\min}}(\hat{r}(k)). \tag{3.61}$$

As shown in the previous subsection, Inequality 3.61 is always satisfies for the smallest viable projection. As a consequence, the greatest vector $\hat{u}(l)$ that satisfy the Inequality 3.60 always exists and it is given by the following equation:

$$\hat{u}^{max} = H_2 \diamond\!\!\!\setminus P_{y_{\min}}(\hat{r}(k)). \tag{3.62}$$

The vector $\hat{u}^{max}$ obtained with Equation 3.62 is an upper bound for the solutions and it is used as initial condition for the Alternating Method. Recall that this method can find a solution $\hat{u}$ for Equation $\hat{D}\hat{u} = \hat{E}\hat{u}$.

Proposition 3.3.5 *Consider the semimodule of constraint* $\Omega_u = \{\hat{u} \ \ |\hat{D}\hat{u} = \hat{E}\hat{u}\}$*, non-empty and the smallest viable projection of the reference demand, that is* $P_{y_{\min}}(\hat{r}(k)) = \hat{r}(k) \oplus y_{\min}(k)$*. By taking* $\hat{u}(0) = H_2 \backslash P_{y_{\min}}(\hat{r}(k))$ *as the initial condition, the Alternating Method converges to the optimal solution to the control problem.*

Proof: The proof follows from Property 2.7.2, which ensures that the Alternating Method will lead to the largest $\hat{u} \in \Omega_u$ *so that* $\hat{u} \preceq H_2 \backslash P_{y_{\min}}(\hat{r}(k))$*.*

Remark: 3.3.7 (Computational complexity) *The computational complexity of the approach based on semimodule generation depends on the fact that the solution to* $Ax = Bx$*, where* $A,B \in \mathbb{Z}^n_{max}$ *are row vectors, are hyperplanes finitely generated. Therefore, in the worst case, to compute the solution to all the rows, the algorithm is double exponential (Butkovic and Hegedus, 1984)(Katz, 2007). However, for special types of semimodules, as the one presented in Corollary 3.3.1, in which the generator of the semimodule belongs to the image of a matrix generated by the Kleene star operation, the complexity is polynomial. On the other hand, the approach based on the "Modified Alternating" method has pseudo-polynomial complexity*[2] *for several important situations and works quite well in practice, since in many cases the input size is suitable.*

Remark: 3.3.8 *Houssin et al. (2013) presented a state restriction, in the form* $\phi x \preceq x$ *(remark that after state expansion this restriction is convex), which is similar to the one presented in Corollary 3.3.1, but in their restriction* $\phi \in$

[2] A pseudo-polynomial complexity algorithm is one that its running time is polynomial in the dimension of the problem input and the magnitudes of the data involved (provided these are given as integers), i.e., a pseudo-polynomial algorithm will display exponential behavior only when confronted with instances containing exponentially large numbers. Such algorithms are technically exponential functions of their input size and are therefore not considered polynomial. (Garey and Johnson, 1979)

$\mathcal{M}_{in}^{ax}[\gamma,\delta]$, *they are operating in another dioid, and the solution to the control problem is based on order reversing mapping (they rely on the Dual Residuation Theory). In addition, they investigate the infinite horizon control problem, whose transitions fire as soon as possible, that is, it is the opposite of Just-in-Time control. Their results are based on the cases in which the matrix* B *is diagonal and the constraint matrix is factorized by the same matrix* B*. Unlike their approach, this thesis is interested in a Just-in-Time control problem defined as a multi-objective optimization problem, which allows to deal with a finite horizon control problem and non-convex constraints of the form* $Dx = Ex$*. Based on Residuation Theory, it is presented necessary and sufficient conditions to solve the problem and, for an important class of constraints, an explicit expression to compute the best solution in polynomial time is obtained. In addition two methods to solve the problem: one based on semimodule generation and another one based on the modified version of the alternating method is presented, which is also a contribution. In this thesis there is no a priori restriction concerning the forms of matrices* B *or the ones associated with the input constraints.*

3.3.4 Numerical Examples

Two numerical examples is presented in this section. The first one is a multi-objective formulation of the problem already studied in De Schutter and van den Boom (2001), whose solution can be explicitly computed by using Corollary 3.3.1. The second one is a more complex problem with several machines and non-convex constraints (in conventional algebra).

Numerical Example 1

The following TEG represents a manufacturing system, which is the same as the one described in De Schutter and van den Boom (2001), except from the fact that the authors describe a model with single input while in this example a more complex model that has two inputs is analyzed, so it is desired to delay as much as possible the admission of raw material into the two inputs of the system. The

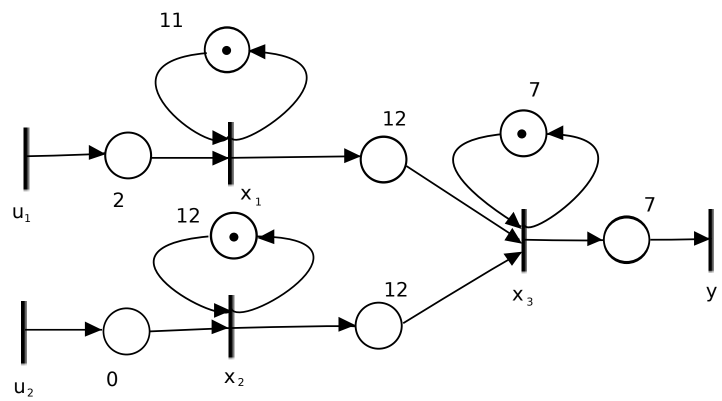


Figure 3.1: Timed events graph of a Manufacturing System

TEG model can written as in Equation 3.1, so that:

$$A = \begin{bmatrix} 11 & . & . \\ . & 12 & . \\ 23 & 24 & 7 \end{bmatrix}, B = \begin{bmatrix} 2 & . \\ . & e \\ 14 & 12 \end{bmatrix}$$

and

$$C = \begin{bmatrix} . & . & 7 \end{bmatrix},$$

in which the element ε is represented by a dot.

As proposed in De Schutter and van den Boom (2001), it is desired the interval between two successive firing dates to be in the interval [2 12], that is ($2 \preceq \triangle u_i(k) \preceq 12$, $i = 1{,}2$). Those restrictions can be rewritten as:

$$\begin{cases} 2u_1(l) \preceq u_1(l+1), \\ -12u_1(l+1) \preceq u_1(l), \\ 2u_2(l) \preceq u_2(l+1), \\ -12u_2(l+1) \preceq u_2(l), \end{cases} \tag{3.63}$$

They can be written in a matrix form as:

$$Ju(l) \preceq Iu(l). \tag{3.64}$$

As a consequence, they can also be written as the equation $(J \oplus I)u(l) = Iu(l)$, in which I is the max-plus identity matrix. These constraints are of the type given in Corollary 3.3.1, so given a reference demand, the solution to the control problem can be computed explicitly, in a polynomial time, by Equation 3.59.

In order to present numerical results, the same values used by De Schutter and van den Boom (2001) are considered. In this case, the prediction horizon $N_p = 8$ is chosen. The matrices $\hat{D}$, $\hat{E}$, H_2 and G_2 will not be showed because of their large size, but they are defined in Equations 3.48, 3.21 and 3.22.

The initial condition is:

$$x(0) = \begin{bmatrix} 0 & 0 & 10 \end{bmatrix}^T,$$

and the reference demand is:

$$\hat{r}(k) = \begin{bmatrix} 40 & 45 & 55 & 66 & 75 & 85 & 90 & 100 \end{bmatrix}^T.$$

From Equation 3.31 the smallest output dates are:

$$y_{\min}(k) = \begin{bmatrix} 31 & 43 & 55 & 67 & 79 & 91 & 103 & 115 \end{bmatrix}^T.$$

These output dates can be understood as the output generated by the system if no control is applied. Therefore, by the reference demand, the smallest viable projection of the reference demand is

$$P_{y_{\min}}(\hat{r}(k)) = \begin{bmatrix} 40 & 45 & 55 & 67 & 79 & 91 & 103 & 115 \end{bmatrix}^T.$$

Using Equation 3.59, the greatest input dates that satisfies $P_{y_{\min}}(\hat{r}(k))$ are given by:

$$\hat{u}_1(k) = \begin{bmatrix} 12 & 23 & 34 & 46 & 58 & 70 & 82 & 94 \end{bmatrix}^T,$$

$$\hat{u}_2(k) = \begin{bmatrix} 12 & 24 & 36 & 48 & 60 & 72 & 84 & 96 \end{bmatrix}^T,$$

and the output dates are given by:

$$\hat{y}(k) = \begin{bmatrix} 33 & 44 & 55 & 67 & 79 & 91 & 103 & 115 \end{bmatrix}^T.$$

The vector $\hat{y}(k)$ is greater than the vector $y_{\min}(k)$, *i.e.*, the control applied makes the system produces output dates closer to reference demand and it also delays the input dates for the system, avoiding inventory generation in the system.

In order to solve the problem by the Modified Alternating Method, the initial condition (an upper bound) is computed by Equation 3.62:

$$\hat{u}_1^{max}(k) = \begin{bmatrix} 12 & 23 & 34 & 46 & 58 & 70 & 82 & 94 \end{bmatrix}^T,$$

$$\hat{u}_2^{max}(k) = \left[\begin{array}{cccccccc} 12 & 24 & 36 & 48 & 60 & 72 & 84 & 96 \end{array}\right]^T.$$

It is interesting to observe that this initial condition is already the solution to the problem.

Remark: 3.3.9 (Computational Time) *Recall that computational complexity of the methods was discussed in Remark 3.3.7. For this example, the solution can be obtained almost instantaneously and the computational time with both methods is the same. It takes by the semimodule approach and the Alternating Method 0.001 seconds (average value for 10 experiments), for a desktop computer, Intel Core i5 2.53GHz, 4GB RAM, Windows 10, 64bits, Cache 3.932 GB using the Max-Plus toolbox of the Computational Package Scicoslab* [3].

Remark: 3.3.10 *The obtained results here are different from those of De Schutter and van den Boom (2001) since the approach in this thesis is interested in a just-in-time formulation with two inputs, which leads to a multi-objective optimization problem. However it is possible to solve the same problem for the single input TEG as presented in De Schutter and van den Boom (2001) by using the presented approach. In order to compare the approaches, recall that in the formulation of De Schutter and van den Boom (2001) the mono-objective function has two parts, one involving the output and the reference demand and another involving the input, which is given in conventional algebra by:*

$$J = J_{out} + J_{in} = \sum_{j=1}^{N_p} max(\hat{y}(k+j|k) - r(k+j), 0) - \sum_{j=1}^{N_p} u(k+j-1). \quad (3.65)$$

So the optimal solution can lead to $\hat{y}(k+j|k) \succeq P_{y_{\min}}(r(k))(j)$, i.e., it can violates the smallest viable projection of the reference demand. This is the case

[3] http://www.scicoslab.org/

of the example presented in De Schutter and van den Boom (2001), whose results are:

The desirable reference demand is

$$\hat{r}(k) = \begin{bmatrix} 40 & 45 & 55 & 66 & 75 & 85 & 90 & 100 \end{bmatrix}^T,$$

and the obtained output dates are

$$\hat{y}(k) = \begin{bmatrix} 33 & 43 & 56 & 67 & 79 & 91 & 103 & 115 \end{bmatrix}^T,$$

by using the obtained input dates

$$\hat{u}(k) = \begin{bmatrix} 12 & 24 & 35 & 46 & 58 & 70 & 82 & 94 \end{bmatrix}^T.$$

Using the proposed methodology in this thesis, this situation never happens because in the just-in-time formulation, constraints avoid such violation. Actually, the just-in-time formulation ensures that J_{in} and $\sum_{j=1}^{N_p} |\hat{y}(k+j|k) - P_{y_{\min}}(r(k))(j)|$ are minimized. In addition the optimal solution can be computed explicitly in a polynomial time by Equation 3.59.

Numerical Example 2

Consider a manufacturing system given by the TEG shown in Figure 3.2. This system has three input transitions, five processing units, which can process only one product at each time, and one output transition. Places p_i, $i \in \{1, \ldots 5\}$, indicate when the process unit i is working. Transitions x_1, x_2, x_3, x_7, x_9 indicate the beginning of the process for units $1, \ldots, 5$, while transitions $x_4, x_5, x_6, x_8, x_{10}$ indicate the end of their respective process. Synthetically, in this model resources

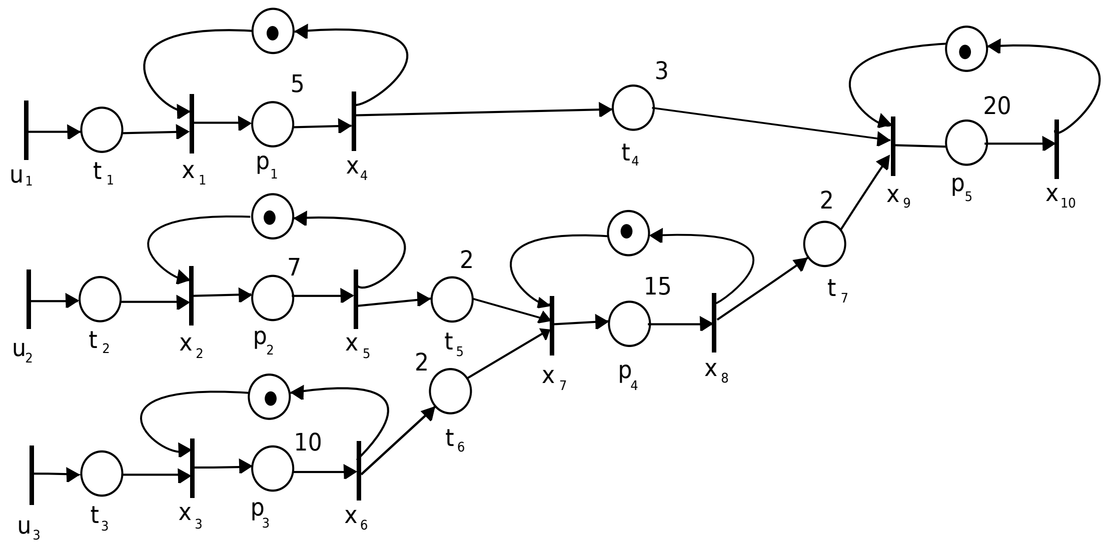


Figure 3.2: Timed events graph of a Manufacturing System

are provided to the system through input transitions u_1, u_2 and u_3. Then they are transported to the first three units. This action is indicated by the places t_1, t_2 and t_3. Transitions x_1, x_2 and x_3 indicate the beginning of the processing event and processing actions are indicated by the places p_1, p_2 and p_3. After processing, resources coming from u_2 and u_3 are transported (places t_5 and t_6) to be processed together in the forth processing unit indicated by p_4. The result of this process is then transported (t_7) to be processed in the fifth (final) unit, indicated by place p_5, together with a part that comes from u_1, which availability is indicated in the place t_4. The conclusion of the manufacturing process is indicated by transition x_{10}. The transportation and processing times are indicated over the respective places.

The TEG model can be represented by the Equation 3.1, with

$$A = \begin{bmatrix} . & . & . & e & . & . & . & . & . & . \\ . & . & . & . & e & . & . & . & . & . \\ . & . & . & . & . & e & . & . & . & . \\ . & . & . & 5 & . & . & . & . & . & . \\ . & . & . & . & 7 & . & . & . & . & . \\ . & . & . & . & . & 10 & . & . & . & . \\ . & . & . & . & 9 & 12 & . & e & . & . \\ . & . & . & . & 24 & 27 & . & 15 & . & . \\ . & . & . & 8 & 26 & 29 & . & 17 & . & e \\ . & . & . & 28 & 46 & 49 & . & 37 & . & 20 \end{bmatrix}, B = \begin{bmatrix} e & . & . \\ . & e & . \\ . & . & e \\ 5 & . & . \\ . & 7 & . \\ . & . & 10 \\ . & 9 & 12 \\ . & 24 & 27 \\ 8 & 26 & 29 \\ 28 & 46 & 49 \end{bmatrix}$$

and

$$C = \begin{bmatrix} . & . & . & . & . & . & . & . & . & e \end{bmatrix}.$$

Suppose that limit the input firing rate in a time interval is necessary, according to a given resource availability. For a system with m inputs, it is possible to write:

$$a_i \preceq \triangle u_i \preceq b_i, \tag{3.66}$$

for each $i = 1{,}2 \ldots m$. Inequality 3.66 is equivalent to:

$$\begin{cases} a_i u_i(k) \preceq u_i(k+1), \\ -b_i u_i(k+1) \preceq u_i(k). \end{cases} \tag{3.67}$$

For the present example, the aim is to ensure that the firing dates of u_1 are in an interval between 2 and 35 time units ($2 \preceq \triangle u_1(k) \preceq 35$); the firing dates of u_2 are in an interval between 0 and 45 time units ($0 \preceq \triangle u_2(k) \preceq 45$) and the firing dates of u_3 are in an interval between 3 and 47 time units ($3 \preceq \triangle u_3(k) \preceq 47$). These constraints can be rewritten as Inequalities 3.67, that is:

$$\begin{cases} 2u_1(k) \preceq u_1(k+1), \\ -35u_1(k+1) \preceq u_1(k), \\ 0u_2(k) \preceq u_2(k+1), \\ -45u_2(k+1) \preceq u_2(k), \\ 3u_3(k) \preceq u_3(k+1), \\ -47u_3(k+1) \preceq u_3(k). \end{cases} \tag{3.68}$$

Inequalities 3.68 can be easily written in a matrix notation:

$$Ju(k) \preceq Iu(k). \tag{3.69}$$

In addition, this inequality can be rewritten as an equation, since $Ju(k) \preceq Iu(k) \Leftrightarrow (J \oplus I)u(k) = Iu(k)$, in which I is the identity matrix in max-plus algebra.

Control problems with Inequalities of the type 3.69 can be solved explicitly in a very efficient way by using Corollary 3.3.1. However, the proposed approach in this thesis can handle more general constraints that take into account several system inputs with different firing numbers and even non-convex ones in conventional algebra. In this sense, suppose that it is also desired that the k^{th} firing of the input u_3 occurs immediately after two time units after the k^{th} firing of u_1 and five time units after k^{th} firing date of u_2, *i.e.*, $u_3(k) = 2u_1(k) \oplus 5u_2(k)$. This constraint equation, which is non convex in conventional algebra, can also be written in matrix form

$$Qu(k) = Ru(k), \tag{3.70}$$

in which Q and R are matrices of appropriate dimensions.

So the overall constraints set, given by Equations 3.69 and 3.70, can be written as $Du(k) = Eu(k)$, in which

$$D = \begin{bmatrix} J \oplus I \\ Q \end{bmatrix}, E = \begin{bmatrix} I \\ R \end{bmatrix}.$$

As cited previously, this semimodule of constraints can be expressed as $u \in \text{Im}\, M$. The matrix M will not showed because of its large size. To solve this example, suppose that the initial condition is given by

$$x(0) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T,$$

and $u_1(0) = u_2(0) = u_3(0) = 0$. So

$$x(1) = \begin{bmatrix} 0 & 0 & 0 & 5 & 7 & 10 & 12 & 27 & 29 & 49 \end{bmatrix}^T.$$

In addition, suppose that the prediction in the horizon [2,3 . . . 10] is required and the reference demand is given by

$$\hat{r}(k) = \begin{bmatrix} 50 & 70 & 100 & 135 & 150 & 155 & 170 & 200 & 270 \end{bmatrix}^T.$$

By Equation 3.31 the smallest output firing dates for the horizon [2,3 . . . 10] are:

$$y_{\min}(k) = \begin{bmatrix} 69 & 89 & 109 & 129 & 149 & 169 & 189 & 209 & 229 \end{bmatrix}^T,$$

whose are the same for the system if no control is applied considering all inputs available at the initial prediction horizon date. Therefore the smallest viable projection of the reference demand is

$$P_{y_{\min}}(\hat{r}(k)) = \begin{bmatrix} 69 & 89 & 109 & 135 & 150 & 169 & 189 & 209 & 270 \end{bmatrix}^T.$$

Using Equation 3.57, the greatest input firing dates that satisfies the $P_{y_{\min}}(\hat{r}(k))$ are

$$\hat{u}_1(k) = \begin{bmatrix} 18 & 38 & 58 & 75 & 78 & 98 & 118 & 153 & 188 \end{bmatrix}^T,$$

$$\hat{u}_2(k) = \begin{bmatrix} 15 & 35 & 55 & 72 & 75 & 95 & 115 & 155 & 200 \end{bmatrix}^T,$$

$$\hat{u}_3(k) = \begin{bmatrix} 20 & 40 & 60 & 77 & 80 & 100 & 120 & 160 & 205 \end{bmatrix}^T,$$

and the largest outputs that respect $P_{y_{\min}}(r(k))$ are

$$\hat{y}(k) = \begin{bmatrix} 69 & 89 & 109 & 129 & 149 & 169 & 189 & 209 & 254 \end{bmatrix}^T.$$

Therefore, as expected, the applied control delays as much as possible the input dates and it makes the system produce the output dates closer to the reference demand.

In order to find a solution by the Alternating Method, Equation 3.62 gives an upper bound for the input firing dates:

$$\hat{u}_1^{max}(k) = \begin{bmatrix} 41 & 61 & 81 & 101 & 101 & 121 & 141 & 181 & 242 \end{bmatrix}^T,$$

$$\hat{u}_2^{max}(k) = \begin{bmatrix} 23 & 43 & 63 & 83 & 83 & 103 & 123 & 163 & 224 \end{bmatrix}^T,$$

$$\hat{u}_3^{max}(k) = \begin{bmatrix} 20 & 40 & 60 & 80 & 80 & 100 & 120 & 160 & 221 \end{bmatrix}^T.$$

The upper bound is used as the initial condition for the Alternating Method. As expected, the obtained results are the same as those obtained by the previous method.

Remark: 3.3.11 (Computational Time) *The obtained solution with both methods are the same, as expected, but the processing times are quite different. It takes the semimodule approach 12.577 seconds while it takes the one based on the Alternating Method 0.012 second (average value for 10 experiments), for a desktop computer, Intel Core i5 2.53GHz, 4GB RAM, Windows 10, 64bits, Cache 3.932 GB. In addition, the semimodule method leads to the matrix M of size* 27×1022.

3.3.5 Conclusion

In this section, the open-loop Just-in-Time control problem in finite horizon for TEG was presented, whose aim is to compute the latest input firing dates in order to respect a given demand profile. This kind of control was studied in a situation in which the input and the system state dynamics are constrained by a given semimodule. The necessary and sufficient conditions to solve the presented problems were given. In addition, two methods to solve them were introduced: one based on semimodule generation and another one based on the proposed modified version of the Alternating Method. Numerical examples have enabled the illustration of the applicability of the results and the discussion of the computational complexity of both methods.

3.4 The Open-Loop Just-in-Time Control Problem in Infinite Horizon

3.4.1 Introduction

If the number of desired deadline dates are not bounded, *i.e.*, if the schedule is generated infinitely, the open-loop Just-in-Time control is performed in infinite horizon. It is important to remark that the problem addressed in this section can be solved by the method presented in the previous section considering the horizon large enough, *i.e.*, the horizon bigger than the transient interval of the system. Based on the general control problem, the following particular open-loop control problem in infinite horizon is introduced.

Definition 3.4.1 (Open-loop Just-in-Time Control Problem in Infinite Horizon) *The Just-in-Time control problem in an infinite horizon can be defined as:*

$$\max \; (u(k)_{\forall k \succ 0})$$

subjected to:

$$x(k) = Ax(k-1) \oplus Bu(k) \tag{3.71}$$

$$x_m(k) = A_m x_m(k-1) \oplus B_m u_m(k) \tag{3.72}$$

$$Cx(k) \preceq C_m x_m(k) \tag{3.73}$$

$$Dx(k) = Ex(k) \tag{3.74}$$

$$x_m(0) = x_m^0 \tag{3.75}$$

in which, x_m^0 is the initial condition for the reference model for the system.

Remark: 3.4.1 *The control problem in Definition 3.4.1 requires finding the maximal system input dates* $u(k)$*,* $k \succeq 0$*, computing the maximum entries* u_i*,* $\forall i$*, of* $u(k)$*.*

3.4.2 Performing the Open-Loop Just-in-Time Control in Infinite Horizon

Firstly the following definitions were presented.

Definition 3.4.2 *((A,B) Max-Plus Geometrically Invariant Sets): (Gonçalves, 2015) A semimodule* $\mathcal{X} \subseteq \mathbb{Z}_{max}$ *is said (A,B) Max-Plus geometrically invariant if for every* $x \in \mathcal{X}$ *there exists an* $u \in \mathbb{Z}_{max}$ *so that* $Ax \oplus Bu \in \mathcal{X}$*.*

Definition 3.4.3 (Controllable Coupled Problem) *(Gonçalves, 2015) A control problem is said controllable coupled if the existence of a solution implies that*

$$\exists M \in \mathbb{N} | \forall x \in \mathcal{X} - \{\varepsilon\}, \forall i,j |x_i - x_j| \preceq M. \tag{3.76}$$

where $\mathcal{X}$ *is the set of all feasible states for the problem. This definition means that there exists a variable* M *such that all system states* x *in the set of feasible states for the problem* $\mathcal{X}$ *has the difference between two entries of* x *smaller than or equal to* M*. Therefore, in controllable coupled problems the difference between the entries of the state vector will be bounded.*

Remark: 3.4.2 *In order to make a controllable coupled problem, the variable* M *can be large enough. Controllable coupled problems deal with synchronizing joint subset not with synchronizing disjoint transitions, what means there is at least one transition that is not connected to the other ones. Many practical applications can be modeled as a controllable coupled problem.*

Assumption: 3.4.1 *In this section a kind of Regulator Control Problem is addressed and this control problem is assumed as a controllable couple problem.*

The Residuation Theory ensures that the greatest solution to Inequality 3.73 exists. Therefore the necessary and sufficient condition for the existence of a solution to the proposed open-loop Just-in-Time Control Problem in infinite horizon is that the system evolution must respect the evolution of a given reference model, while keeping the state inside the semimodule $\mathcal{R} = \{x(k)|Dx(k) = Ex(k)\}$. Therefore, if the optimization problem is solvable, it is possible to show that there exists a (A,B) max-plus geometrically invariant set $\mathcal{X}$ inside $\mathcal{R}$ by using linearity.

Definition 3.4.4 *(Maximal (A,B) Max-Plus Geometrically Invariant Sets): Gonçalves (2015) Given the semimodule $\mathcal{R}$, $\mathcal{X}^*(A,B,\mathcal{R})$ is the Maximal (A,B) Max-Plus Invariant Set inside $\mathcal{R}$. Formally, if $\mathcal{X}$ is a (A,B) Max-Plus Geometrically Invariant Set,*

$$\mathcal{X}^*(A,B,\mathcal{R}) = \bigcup\{\mathcal{X}|\mathcal{X} \subseteq \mathcal{R}\}. \tag{3.77}$$

$\mathcal{X}^*$ is the set that contains all the (A,B) Max-Plus Geometrically Invariant Sets contained in $\mathcal{R}$.

Assumption: 3.4.2 *Hereafter, the optimization problem is assumed as solvable. Therefore, the set $\mathcal{X}^*$ is assumed as non-empty (for more details see Gonçalves (2015)).*

If $\mathcal{X}^*$ is finitely generated and the problem is a controllable coupled problem, it can be characterized by the image of a matrix $\mathcal{M}$, *i.e.*, $\mathcal{X}^* \in \mathsf{Im}\ \mathcal{M}$ which implies that $x(k) = \mathcal{M}v(k)$, the matrix $\mathcal{M}$ can be found by several methods (see for instance Butkovic and Hegedus (1984) and Katz (2007)). Therefore,

$$Ax(k-1) \oplus Bu(k) = \mathcal{M}v(k).$$

In order to solve the problem, it is sought a trajectory inside $\mathcal{X}^*$, *i.e.*, $x(k) \in \mathcal{X}^*, \forall k \succeq 1$. Therefore, as $\mathcal{X}^*$ is non empty by Assumption 3.4.2, there exists $u(k)$ such that $Ax(k-1) \oplus Bu(k) = \mathcal{M}v(k) \in \mathcal{X}^*$.

An initial condition inside $\mathcal{X}^*$ can be obtained by finding a solution to the equation

$$A\mathcal{M}v(0) \oplus Bu(1) = \mathcal{M}v(1) \tag{3.78}$$

since $x(k) = \mathcal{M}v(k)$. The constraint

$$Cx(k) \preceq Cx_m(k) \Rightarrow C\mathcal{M}v(k) \preceq Cx_m(k)$$

implies that the greatest value to $v(k)$ is given by Residuation Theory as:

$$v(k)^{max} = C\mathcal{M} \backslash Cx_m(k).$$

Therefore, the system initial condition $x(0) = \mathcal{M}v(0)$, inside $\mathcal{X}^*$ and according to the reference model, can be found by Modified Alternating Algorithm considering the $v(0)^{max}$ and $v(1)^{max}$ as the initial conditions for the algorithm.

For each date k, the greatest input dates $u(k)$ can also be found by Modified Alternating Algorithm (Cuninghame-Green and Butkovic (2003) Gomes da Silva and Maia (2014)) from the initial state $x(0) \in \mathcal{X}^*$ by the following equation:

$$\begin{bmatrix} Ax(k-1) & B \end{bmatrix} \begin{bmatrix} t \\ u(k) \end{bmatrix} = \mathcal{M}v(k), \tag{3.79}$$

being $t = 0$ an auxiliary variable and considering $v(k)^{max}$ the initial condition for the algorithm in step k.

Remark: 3.4.3 *An interesting particular reference model is autonomous and evolves by a rate* λ_m *with the structure* $x_m(k) = \Lambda_m x_m(k-1)$, *being*

$$\Lambda_m = \begin{bmatrix} \lambda_m & . & . & \cdots & . \\ . & \lambda_m & . & \cdots & . \\ . & . & \lambda_m & \cdots & . \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ . & . & . & \cdots & \lambda_m \end{bmatrix} = \ diag \left\{ \ \lambda_m, \ \ \lambda_m, \ \ \lambda_m, \ \ \cdots \ \ \lambda_m \ \right\}, \tag{3.80}$$

then it does not have a transient interval. Therefore, a particular control problem that deals with this kind of reference model is presented in the following.

Assumption: 3.4.3 *The results obtained so far, allow us to state that if the control problem is solvable, then there exists a biggest system trajectory that is achievable inside* $\mathcal{X}(A,B,\mathcal{R})$, *which is of the form of Equation 3.71. Moreover, we can easily show that this trajectory is always upper bounded by the biggest reference model, whose* $A_m = A$, $B_m = B$ *and* $Ax_m(k-1) \oplus Bu_m(k) \preceq C \diagdown\!\!\!\circ (C_m x_m(k))$. *In fact, in this case we can have desired dynamics by choosing* $u_m(k)$ *appropriately. So hereafter, in order to simplify the presentation and the mathematical developments, we assume that the reference model follows the system dynamics, that is:* $A_m = A$, $B_m = B$ *and* $C_m = C$.

Particular Reference Model

Among all possible reference models, a particular interesting one is a model that operates without a transient behavior. Likewise, in several situations, this behavior is also desirable for the system. Therefore, a way to find a solution to an open-loop Just-in-Time control problem is to consider an important class of reference model that evolves by a rate λ_m as $x_m(k) = \Lambda_m x_m(k-1)$.

Definition 3.4.5 ***(Open-Loop Just-in-Time Control in Infinite Horizon With a Particular Reference Model)***

The Just-in-Time control problem in infinite horizon can be defined as:

$$\max \ (u(k)_{\forall k \succ 0})$$

subjected to:

$$x(k) = Ax(k-1) \oplus Bu(k) \tag{3.81}$$

$$x_m(k) = \Lambda_m x_m(k-1) \tag{3.82}$$

$$Cx(k) \preceq Cx_m(k) \tag{3.83}$$

$$Dx(k) = Ex(k) \tag{3.84}$$

$$x_m(0) = x_m^0 \tag{3.85}$$

in which, x_m^0 is the initial condition for the reference model for the system.

Remark: 3.4.4 *The control problem in Definition 3.4.5 requires finding the maximal system input dates $u(k)$, $k \succeq 0$, computing the maximum entries u_i, $\forall i$, of $u(k)$.*

The main goal of open-loop Just-in-Time control is to make the system evolve in accordance with a desirable schedule, in this way the constraint in Equation 3.83 can be rewritten as:

$$\bar{C}\bar{x}(k) \preceq \bar{C}_m \bar{x}_m(k) \tag{3.86}$$

with

$$\bar{C} = \begin{bmatrix} C \\ \varepsilon_{n\times m} \end{bmatrix}^T, \bar{C}_m = \begin{bmatrix} C_m \\ \varepsilon_{n\times m} \end{bmatrix}^T, \bar{x}_m(k) = \begin{bmatrix} x_m(k) \\ u_m(k) \end{bmatrix},$$

and $\varepsilon_{n\times m}$ is a matrix with entries equal to ε of appropriate dimensions. So, the semimodule constraint applied to the inputs ($D^u u(k) = E^u u(k)$) and the semimodule constraint applied to the state ($D^x x(k) = E^x x(k)$) can be rewritten as

$$\bar{D}\bar{x}(k) = \bar{E}\bar{x}(k) \tag{3.87}$$

with

$$\bar{D} = \begin{bmatrix} D^x & D^u \end{bmatrix} \tag{3.88}$$

and

$$\bar{E} = \begin{bmatrix} E^x & E^u \end{bmatrix}. \tag{3.89}$$

Property: 3.4.1 *The solution to the control problem belongs to the set $\mathcal{T}$, being this set defined as*

$$\mathcal{T} = \{\bar{x} | \exists k \ \bar{C}\bar{x} \preceq \bar{C}_m \bar{x}_m(k) \ \text{and} \ \bar{D}\bar{x} = \bar{E}\bar{x}\}$$

The solution to the constraint in Equation 3.84 can be expressed as a semimodule whose generator can be computed by several methods (Butkovic and Hegedus, 1984)(Allamigeon et al., 2008). Therefore, the solution to this equation can be written as:

$$\bar{x}(k) = Mv \tag{3.90}$$

in which, the matrix M is generated by an appropriate method (see for instance Maia et al. (2011b)).

Proposition 3.4.1 (Upper Limit to State Vector) *The upper bound $l_s(k) \in \mathcal{T}$ to $\bar{x}(k)$ that solves the control problem at date k is given by:*

$$l_s(k) = M[M \diamond\!\!\!\!\setminus (\bar{C} \diamond\!\!\!\!\setminus (\bar{C}_m \bar{x}_m(k)))] \tag{3.91}$$

Proof: Since the system output dates need to be smaller than or equal to the reference model output dates, $\bar{C}\bar{x}(k) \preceq \bar{C}_m \bar{x}_m(k)$. The Residuation Theory sets the greatest value to $\bar{x}(k)$ given by $\bar{x}(k) = \bar{C} \diamond\!\!\!\!\setminus (\bar{C}_m \bar{x}_m(k))$. By the Theory of Semimodule, $\bar{D}\bar{x}(k) = \bar{E}\bar{x}(k) \Rightarrow \bar{x}(k) = Mv$. Thereby $Mv \preceq \bar{C} \diamond\!\!\!\!\setminus (\bar{C}_m \bar{x}_m(k)) \Rightarrow v = M \diamond\!\!\!\!\setminus (\bar{C} \diamond\!\!\!\!\setminus (\bar{C}_m \bar{x}_m(k))) \Rightarrow l_s(k) = M[M \diamond\!\!\!\!\setminus (\bar{C} \diamond\!\!\!\!\setminus (\bar{C}_m \bar{x}_m(k)))]$.

Lemma 3.4.1 *The greatest element $\bar{x}^{up} \in \mathcal{T}$ always exists if the semimodule $\bar{D}\bar{x}(k) = \bar{E}\bar{x}(k)$ is non empty.*

Proof: Since the semimodule $\bar{D}\bar{x}(k) = \bar{E}\bar{x}(k)$ is non empty, the solution is given by $\bar{x}(k) = Mv$. If $\bar{x}(k) \preceq l_s(k) \Rightarrow Mv \preceq l_s(k)$, so, by the Residuation Theory, $v \preceq (M \diamond\!\!\!\!\setminus l_s(k)) \Rightarrow \tilde{x}(k) = Mv \in \mathcal{T}$ and $x^{up} = M(M \diamond\!\!\!\!\setminus l_s(k))$.

Lemma 3.4.2 *If the semimodule constraint to the control problem can be written as $Ex \preceq x$, the greatest element $x^{up} \in \mathcal{T}$ always exists.*

Proof: Let $\varphi = C \diamond\!\!\!\!\setminus (C_m x_m)$ and $Cx \preceq C_m x_m \Rightarrow x \preceq \varphi$. Using the Kleene star operation, the inequality $Ex \preceq x$ is such that $E^* x = x$. Consider $y = E^* \diamond\!\!\!\!\setminus \varphi$ the greatest solution to:

$$E^* y \preceq \varphi \Leftrightarrow E^*(E^* y) \preceq \varphi,$$

because $E^* = E^* E^*$. Then, $E^* y$ is also solution to the previous equation and therefore:

$$E^*y \preceq y.$$

But, $E^* \succeq I \Rightarrow E^*y \succeq y \Rightarrow E^*y = y$, *i.e.,* $y \in \mathcal{R}_x$, *being* $\mathcal{R}_x = \{x|E^* = x\}$, *and* $x^{up} \preceq y \Rightarrow x^{up} = y$.

Sufficient Conditions

Every system has a maximum performance λ_A (the maximum cycle mean of a TEG) and, to control a system. Suppose that, in order to control a certain system, it is necessary to ensure that the system evolves according to a desirable performance λ_m. So, it is desirable that the chosen value of λ_m is such that $\lambda_m \succeq \lambda_A$ because, otherwise, the system cannot evolve in accordance with the desirable performance. If the trajectory demand generated by reference model is non-viable to the chosen λ_m, $y_m(k) \not\succeq y_{\min}(k)$ to some $k \succeq q$. However, it is possible to find the smallest viable projection of $y_m(k)$ (on $y_{\min}(k)$) $P_{y_{\min}(k)}(y_m(k))$. The smallest viable reference closest to the desirable one is given by

$$P_{y_{\min}(k)}(y_m(k)) = y_{\min}(k) \oplus y_m(k), \forall k \succeq q. \tag{3.92}$$

If $y_m(k)$ is viable to all $k \succeq q$, then $P_{y_{\min}(k)}(y_m(k)) = y_m(k)$.

Remark: 3.4.5 *The reference model will evolve in accordance with the following equation:*

$$\bar{x}_m(k) = \lambda_m \bar{x}_m(k-1). \tag{3.93}$$

In this way, the k^{th} firing dates of the transitions are related to the initial condition by the following equation:

$$\bar{x}_m(k) = \lambda_m^k \bar{x}_m(0). \tag{3.94}$$

The Equation 3.94 can be placed in Equation 3.91, resulting in:

$$l_s(k) = M[M \diagdown (\bar{C} \diagdown (\bar{C}_m \lambda_m^k \bar{x}_m(0)))] \tag{3.95}$$

Proposition 3.4.2 (Evolution to the State Upper Limit) *The evolution of state upper limit for the system described by Equation 3.94 is given by the following relation:*

$$l_s(k+1) = \lambda_m l_s(k) \tag{3.96}$$

Proof: The only variable element in Equation 3.91 is $x_m(k)$ and by using the Equation 3.93 it is straightforward to verify that $l_s(k+1) = \lambda_m l_s(k)$.

Remark: 3.4.6 *The state upper limit vector $l_s(k)$ is defined over $\bar{x}_m(k)$ which entries are the state $x_m(k)$ for the reference model and the firing dates of the inputs $u_m(k)$. Therefore, it is possible to write*

$$l_s(k) = \begin{bmatrix} l_s^x(k) \\ l_s^u(k) \end{bmatrix} \tag{3.97}$$

where $l_s^x(k)$ is the internal upper bound and $l_s^u(k)$ is the input upper bound at date k.

The Reached System States:

Considering the equations 3.81, 3.83 and 3.87, the reached system states must comply with the following equations:

$$\begin{aligned} x(k) &= Ax(k-1) \oplus Bu(k) \\ D^x x(k) &= E^x x(k) \\ D^u u(k) &= E^u u(k) \end{aligned}$$

These equations can be written as a equation $\hat{D}\hat{x}(k) = \hat{E}\hat{x}(k)$, in which

$$\hat{x}(k) = \left[\begin{array}{ccc} x(k-1) & x(k) & u(k) \end{array} \right]^T.$$

Therefore, considering an extended upper bound vector to $\hat{x}(k)$ equal to

$$\hat{l}_s(k) = \left[\begin{array}{ccc} l_s^x(k-1) & l_s^x(k) & l_s^u(k) \end{array} \right]^T$$

and the equation $\hat{D}\hat{x}(k) = \hat{E}\hat{x}(k)$ is equivalent to

$$\begin{cases} \hat{D}\hat{x}(k) = Iz(k) \\ \hat{E}\hat{x}(k) = Iz(k) \end{cases}$$

that can be written as

$$\left[\begin{array}{c} \hat{D} \\ \hat{E} \end{array} \right] \hat{x}(k) = \left[\begin{array}{c} I \\ I \end{array} \right] z(k). \tag{3.98}$$

Therefore, the maximum solution to the vector $\hat{x}(k)$ equal to or smaller than $\hat{l}_s(k)$, if the solution exists, can be found by the Modified Alternating algorithm.

Remark: 3.4.7 *If a solution to Equation 3.98 exists, the auxiliary variable t in $\hat{x}(k)$ must converge to zero and remains zero at $x_m(k)$, for a given extended upper bound.*

Alternatively, the maximum solution can be found using the Theory of Semi-module, since $\hat{D}\hat{x}(k) = \hat{E}\hat{x}(k) \Rightarrow \hat{x}(k) \in \mathsf{Im}\hat{M}$, $\forall k$, therefore

$$\hat{x}(k) = \hat{M}w(k),$$

and for each date k it is possible find an upper limit to state $\hat{l}_s(k)$, then

$$w^{max}(k) = \hat{M} \setminus \hat{l}_s(k).$$

Consequently, the maximum reached state by the system will be given by:

$$x^{max}(k) = \hat{M} w^{max}(k). \tag{3.99}$$

3.4.3 Complexity Issues

The methodologies presented in this section consider both an upper bound to the state vector and the constraints written as a semimodule equation, *i.e.*,

$$D\bar{x} = E\bar{x},$$

so $\bar{x}(k) \in \mathsf{Im}\ \mathcal{M}$, if it is assumed the semimodule finitely generated. Two important methodologies were used to find the solution, the semimodule theory (Butkovic and Hegedus, 1984) and the Modified Alternating algorithm (Cuninghame-Green and Butkovic, 2003)(Gomes da Silva and Maia, 2014)..

As previously discussed, the first methodology has a double exponential complexity in relation to the length of the horizon, so the size of the semimodule and the computational time will grow double exponentially as the horizon grows. The second methodology has a pseudo-polynomial complexity, this fact means that the computational time to find the solution will grow polynomially in relation to the horizon. Therefore, the computational time and the computational memory cannot be feasible to solve some problems since they can be very large.

At this point, some algebraic tools that simplify the methodology in relation to computational time and memory can be used to find the solution to an important class of problems of practical interest. These tools are presented in the following section.

3.4.4 Algebraic Results to Compute the Just-in-Time Control in Infinite Horizon

Initially, consider the following theorem.

Theorem 3.4.1 *(Adapted from Gonçalves (2015)) The Maximal (A,B) Max-Plus Geometrically Invariant set* $\mathcal{X}^*$ *inside* $\mathcal{S} = \{x|Dx = Ex\}$ *is non empty if and only if the set* $\mathcal{V}(A,B,D,E) = \{(u,v,\lambda)|Av \oplus Bu = \lambda v$ *and* $Dv = Ev\}$ *is non empty, and the entries of* $\mathcal{V}(A,B,D,E)$ *are proper*[4].

Then it is possible to define the following control problem.

Definition 3.4.6 ***(Just-in-Time Control in an Infinite Horizon with Transientless Reference Model)*** *The Just-in-Time control problem in an infinite horizon can be defined as:*

$$\max \; (u(k)_{\forall k \succ 0})$$

subjected to:

$$x(k) = Ax(k-1) \oplus Bu(k) \tag{3.100}$$

$$x_m(k) = \lambda x_m(k-1) \tag{3.101}$$

$$Cx(k) \preceq Cx_m(k) \tag{3.102}$$

$$Dx(k) = Ex(k) \tag{3.103}$$

[4] That is, they have no ε element.

Remark: 3.4.8 *The control problem in Definition 3.4.6 requires finding the maximal system input dates* $u(k)$*,* $k \succeq 0$*, computing the maximum entries* u_i*,* $\forall i$*, of* $u(k)$*.*

Remark: 3.4.9 *Recall that, if the optimization problem (Definition 3.4.6) has a solution, there exists a non-empty (A,B) Max-Plus Geometrically Invariant Set* $\mathcal{X}^*$ *inside* $\mathcal{S} = \{x|Dx = Ex\}$*. So the set* $\mathcal{V}(A, B, D,E)$ *must be non-empty.*

Remark: 3.4.10 *If the optimization problem is solvable, then the set* $\mathcal{V}(A, B, D,E)$ *is non-empty.*

The generation of the set of constraints $\mathcal{X}^*$ is a significative computational challenge since the methods to generate semimodules have double exponential complexity (see Katz (2007)). However, Theorem 3.4.1 leads to an useful constraint semimodule compatible with the model and the constraints, which operates at a rate λ and has no transient behavior, with a relatively low computational cost. In this sense, consider the following definition:

Definition 3.4.7 (Model-Compatible Constraint Semimodule) *A model-compatible constraint set is given by* $\mathcal{R}_m(\lambda) = \{\exists u, v|Av \oplus Bu = \lambda v$ *and* $Dv = Ev\}$*.*

It is important to highlight that $\mathcal{R}_m(\lambda)$ is non empty if the control problem is solvable.

As mentioned previously, among all possible reference models a particular interesting one is a model that operates without transient behavior. In the same sense, in several situations, this behavior is also desirable for the system. So, consider the following assumption:

Assumption: 3.4.4 *It is assumed that $x(k)$ and $x_m(k)$, $\forall k \succeq 0$, belong to the model-compatible constraint set $\mathcal{R}_m(\lambda)$.*

The Remark 3.4.9 has motivated the seek for a feasible reference model trajectory inside the set $\mathcal{R}_m(\lambda)$. In this sense, it is necessary to find the smallest feasible λ and an appropriated initial condition.

Assumption: 3.4.5 ***(Feasible Transientless Reference Model With Largest Production Rate)*** *Hereafter, the feasible reference model is given by $Ax_m(k) \oplus Bu_m(k) = \lambda_{\min} x_m(k)$, being $\lambda_{\min} = \min\{\lambda | \mathcal{R}_m(\lambda) \neq \emptyset\}$.*

Remark: 3.4.11 *In order to compute $\lambda_{\min}$ see Gaubert and Sergeev (2013).*

The Modified Alternating Algorithm (Gomes da Silva and Maia, 2014) can be used to find the initial condition $x_m(0)$ in an efficient way.

Remark: 3.4.12 *The initial condition $x(0) = x^{up} = \max\{v | v \in \mathcal{R}_m(\lambda)\}$, such that $(Cx(0)) \preceq Cx_m(0)$, belongs to $\mathcal{X}^*$ and consequently belongs to $\mathcal{V}(A,B,D,E)$. As a result, the initial condition will be such that $Ax(0) \preceq \lambda x(0)$.*

Proposition 3.4.3 *The state of the system, at a k^{th} transition, must belong to the set $\mathcal{Z}(\lambda) = \{x | Cx \preceq \lambda^k (Cx_m(0))$ and $x \in \mathcal{R}_m(\lambda)\}$, which has a biggest solution denoted as $\lambda^k x^{up}$, whose x^{up} is given by $\max\{v | v \preceq C \diagdown\!\!\!\circ (Cx_m(0))$ and $v \in \mathcal{R}_m(\lambda)\}$.*

Proof: According to Assumption 3.4.4 the state must evolves as $x(k) = \lambda^k x(0)$ and must respect $Cx(k) \preceq \lambda^k Cx(0)$. Remark 3.4.12 ensures that $x(0) = x^{up}$ and Theorem 3.4.1 ensures that a proper value to $x(0)$ exists. Then $x(k) = \lambda^k x^{up}$.

The fastest system behavior is particularly useful. It can be achieved by computing $\max\{v | v \preceq C \diagdown\!\!\!\circ (Cx_m(0))$ and $v \in \mathcal{R}_m(\lambda_{\min})\}$. By Assumption 3.4.4

and Remark 3.4.3, the equations $Av \oplus Bu = \lambda v$ and $Dv = Ev$, can be rewritten as a max-plus linear equation in the following way:

$$\begin{bmatrix} A & B \\ D & [\varepsilon] \end{bmatrix} \begin{bmatrix} v \\ u \end{bmatrix} = \begin{bmatrix} \lambda & [\varepsilon] \\ E & [\varepsilon] \end{bmatrix} \begin{bmatrix} v \\ u \end{bmatrix}.$$

By the theory of semimodules $[v^T u^T]^T \in \mathsf{Im}Q$, which implies that we can write $v = Q_v z$ and $u = Q_u z$. So we can always compute the biggest initial condition as $x^{up} = Q_v z$, for a certain z. Since $Cx^{up} \preceq Cx_m(0)$, we have:

$$x^{up} = Q_v[(CQ_v \backslash\!\!\!\!\circ Cx_m(0)] \tag{3.104}$$

Proposition 3.4.4 *The proposed control problem is solvable and the upper bound for the inputs is given by* $u^{up}(k) = \lambda^k (C \backslash\!\!\!\!\circ x^{up})$.

Proof: By Proposition 3.4.3 the biggest state of the system at a k^{th} *transition is given by* $x(k) = \lambda^k x^{up}$. *Since* $Ax(k-1) \oplus Bu(k) = x(k)$, *Residuation theory ensures that the biggest* $u^{up}(k)$ *is given by* $\lambda^k(C \backslash\!\!\!\!\circ x^{up})$.

Recall that $x^{up} = \max\{v | v \preceq C \backslash\!\!\!\!\circ (Cx_m(0))$ and $v \in \mathcal{R}_m(\lambda_{\min})\}$ can be computed using the modified alternating algorithm, considering $v_0 = C \backslash\!\!\!\!\circ (Cx_m(0))$ and $u_0 = B \backslash\!\!\!\!\circ (\lambda_{min} v_0)$ the initial condition for the algorithm.

Remark: 3.4.13 (Methods of Solution and Computational complexity) *There are three methodologies to solve the infinite horizon control problems presented in this section. The first one considers the finite control horizon big enough (methodology presented in a previous section), i.e., if the control horizon is finite, but it is big enough, it can be considered an infinite horizon for some applications and it can be used to solve the examples in the following section. The second one is the methodology that aims to finding a solution to the semimodule equa-*

tion in order to solve the problem, and the third methodology uses the modified alternating algorithm to find the solution to the control problem.

As mentioned previously, the computational complexity of the approach based on semimodule generation depends on the fact that the solution to $Ax = Bx$, *where* $A,B \in \mathbb{Z}_{max}^n$ *are row vectors, are hyperplanes finitely generated. Therefore, in the worst case, to compute the solution to all rows, the algorithm is double exponential (Butkovic and Hegedus, 1984)(Katz, 2007). However, for special types of semimodules, as the one presented in Corollary 3.3.1, in which the generator of the semimodule belongs to the image of a matrix generated by the Kleene star theorem, the computational complexity is polynomial.*

Other issue is the computational memory, the approach of semimodule generation considers a matrix equation row by row. Firstly, the methodology considers the solution to the first row and uses this solution to find the solution to the second row and so on. Then, for each row of matrix equation the methodology needs to save a matrix, that can have a big size in simple applications (the methodology is presented in Section 2.6.1). Therefore, for some applications the computational memory necessary to find a solution to the control problem can be impracticable, i.e, the methodology can exceed the memory of the computer.

So, the first method and the second method, mentioned in this remark, find the set of all solutions to the control problems, but in general they need an expressive computational effort to find the optimal solution.

On the other hand, if just one solution is necessary, the approach based on the "Modified Alternating" method has pseudo-polynomial complexity for several important situations and works quite well in practice, since in many cases the input size is suitable.

The Modified Alternating algorithm is very simple and can deal with big matrices without exceeding the memory stack size since it uses mainly the Theory of Residuation.

3.4.5 Numerical Examples

A Transportation Network

The transportation system used in this example are described in detail in Katz (2007) and de Vries et al. (1998) and it is represented in Figure 3.3. To model a transportation network as a Timed Event Graph, it is assumed that the transitions are controllable and the transitions $x_i(k), \forall i$, denote the date of the k^{th} departure of trains leaving stations.

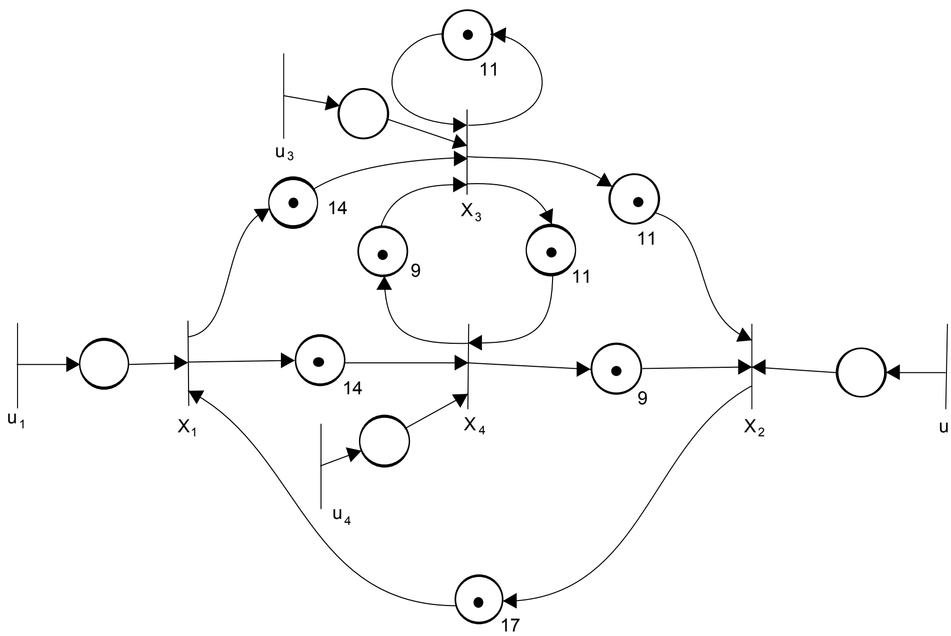


Figure 3.3: The Model to a Transportation System

The control problem objectives presented in Katz (2007) are the requirement that the time between two consecutive train departure does not exceed a given limit and the passengers waiting time coming from a given station and going to another station must not exceed a given limit. To this end, the system was described as in Equation 3.1 and the constraint obtained from Katz (2007) is given by:

$$E_r x(k+1) \preceq x(k), \tag{3.105}$$

with

$$E_r = \begin{bmatrix} -15 & \varepsilon & -18 & -18 \\ -21 & -15 & \varepsilon & \varepsilon \\ \varepsilon & -15 & -15 & -15 \\ \varepsilon & -13 & -13 & -15 \end{bmatrix}.$$

The constraint can be written as:

$$\begin{bmatrix} E_r & \varepsilon \end{bmatrix} \begin{bmatrix} x(k+1) \\ x(k) \end{bmatrix} \preceq \begin{bmatrix} \varepsilon & I \end{bmatrix} \begin{bmatrix} x(k+1) \\ x(k) \end{bmatrix}.$$

Using the previously methodology and the algorithm presented in Katz (2007), the maximal (A,B)-invariant set $\mathcal{X}^* \in \mathsf{Im}\ \mathcal{M}$ and $\mathcal{M}$ is given by:

$$\mathcal{M} = \begin{bmatrix} 18 & 17 & 17 & 17 & 17 \\ 15 & 14 & 15 & 15 & 15 \\ 18 & 17 & 18 & 18 & 18 \\ 19 & 18 & 19 & 19 & 19 \\ 3 & 2 & 2 & 4 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 4 & 3 & 4 & 4 & 4 \\ 5 & 4 & 6 & 5 & 5 \end{bmatrix}.$$

This numerical example will be solved considering the fastest behavior of the reference model $x_m(k) = Ax_m(k-1) \oplus Bu_m(k)$, given when the input vector $u_m(k) = [\varepsilon]$, $\forall k \succeq 0$. Then, arbitrarily choosing

$$x_m(0) = \begin{bmatrix} 20 & 20 & 20 & 20 & 6 & 6 & 6 & 6 \end{bmatrix}^T,$$

the system reference trajectory for $k = 2$ until $k = 6$ is given by:

$$x_m(2) = \begin{bmatrix} 37 & 31 & 34 & 34 \end{bmatrix},$$

$$x_m(3) = \begin{bmatrix} 48 & 45 & 51 & 51 \end{bmatrix},$$

$$x_m(4) = \begin{bmatrix} 62 & 62 & 62 & 62 \end{bmatrix},$$

$$x_m(5) = \begin{bmatrix} 79 & 73 & 76 & 76 \end{bmatrix},$$

$$x_m(6) = \begin{bmatrix} 90 & 87 & 93 & 93 \end{bmatrix}.$$

Solving the Equation 3.78, the system initial condition is given by:

$$x(0) = \begin{bmatrix} 19 & 16 & 19 & 20 & 5 & 2 & 5 & 6 \end{bmatrix}^T,$$

Then, solving the equation $Ax(k-1) \oplus Bu(k) = \mathcal{M}v_{(k)}$, considering the upper value to $v_{(k)}$ equal to $v_{(k)}^{up} = (C\mathcal{M} \diamond\!\!\!\setminus Cx_m(k))$, by using the Modified Alternating Algorithm for $k = 2$ until $k = 6$, the maximal input dates and state, that respect the deadline dates, are given by:

$$u_{max}(2) = x(2) = \begin{bmatrix} 33 & 30 & 33 & 34 \end{bmatrix}^T,$$

$$u_{max}(3) = x(3) = \begin{bmatrix} 48 & 45 & 48 & 49 \end{bmatrix}^T,$$

$$u_{max}(4) = x(4) = \begin{bmatrix} 61 & 58 & 61 & 62 \end{bmatrix}^T,$$

$$u_{max}(5) = x(5) = \begin{bmatrix} 75 & 72 & 75 & 76 \end{bmatrix}^T,$$

$$u_{max}(6) = x(6) = \begin{bmatrix} 90 & 87 & 90 & 91 \end{bmatrix}^T.$$

However, if no control is applied, the following trajectory starting from the same initial state can be obtained, considering all the inputs at the starting control date:

$$x(2) = \begin{bmatrix} 32 & 29 & 31 & 31 \end{bmatrix}^T,$$

$$x(3) = \begin{bmatrix} 46 & 42 & 46 & 46 \end{bmatrix}^T,$$

$$x(4) = \begin{bmatrix} 59 & 57 & 60 & 60 \end{bmatrix}^T,$$

$$x(5) = \begin{bmatrix} 74 & 71 & 73 & 73 \end{bmatrix}^T,$$

$$x(6) = \begin{bmatrix} 88 & 84 & 88 & 88 \end{bmatrix}^T.$$

Therefore, the applied control delays as much as possible the input dates and it makes the system produce the output dates closer to the reference demand.

Unlike Katz (2007), in this thesis, it is applied the Just-in-time control policy, the firing dates of state transitions are the maximum possible and respect the problem constraints, *i.e.*, the dates are the maximum departure time to a train to leave a station in order to respect the constraints. The obtained system states from Katz (2007) are:

$$x(2) = \begin{bmatrix} 32 & 29 & 32 & 33 \end{bmatrix}^T,$$

$$x(3) = \begin{bmatrix} 46 & 43 & 46 & 47 \end{bmatrix}^T,$$

$$x(4) = \begin{bmatrix} 60 & 57 & 60 & 61 \end{bmatrix}^T,$$

$$x(5) = \begin{bmatrix} 74 & 71 & 74 & 75 \end{bmatrix}^T,$$

considering the system initial state:

$$x(0) = \begin{bmatrix} 4 & 0 & 4 & 5 \end{bmatrix}^T.$$

Remark: 3.4.14 (Computational Time) *The obtained solution by the proposed methods are the same, as expected, but the processing times are quite different. It takes the semimodule approach in finite horizon, considering the horizon* $N_p = 6$*, 1.637 seconds (using the Kleene star operator), it takes the semimodule approach in infinite horizon 1 minute and 31.892 seconds (finding the maximal*

(A,B)-invariant set), while it takes the one based on the Alternating Method 0.230 seconds (average value for 10 experiments), for a desktop computer, Intel Core i5 2.53GHz, 4GB RAM, Windows 10, 64bits, Cache 3.932 GB.

A Small Manufacturing System

This example is presented in order to illustrate the computational complexity of finding the maximum (A,B)-invariant set $\mathcal{X}^*$. A simple manufacturing system under some constraints is considered but, because of the large size of the matrices, $\mathcal{X}^*$ cannot be computed. However, the algebraic properties to solve this problem with lower computational cost can be used. This example is interesting because the production rate λ of system can be determined. The fastest behavior ($\lambda_{\min}$) is considered.

The example presented in this section is described in Gomes da Silva and Maia (2015b). Consider a small manufacturing system composed by three different areas, an input area and two units processor areas. The TEG model for the manufacturing system is presented in Figure 3.4. The input area has three input transitions (u_1, u_2 and u_3). The items placed in u_1, u_2 and u_3 take 1, 3 and 4 time units, respectively, to arrive in the first unit processor. This unit processor has unitary capacity and it begins a new process at least six time units after the previous process.

In the first unit processor the system has an input transition for a complement item enters the unit. This complement item enters by transition u_4 and takes two time units to be ready. The first unit processor also has the main process. The three items from input area are processed together (Place P_4) and takes five time units to get ready. The main process in unit processor one has unitary capacity.

The two pieces are sent to the second unit processor to be processed together.

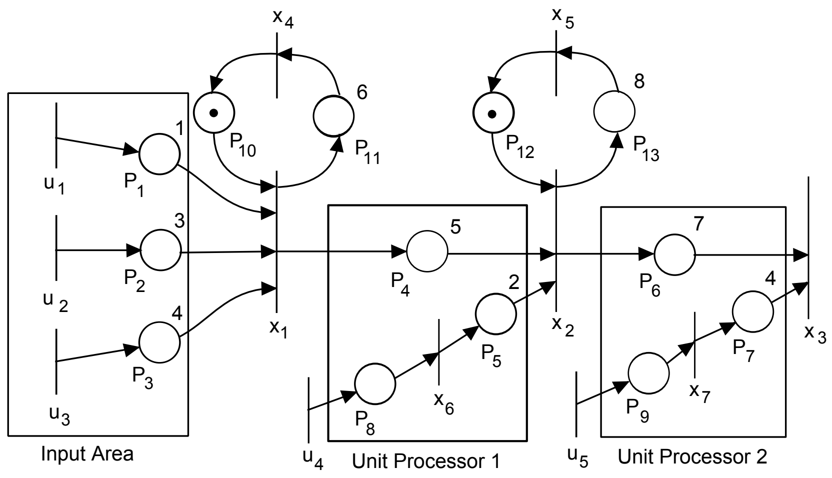


Figure 3.4: A Small Manufacturing System

This main processor also has unitary capacity and takes seven time units to get ready. When the product is ready, a box enters the system by input transition u_5 and takes four time units to get ready (Place P_7).

It is desirable that the evolution of the system state happens in accordance with some performance and constraints and, to this end, some constraints will be applied to the system state. The first constraint is that the k^{th} firing date of transition x_3 needs to happen before the k^{th} firing date of transition x_1 plus eighteen time units and the k^{th} firing date of transition x_2 plus ten time units. The transition x_4 needs to fire before transition x_2 plus two time units and transition x_5 fires before transition x_3 plus three time units. This last two constraints help to ensure the unitary capacity of processors.

In order to give a better performance some constraints will be applied to the inputs, because some input transitions can fire delayed. The pieces necessary to begin the production arrive by inputs u_1, u_2 and u_3, so the input in u_4 and u_5 can be delayed. The k^{th} input firing date of transition u_4 needs to happen after seven time units after the k^{th} firing of input transition u_3 and the input u_5 after five time units after input u_4. The input $u_1(k)$ needs to fire three time units after $u_3(k)$ and the firing date of input $u_2(k)$ after one time unit after $u_3(k)$. Therefore, the state constraints and input constraints can be written mathematically as the following equations.

$$\begin{aligned}
&x_3(k) \preceq 18x_1(k) \oplus 10x_2(k),\\
&x_4(k) \preceq 2x_2(k),\\
&x_5(k) \preceq 3x_3(k),\\
&u_4(k) = 7u_3(k),\\
&u_5(k) = 5u_4(k),\\
&u_1(k) = 3u_3(k),\\
&u_2(k) = 1u_3(k).
\end{aligned}$$

This constraints can be easily written as $\bar{D}\bar{x} = \bar{E}\bar{x}$. Considering the open-loop Just-in-Time Control problem in a finite horizon as a particular case of the infinite horizon control problem, in this example the results only to open-loop Just-in-Time Control problem in infinite horizon are computed. Using the presented methodologies in previous sections, the system was described by max-plus algebra as Equation 3.100, however, as previously mentioned, the maximal (A,B)-invariant set $\mathcal{X}^*$ cannot be computed because of the large size of the system matrices. On average, 27 minutes were spent until exceed the stack size of memory [5].

In order to develop the fastest behavior the rate $\lambda_{\min} = 8$, and by choosing arbitrarily an upper bound to initial condition of reference model ($\bar{x}_m^{up}(0)$) equal to

$$\bar{x}_m^{up}(0) = \begin{bmatrix} 21 & 21 & 21 & 21 & 21 & 21 & 21 & 3 & 1 & 0 & 7 & 12 \end{bmatrix},$$

the initial state of the system inside $\mathcal{Z}$, obtained from Equation 3.104, is equal to:

$$\bar{x}(0) = \begin{bmatrix} 3 & 11 & 21 & 13 & 21 & 6 & 14 & 3 & 1 & 0 & 7 & 12 \end{bmatrix},$$

then, for $\bar{x}_m(0)$ be inside $\mathcal{R}_m$, in this example it is considered the reference model initial

[5] For a Desktop Computer, Intel Core i5 2.53GHz, 4GB RAM, Windows 10, 64bits, Cache 3.932 GB.

condition $\bar{x}_m(0) = \bar{x}(0)$.

Using the previously presented methodology, the input dates, for $k = 1$ until $k = 6$, to solve the control problem are given by:

$$u(1) = \begin{bmatrix} 16 & 14 & 13 & 20 & 25 \end{bmatrix},$$

$$u(2) = \begin{bmatrix} 24 & 22 & 21 & 28 & 33 \end{bmatrix},$$

$$u(3) = \begin{bmatrix} 32 & 30 & 29 & 36 & 41 \end{bmatrix},$$

$$u(4) = \begin{bmatrix} 40 & 38 & 37 & 44 & 49 \end{bmatrix},$$

$$u(5) = \begin{bmatrix} 48 & 46 & 45 & 52 & 57 \end{bmatrix},$$

$$u(6) = \begin{bmatrix} 56 & 54 & 53 & 60 & 65 \end{bmatrix}.$$

So, the maximum reached states by the system to $1 \preceq k \preceq 6$ are:

$$x^{max}(1) = \begin{bmatrix} 17 & 22 & 29 & 23 & 30 & 20 & 25 \end{bmatrix},$$

$$x^{max}(2) = \begin{bmatrix} 25 & 30 & 37 & 31 & 38 & 28 & 33 \end{bmatrix},$$

$$x^{max}(3) = \begin{bmatrix} 33 & 38 & 45 & 39 & 46 & 36 & 41 \end{bmatrix},$$

$$x^{max}(4) = \begin{bmatrix} 41 & 46 & 53 & 47 & 54 & 44 & 49 \end{bmatrix},$$

$$x^{max}(5) = \begin{bmatrix} 49 & 54 & 61 & 55 & 62 & 52 & 57 \end{bmatrix}.$$

$$x^{max}(6) = \begin{bmatrix} 57 & 62 & 69 & 63 & 70 & 60 & 65 \end{bmatrix}.$$

The output dates are

$$y(k) = \begin{bmatrix} 29 & 37 & 45 & 53 & 61 & 69 \end{bmatrix},$$

and the desirable trajectory to the output dates is

$$y_m(k) = \begin{bmatrix} 29 & 37 & 45 & 53 & 61 & 69 \end{bmatrix}.$$

Analyzing the vectors $x^{max}(k)$, the controlled system evolves, in the 50 steps simulated, respecting all the constraints imposed. Due to the system periodicity, the result is guaranteed to all k (see Baccelli et al. (1992)).

However, if no control is applied, the following output trajectory, starting from the same initial state, can be obtained considering all inputs available at the initial prediction horizon date:

$$y(k) = \begin{bmatrix} 28 & 36 & 44 & 52 & 60 & 68 \end{bmatrix}.$$

Therefore, as expected, the applied control delays as much as possible the input dates and it makes the system produce output dates closer to the reference demand.

Remark: 3.4.15 (Computational Time) *The obtained solution by the proposed methods are the same, as expected, but the processing times are quite different. It takes the semimodule approach in finite horizon, considering arbitrarily the horizon $N_p = 6$, 23.007 seconds, it takes the semimodule approach in infinite horizon 17.994 seconds, while it takes the one based on the Alternating Method 0.129 seconds (average value for 10 experiments), for a desktop computer, Intel Core i5 2.53GHz, 4GB RAM, Windows 10, 64bits, Cache 3.932 GB.*

3.4.6 Conclusion

This section presented the open-loop Just-in-Time control in infinite horizon for max-plus linear systems under some constraints (non convex constraints in conventional algebra). In order to solve the problems, general methodologies based on (A,B)-invariant sets, the Residuation Theory and the Theory of Semimodules were proposed. However, due to the computational complexity of general methods, algebraic properties on max-plus algebra were used to solve an important class of problems of practical interest. The necessary and sufficient conditions to solve the problems were presented and discussed. Numerical examples illustrated the methodologies and the applicability of the results. It is important to remark that if the horizon is bounded, the methodologies presented in this section can also solve the problems in a finite horizon and, using both methods, the results are the same.

3.5 Feedback Control Problem

3.5.1 Introduction

The idea of feedback in control system theory is to use any available information about the system behavior or system parameter to continuously adjust the control input along with an input reference. For example, in a manufacturing system, if a machine has unitary capacity, the feedback is important to inform the system when the machine is idle.

Using feedback control, the desirable behavior of a system becomes less sensitive to disturbances and errors. A system output can track automatically a desirable reference signal by seeking to minimize the difference between the output date and the reference date. On the other hand, complex equipment can be necessary to monitor the desirable information of the system.

In Discrete Event Systems one concept of stability is related to the number of tokens in each internal place as given in the following definition.

Definition 3.5.1 (Stability) *A TEG is said to be stable if, for any input, the number of tokens in places remains bounded.*

Remark: 3.5.1 *Though there is other definition about stability in literature, the work in this thesis is interested only in Definition 3.5.1 about stability of Discrete Event Systems.*

The advantage of feedback control is to ensure that the controlled system is stable. To illustrate this issue, consider the TEG in Figure 3.5. In this figure, the holding time of a token in place P_3 is three time units and the holding time in place P_5 is six time units, this fact can make the number of tokens in place P_4 grows unbounded since a token can be placed in P_4 each three time units and one token leaves P_4 after at least six time units.

In other hand, if the same net with feedback is considered, as presented in Figure 3.6, a token will enter place P_3 at least after six time units due to the holding time in the feedback.

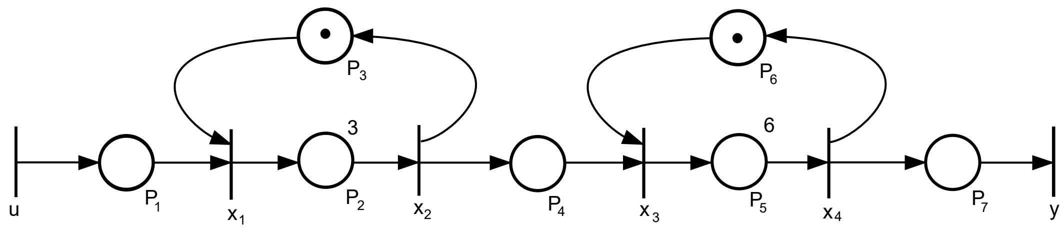


Figure 3.5: Timed Event Graph Without Feedback

Therefore, it is not possible to accumulate tokens in place P_4, making the system stable (for more details see Maia (2003)).

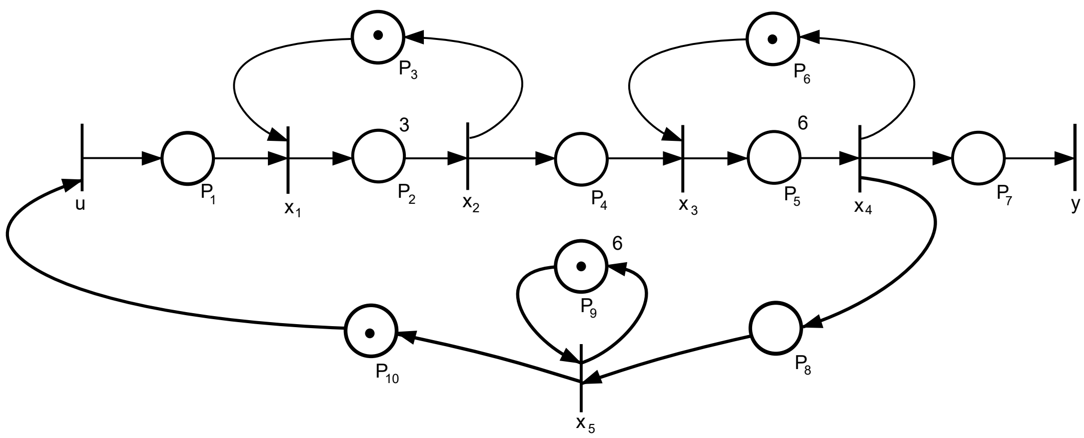


Figure 3.6: Timed Event Graph With Feedback

Other advantage of feedback control is presented in the following.

Definition 3.5.2 (Structural Controllability) *(Maia, 2003) An event graph is structurally controllable if there exists a path for all internal transition from at least one input transition.*

Definition 3.5.3 (Structural Observability) *(Maia, 2003) An event graph is structurally observable if there exists a path from all internal transition for at least one output transition.*

Theorem 3.5.1 (Stabilization by Feedback) *(Baccelli et al., 1992) All event graph structurally controllable and observable can be stabilized by the output feedback.*

The proof for Theorem 3.5.1 is based on that all event graph structurally controllable and observable can be stabilized if one connection by feedback is established from all outputs for all inputs. This procedure ensures that the event graph in closed-loop is strongly connected.

Based on the previously presented in this section, feedback control is addressed in this thesis. Sometimes, it is desirable a minimal restrictive feedback matrix, *i.e.*, it is sought the smallest causal feedback matrix. However, as deadline dates are considered and the desirable output dates are the greatest dates smaller than or equal to the deadline dates, the control objective is computing the maximal feedback matrix so that $Fx(k-1)$ is the maximum. In this section, necessary and sufficient conditions to find a feedback matrix are shown. The feedback control problem formulation is presented next.

Assumption: 3.5.1 *To perform the feedback control, canonical initial conditions are assumed, that is* $x(k) = [\varepsilon]$, $\forall k \prec 0$.

Definition 3.5.4 (Feedback Control Problem Formulation) *The feedback control problem can be defined as:*

$$\max\ (Fx(k)_{\forall k \succ 0})$$

subjected to:

$$x(k) = Ax(k-1) \oplus Bu(k) \tag{3.106}$$

$$Ax_m(k) \oplus Bu_m(k+1) = \lambda x_m(k) \tag{3.107}$$

$$Cx(k) \preceq Cx_m(k) \tag{3.108}$$

$$Dx(k) = Ex(k) \tag{3.109}$$

3.5.2 Performing the Feedback Control Problem

In the previous subsection an open-loop version of this problem was approached, ensuring that exists u^{up} and x^{up} such that $Ax^{up} \oplus Bu^{up} = \lambda x^{up}$, being $u^{up} = \lambda^k (B \backslash\!\!\!\!\!\!\diagdown x^{up})$. In this context, we have the following results.

Proposition 3.5.1 (Existence of Solution) *A feedback matrix F always exists if any vector $v \in \mathcal{Z}(\lambda)$, is such that $\mathcal{Z}(\lambda) = \{x | Cx \preceq \lambda^k (Cx_m(0))$ and $x \in \mathcal{R}_m(\lambda)\}$. One matrix F is given by:*

$$F = \frac{uv^T}{v^T v}. \tag{3.110}$$

Proof: The vector v must be inside $\mathcal{R}_m(\lambda)$, then v is such that

$$Av \oplus Bu = \lambda v,$$

so the maximal input u, called u_{max}, is given by Residuation Theory as $u_{max} = B \backslash (\lambda v)$. Since $x \in R_m(\lambda)$, the trajectory must evolve as $x(k) = \lambda^k v$. So $u_{max}(2) = \lambda u_{max} \Rightarrow u_{max}(k) = \lambda^{(k-1)} u_{max}$. Since the feedback matrix is given by

$$F = \frac{uv^T}{v^T v}, \tag{3.111}$$

we can show that $u_{max} = Fv$. Making $x(0) = v$ and $u(1) = u_{max}$, it is possible to show that $u(k) = \lambda^k u \Rightarrow u(k) = \lambda^k F x(0) \Rightarrow u(k) = Fx(k-1)$.

As a result, if a feedback matrix to the control problem exists, there exists the greatest one given by the next proposition.

Proposition 3.5.2 (Greatest Feedback Matrix) *If a solution exists, there is the greatest one that respect the deadline dates given by*

$$F^{max} = B \backslash [\lambda v] \not/ v \tag{3.112}$$

being $v \in \mathcal{Z}(\lambda)$.

Proof: It is sought a trajectory inside $\mathcal{Z}(\lambda) = \{x | Cx \preceq \lambda^k (Cx_m(0))$ and $x \in \mathcal{R}_m(\lambda)\}$, so if $x(0) \in \mathcal{Z}(\lambda)$ it is possible to make $x(k) \in \mathcal{Z}(\lambda), \forall k$. Therefore, since $x(k) = \lambda^k v$, for any k, the equation $Ax(k) \oplus Bu(k+1) = x(k+1)$ implies by Proposition 3.5.1 that there exists F such that $Av \oplus BFv = \lambda v$ so, by the Residuation Theory,

$$F^{max} = B \backslash [\lambda v] \not/ v.$$

Realization Issues About Feedback Control

Some results presented in this context can be found in Baccelli et al. (1992) and Maia et al. (2013). However, in Maia et al. (2013) the authors are interested in the minimal feedback matrix, *i.e.*, in a feedback causal matrix that increases as least as possible the eigenvalue of the closed-loop matrix $A_{cl} = (A \oplus BF^{max})$. This work is interested in the maximum feedback matrix in order to comply with deadline dates, and the delay caused by the feedback matrix is desired. Even though, the greatest feedback matrix F^{max} found in Proposition 3.5.2 can be non realizable (the matrix can be non causal). Considering the closed-loop matrix $A_{cl} = (A \oplus BF^{max})$, it is possible to find a realizable matrix (causal matrix) F_c if A_{cl} is irreducible and this matrix has an eigenvalue bigger than 0.

If the problem has a solution as defined in Proposition 3.5.1, there exists a non causal control law $u_{nc}(k) = F_{nc}x(k-1)$, in which

$$x(k) = (A \oplus BF_{nc})x(k-1). \tag{3.113}$$

Using the Equation 3.113 for an initial condition $x(0)$,

$$u_{nc}(k) = F_{nc}(A \oplus BF_{nc})^{m-1}x(k-m), \forall k \succeq m. \tag{3.114}$$

The controlled inputs for $k \prec m$ can be considered as $u(k) = \lambda^k u_m(0)$. This means that all inputs for $k \prec m$ occur before starting the feedback control. So, as it is desirable that $x(k) = Ax(k-1) \oplus Bu(k) \preceq \lambda^k x_m(0)$ these implies that $u(k) = \lambda^k u_m(0)$, for $k \prec m$.

From spectral theory of matrices, it is possible to show for a irreducible matrix H, $H^{k+c} = \gamma^c H^k$, $\forall k \succeq p$, for a large p, in which γ is the eigenvalue and c the cyclicity of the matrix H. So $A_{cl} \succeq I$ and irreducible with eigenvalue greater than 0. Therefore, it is always possible to find a causal matrix

$$F_c = F_{nc}(A \oplus BF_{nc})^{m-1} \tag{3.115}$$

by the necessary increment of m.

Remark: 3.5.2 *An other way to find a causal matrix was presented in Gonçalves et al. (2014)*

using a predictor algorithm.

3.5.3 Numerical Examples

A Small Manufacturing System

The current example is the same previously presented in Subsection 3.4.5 and modeled by a Petri Net in Figure 3.7.

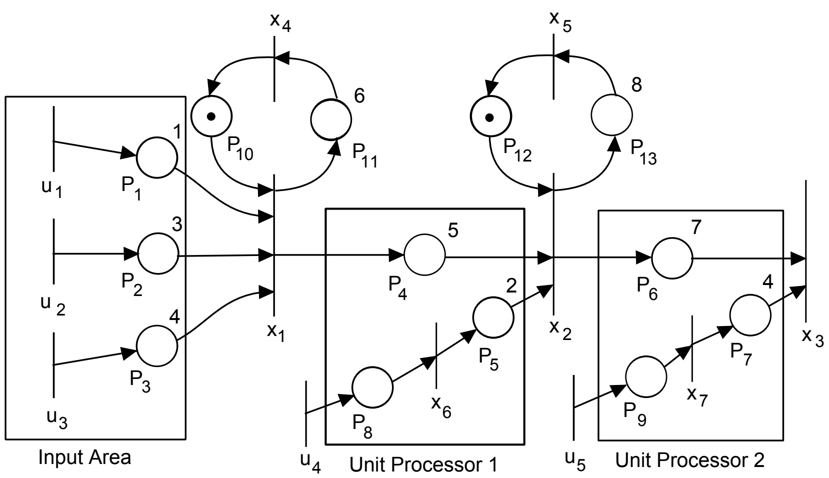


Figure 3.7: A Small Manufacturing System

Concerning the feedback control and using the presented results, the maximal feedback control is given by:

$$F^{max} = \begin{bmatrix} 13 & 11 & 10 & 17 & 22 \\ 5 & 3 & 2 & 9 & 14 \\ -5 & -7 & -8 & -1 & 4 \\ 3 & 1 & 0 & 7 & 12 \\ -5 & -7 & -8 & -1 & 4 \\ 10 & 8 & 7 & 14 & 19 \\ 2 & 0 & -1 & 6 & 11 \\ 13 & 11 & 10 & 17 & 22 \\ 15 & 13 & 12 & 19 & 24 \\ 16 & 14 & 13 & 20 & 25 \\ 9 & 7 & 6 & 13 & 18 \\ 4 & 2 & 1 & 8 & 13 \end{bmatrix}^T .$$

Therefore $F_{nc} = F^{max}$ since F^{max} is non causal. However, using the Equation 3.115, the maximal causal feedback matrix is given by:

$$F_c = \begin{bmatrix} 27 & 25 & 24 & 31 & 36 \\ 19 & 17 & 16 & 23 & 28 \\ 9 & 7 & 6 & 13 & 18 \\ 17 & 15 & 14 & 21 & 26 \\ 9 & 7 & 6 & 13 & 18 \\ 24 & 22 & 21 & 28 & 33 \\ 16 & 14 & 13 & 20 & 25 \\ 27 & 25 & 24 & 31 & 36 \\ 29 & 27 & 26 & 33 & 38 \\ 30 & 28 & 27 & 34 & 39 \\ 23 & 21 & 20 & 27 & 32 \\ 18 & 16 & 15 & 22 & 27 \end{bmatrix}^T ,$$

for m equal to 2. Therefore, the causal feedback control law will be $u(k) = F_c x(k-2)$, the inputs $u(k) = \lambda^k u_m(0)$, $\forall k \prec 2$. The Just-in-Time policy of control is applied, so the reached states using the matrix F_c are:

$$x^{max}(1) = \begin{bmatrix} 17 & 22 & 29 & 23 & 30 & 20 & 25 \end{bmatrix},$$

$$x^{max}(2) = \begin{bmatrix} 25 & 30 & 37 & 31 & 38 & 28 & 33 \end{bmatrix},$$

$$x^{max}(3) = \begin{bmatrix} 33 & 38 & 45 & 39 & 46 & 36 & 41 \end{bmatrix},$$

$$x^{max}(4) = \begin{bmatrix} 41 & 46 & 53 & 47 & 54 & 44 & 49 \end{bmatrix},$$

$$x^{max}(5) = \begin{bmatrix} 49 & 54 & 61 & 55 & 62 & 52 & 57 \end{bmatrix}.$$

$$x^{max}(6) = \begin{bmatrix} 57 & 62 & 69 & 63 & 70 & 60 & 65 \end{bmatrix}.$$

The output dates are

$$y(k) = \begin{bmatrix} 29 & 37 & 45 & 53 & 61 & 69 \end{bmatrix},$$

and the desirable trajectory of the output dates is

$$y_m(k) = \begin{bmatrix} 29 & 37 & 45 & 53 & 61 & 69 \end{bmatrix}.$$

As expected, these results are the same as those previously presented in this example for $k \succeq 2$. With the feedback control, the controlled system evolves, in the 50 steps simulated, respecting all the constraints imposed. Again, due to the system periodicity, the result is guaranteed for all k (see Baccelli et al. (1992)).

A Transportation Network

This example was also previously presented in Subsection 3.4.5, described in detail in Katz (2007) and de Vries et al. (1998), and represented in Figure 3.8.

The control problem objectives presented in Katz (2007) are the time between two consecutive departure trains must not exceed a given limit and the passengers waiting time coming

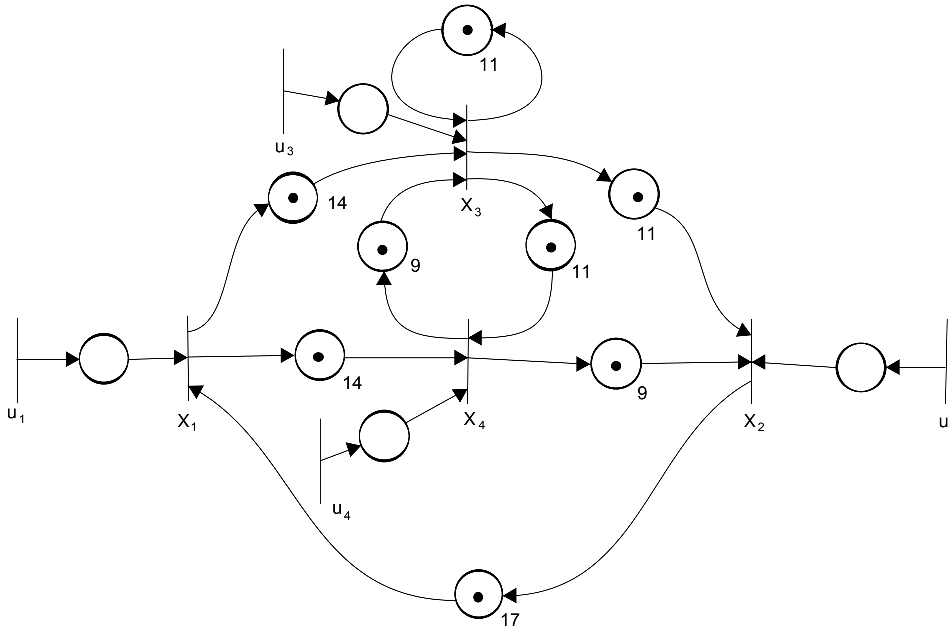


Figure 3.8: The Model to Transportation System

from a given station and going to another station must not exceed a given limit. To this end, the system was described as in Equation 3.1 and the constraint obtained from Katz (2007) is given by:

$$E_r x_m(k) \preceq x_m(k-1), \tag{3.116}$$

with

$$E_r = \begin{bmatrix} -15 & \varepsilon & -18 & -18 \\ -21 & -15 & \varepsilon & \varepsilon \\ \varepsilon & -15 & -15 & -15 \\ \varepsilon & -13 & -13 & -15 \end{bmatrix}.$$

Considering $x_m(0) = v \Rightarrow x_m(k) = \lambda^k v$. The constraint $E_r x_m(k) \preceq x_m(k-1)$ is equivalent to $E_r \lambda v \preceq v \Rightarrow E_r \lambda v \oplus v = v \Rightarrow (\lambda E_r \oplus I)v = v \Rightarrow Dv = Ev$.

Concerning the feedback control problem, the matrix F^{max} for the expanded state vector $x^e = [x(k-1)\ x(k)]^T$, is given by:

$$F^{max} = \begin{bmatrix} 14 & 17 & 14 & 13 & 28 & 31 & 28 & 26 \\ 11 & 14 & 11 & 10 & 25 & 28 & 25 & 23 \\ 14 & 17 & 14 & 13 & 28 & 31 & 28 & 26 \\ 14 & 17 & 14 & 13 & 28 & 31 & 28 & 26 \end{bmatrix}.$$

This result is similar to the greatest feedback matrix found in Maia et al. (2011b) using the super eigenvalue methodology for the same example. However, it is different from the one obtained by Katz (2007) because it is not interested in Just-in-Time control. The matrix obtained by Katz (2007) is

$$F = \begin{bmatrix} 14 & 14 & 14 & 13 & 14 & 14 & 14 & 14 \\ 11 & 14 & 11 & 10 & 14 & 14 & 14 & 14 \\ 14 & 14 & 14 & 13 & 14 & 14 & 14 & 14 \\ 14 & 14 & 14 & 14 & 14 & 14 & 14 & 14 \end{bmatrix}.$$

As expected, the input dates and internal dates are the same as previously presented. Unlike Katz (2007), the controller is found in a simpler way and, in this work, the Just-in-time control policy is applied, the firing dates of state transitions are the maximum possible and respect the problem constraints, *i.e.*, the dates are the maximum departure time for a train to leave a station in order to respect the constraints.

$$u_{max}(2) = x(2) = \begin{bmatrix} 33 & 30 & 33 & 34 \end{bmatrix}^T,$$

$$u_{max}(3) = x(3) = \begin{bmatrix} 48 & 45 & 48 & 49 \end{bmatrix}^T,$$

$$u_{max}(4) = x(4) = \begin{bmatrix} 61 & 58 & 61 & 62 \end{bmatrix}^T,$$

$$u_{max}(5) = x(5) = \begin{bmatrix} 75 & 72 & 75 & 76 \end{bmatrix}^T,$$

$$u_{max}(6) = x(6) = \begin{bmatrix} 90 & 87 & 90 & 91 \end{bmatrix}^T.$$

Small Manufacturing System II

Consider the manufacturing system with three machines (M_1, M_2 and M_3) endowed with unitary capacity in Figure 3.9. This system was originally presented in Maia et al. (2005). The raw materials are processed by machines M_1 and M_2 whose inputs are given by u_1 and u_2, respectively. The product of these two machines are grouped and processed by machine M_3, resulting in a final product with the date of deliver given by y.

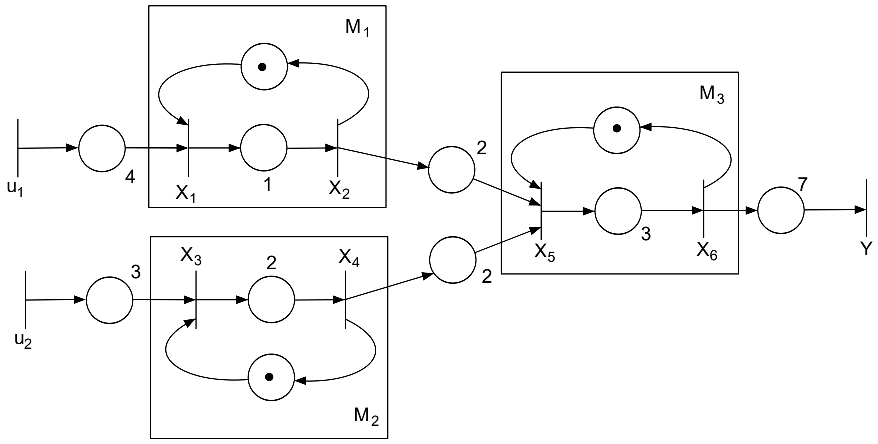


Figure 3.9: Manufacturing System

This system can be described by state space equations in max-plus algebra given by:

$$\begin{cases} x(k) = Ax(k-1) \oplus Bu(k) \\ y(k) = Cx(k) \end{cases}$$

with

$$A = \begin{bmatrix} . & e & . & . & . & . \\ . & 1 & . & . & . & . \\ . & . & . & e & . & . \\ . & . & . & 2 & . & . \\ . & 3 & . & 4 & . & 2 \\ . & 6 & . & 7 & . & 3 \end{bmatrix}, B = \begin{bmatrix} 4 & . \\ 5 & . \\ . & 3 \\ . & 5 \\ 7 & 7 \\ 10 & 10 \end{bmatrix} \text{ and } C = \begin{bmatrix} . & . & . & . & . & 7 \end{bmatrix}.$$

In order to solve this example, initially it will be considered the system working in open-loop (without feedback) and all inputs $u(k)$, $\forall k \succeq 1$, equal to e, *i.e.*, all the raw material will be available at system initial date. This behavior is useful to keep the machines working as long as possible, however, this behavior is inefficient since it will produce internal inventory. This assertion can be confirmed with the open-loop state vectors $x^{ol}(k)$:

$$x(0) = \begin{bmatrix} 4 & 5 & 3 & 5 & 7 & 10 \end{bmatrix}$$

$$x^{ol}(1) = \begin{bmatrix} 5 & 6 & 5 & 7 & 10 & 13 \end{bmatrix}$$

$$x^{ol}(2) = \begin{bmatrix} 6 & 7 & 7 & 9 & 13 & 16 \end{bmatrix}$$

$$x^{ol}(3) = \begin{bmatrix} 7 & 8 & 9 & 11 & 16 & 19 \end{bmatrix}$$

$$x^{ol}(4) = \begin{bmatrix} 8 & 9 & 11 & 13 & 19 & 22 \end{bmatrix}$$

$$x^{ol}(5) = \begin{bmatrix} 9 & 10 & 13 & 15 & 22 & 25 \end{bmatrix}$$

$$x^{ol}(6) = \begin{bmatrix} 10 & 11 & 15 & 17 & 25 & 28 \end{bmatrix}$$

$$x^{ol}(7) = \begin{bmatrix} 11 & 12 & 17 & 19 & 28 & 31 \end{bmatrix}$$

$$x^{ol}(8) = \begin{bmatrix} 12 & 13 & 19 & 21 & 31 & 34 \end{bmatrix}$$

Analyzing the vectors x^{ol}, it is possible to see that the transitions x_1 and x_2 fire every time unit from four time units and five time units. The transitions x_3 and x_4 fire every two time units and the transitions x_5 and x_6 fire every three time units. Therefore, this system is unstable because it can accumulate tokens in places between transitions x_2 and x_4 with transition x_5. This internal inventory is undesirable for some applications.

However, to deal with this issue, the feedback control can make the system stable. In open-loop, the system has a production rate equal to one product each three time units ($\lambda = 3$). Using the Equation 3.112, the maximal feedback in order to respect $\lambda = 3$ can be computed.

This maximal feedback matrix is given by:

$$F_m = \begin{bmatrix} -1 & -2 & 0 & -2 & -4 & -7 \\ -1 & -2 & 0 & -2 & -4 & -7 \end{bmatrix}.$$

However, the maximal feedback matrix is non causal since it has non positive entries. Using the Equation 3.115, the maximal causal feedback matrix for the system can be computed with $m = 4$ and it is given by:

$$F_c = \begin{bmatrix} 8 & 7 & 9 & 7 & 5 & 2 \\ 8 & 7 & 9 & 7 & 5 & 2 \end{bmatrix}.$$

Computing the system state with the feedback matrix ($x^{cl} = (A \oplus BF)x(k-m)$), the following system states are obtained.

$$x^{cl}(1) = \begin{bmatrix} 7 & 8 & 6 & 8 & 10 & 13 \end{bmatrix}$$

$$x^{cl}(2) = \begin{bmatrix} 10 & 11 & 9 & 11 & 13 & 16 \end{bmatrix}$$

$$x^{cl}(3) = \begin{bmatrix} 13 & 14 & 12 & 14 & 16 & 19 \end{bmatrix}$$

$$x^{cl}(4) = \begin{bmatrix} 16 & 17 & 15 & 17 & 19 & 22 \end{bmatrix}$$

$$x^{cl}(5) = \begin{bmatrix} 19 & 20 & 18 & 20 & 22 & 25 \end{bmatrix}$$

$$x^{cl}(6) = \begin{bmatrix} 22 & 23 & 21 & 23 & 25 & 28 \end{bmatrix}$$

$$x^{cl}(7) = \begin{bmatrix} 25 & 26 & 24 & 26 & 28 & 31 \end{bmatrix}$$

$$x^{cl}(8) = \begin{bmatrix} 28 & 29 & 27 & 29 & 31 & 34 \end{bmatrix}$$

Analyzing the vectors x^{cl}, it is possible to conclude that the production rate is respected and the output dates are equal to the output dates in open-loop system simulation, therefore the feedback does not cause delay in the system. It is also possible to conclude that the input dates in closed-loop simulation are greater than those in the open-loop simulation. This fact avoid the internal inventory in the system since the raw material will enter the system only when necessary.

The feedback control makes more robust the system in relation to internal parameter varying since it is able to limit the number of tokens in the system and, consequently, ensure the system stability for any sequence of inputs.

The system with the computed feedback is presented in Figure 3.10, in which it is possible to see that the system graph with feedback is strongly connected, so the system is stable for any input.

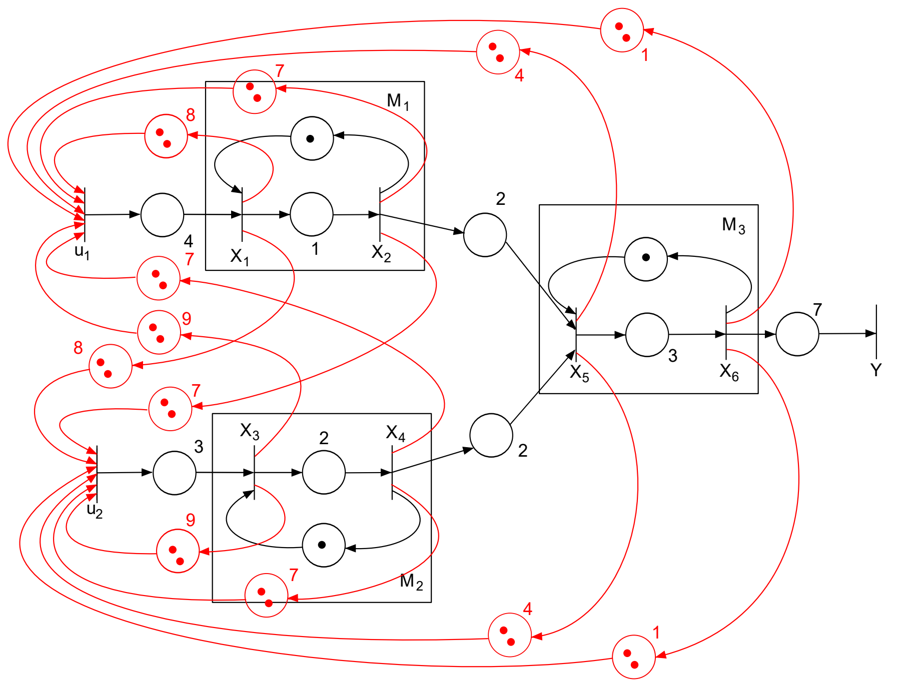


Figure 3.10: Manufacturing System

Based on ideas from Maia et al. (2005), the two control methodologies can be combined (the open-loop control and the feedback control). The combination is useful to ensure the desired behavior of the system and know the input dates in order to comply with deadline dates. The open-loop control will be useful to find the maximal input dates to enter the raw material in the system. The feedback control will be useful to ensure a more robust system in relation to system parameter varying.

This example was solved using the methodology of open-loop just-in-time control in a finite horizon. The maximal input dates found are given by:

$$u_1^{max}(k) = \begin{bmatrix} 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 \end{bmatrix}$$

$$u_2^{max}(k) = \begin{bmatrix} 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 \end{bmatrix}$$

Therefore, the feedback matrix can be used together with the maximal input dates to feed the system with the raw material.

3.5.4 Conclusion

This section presented the constrained Feedback control problem in the Just-in-Time context. The necessary conditions for the feedback matrix existence are developed as well as the maximum feedback matrix. In the end, three numerical examples illustrated the methodology and the applicability of results.

3.6 Synthesis of Controllers

In this section the characteristics of controllers developed in this thesis are presented. The controllers are used only in Discrete Event Systems, therefore, the open-loop control in finite horizon is useful to systems endowed with not huge transient behavior since after the transient part the system can have a periodic behavior, so the input dates will be periodic.

The open-loop control is useful to systems with huge transient behavior since the solution in finite horizon can be impracticable, *i.e.*, the computational time and computational memory cannot be feasible.

In addition, the open-loop control can ensure the optimal control but cannot ensure the stability addressed in this thesis. Therefore, the open-loop control is better to systems with negligible parameters variation, such as completely automated and traffic light systems.

However, the feedback control can ensure stability for any input, then it is useful to systems endowed with uncertain or variable parameters such as non-completed automated manufacturing systems with human operations.

Finally, the results found by using any of these methods will respect the constraints and, consequently, the reference demand (deadline dates).

Chapter 4

Final Discussion

4.1 Conclusion

Based on the formulation and results presented in this thesis, it is possible to conclude, first of all, that the contributions of this work will be useful to model, analyze, control, evaluate the performance and optimize Max-Plus Linear Dynamical Systems.

It is important to reinforce the fact that timed event graphs are graphical tools able to model discrete event systems where there is no concurrence by resources. The max-plus algebra is useful to describe the behavior of a system modeled as a timed event graph in a linear way, what is not the case in the conventional algebra. Concepts of classical system theory, such as state space equations, eigenvalues and eigenvectors, can be inherited from classical system theory.

The first contribution of this work is a general problem formulation based on the optimization theory. This formulation is useful to represent optimal control problems since it is possible to determine a main objective and constraints of interest. An important constraint in the general formulation is the semimodule equation because it can include non-convex constraints in the system. Some important constraints can be written as a semimodule equation such as, for example, temporal limitation of a manufacturing process duration or the maximal waiting time of passengers in a transportation system.

It was shown that the general formulation is simpler than some previous formulations published in papers by different authors. It is also more efficient since: the formulation deals with the direct realization of the problem; it deals only with the max-plus algebra allowing the use of non-convex constraints; it is possible to solve the problems in different ways; it obtains two different control policies.

From the general problem formulation it was possible to develop the open-loop and feedback control policies. The first policy developed was the open-loop control in the finite horizon. This methodology is useful when the deadline dates are finite. However, the control problem in a finite horizon can be understood as an infinite horizon by the expansion of the control horizon for a big enough size (greater than the transient interval). Two methodologies to solve the control problem was developed, one based on semimodule, which has double exponential computational complexity, and one based on the Modified Alternating Algorithm, which has pseudo-polynomial computational complexity. Using the second method, the maximal input dates that respect a viable deadline dates are computed with a simple algorithm in short time.

The second problem stated was the open-loop control policy in infinite horizon. The first methodology to solve the problem is based on (A,B)-invariant sets and the second methodology is based on semimodules and residuation theory. Both methodologies have high computational complexity. In the first case, finding the maximum (A,B)-invariant set inside a space needs a huge computational effort. In the second case, finding the space that has all solutions to the semimodule equations has a double exponential computational complexity. Although, the methodologies are able to solve important complex problems of practical interest, as previously mentioned, it is possible to consider non-convex constraints in conventional algebra.

In order to propose a solution with lower computational complexity, algebraic tools in max-plus algebra were applied. Using the results presented in Gonçalves (2015), it was possible to solve the problem considering a particular reference model, without any transient behavior that evolves by a rate λ.

Finally, the feedback control policy, in the Just-in-Time context, was addressed in this work. The necessary conditions for the feedback matrix existence and a method to compute the maximum feedback matrix in order to comply with deadline dates were presented.

The theoretical results show that, respecting the proposed conditions, it is always possible to apply the Just-in-Time control policy. The necessary and sufficient conditions to solve the control problems are developed as well as the optimal solutions to the classes of proposed problems. In the end of each section, along with theoretical development, numerical examples were presented with the intention to show the applicability of the new results in important classes of systems of practical interest.

The proposed optimization control problem is useful to companies, *e.g.*, industrial companies or small warehouses, since it is able to control this kind of systems by discrete event systems theory without complex mathematical equations. It is also able to determine the maximal input dates in order to comply with the demand trajectory.

In conclusion, it is possible to use the contributions of this thesis to analyze the dynamic of the system. For example, in a queuing system, if the smallest viable deadline dates are not feasible for a system of interest, the number of servers can be improved.

4.2 Future Works

As perspective of future works from this thesis, the following items are listed:

- Applying the obtained results in real systems, even in prototypes of real systems, to develop and analyze the real behavior of the system, as well as developing easy ways to apply the contributions in practice.

- Investigating new control policies that can be obtained from the general formulation like the cyclic control, given the cyclicity of some max-plus linear systems, in special industrial systems. Therefore the control policy can be computed in a finite horizon considering one cycle of production, as well as applying new constraints in the control problem.

- Developing the theory of variant max-plus linear systems, initially presented in Gomes da Silva and Maia (2015a). This theory considers the entries of matrices in state space equations as variables, stochastically or in an interval of parameters.

Bibliography

Allamigeon, X., Gaubert, S., and Goubault., E. (2008). Inferring min and max invariants using max-plus polyhedra. In *Proceedings of the 15th International Static Analysis Symposium (SAS'08), Springer LNCS number 5079*, pages 189–204.

Amari, S. and Isabel Demongodin, Jean Jacques Loiseau, C. M. (2012). Max-plus control design for temporal constraints meeting in timed event graphs. In *IEEE Transactions on Automatic Control*, volume 57, pages 462 – 467.

Baccelli, F., Cohen, G., Olsder, G., and Quadrat, J. (1992). *Synchronization and Linearity: An Algebra for Discrete Event Systems.* John Wiley and Sons, New York.

Banks, J., Carson, J., Nelson, B., and Nicols, D. (2005). *Discrete Event System Simulation.* Pearson, 4th edition.

Butkovic, P. and Hegedus, G. (1984). An elimination method for finding all solutions of the system of linear equations over an extremal algebra. *Ekon. Mat. Obzor*, 20:203–215.

Cassandras, C. G. and Lafortune, S. (2008). *Introduction to Discrete Event Systems.* Kluwer Academic Publishers - Springer, 2 edition.

Cohen, G., Gaubert, S., and Quadrat, J. (2004). Duality and separation theorems in idempotent semimodules. *Linear Algebra and Appl.*, 379:395–422.

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

De Schutter, B. and van den Boom, T. (2001). Model predictive control for max-plus-linear discrete event systems. *Automatica*, 37:1049–1056.

de Vries, R., De Schutter, B., and De Moor, B. (1998). On max-algebraic models for transportation networks. *Workshop On Discrete Event Systems (WODES'98)*, pages 457–462.

Garcia, C. E., Prett, D. M., and Morari, M. (1989). Model predictive control: Theory and pratice - a survey. *Automatica*, 25(3):335–348.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company Publishers.

Gaubert, S. (1992). *Théorie des systèmes linéaires dans les dioides.* Thèse de doctorat, Ecole des Mines de Paris, Paris.

Gaubert, S. and Sergeev, S. (2013). The level method for the two-sided eigenproblem. *Discrete Event Dinamical Systems Journal*, 23:105–134.

Gomes da Silva, G. and Maia, C. A. (2014). On just-in-time control of timed event graphs with input constraints: a semimodule approach. *Discrete Event Dynamic Systems*, 26:351–366. DOI: 10.1007/s10626-014-0200-z.

Gomes da Silva, G. and Maia, C. A. (2015a). Controle just-in-time aplicado à um sistema de transporte urbano max-plus linear. In *Simpósio Brasileiro de Automação Inteligente*, pages 170–175, Natal - Rio Grande do Norte, Brazil.

Gomes da Silva, G. and Maia, C. A. (2015b). A multiobjective formulation for just-in-time control of constrained max-plus linear systems in infinite horizon. In *IEEE Conference on Decision and Control*, pages 350–355, Osaka - Japan. DOI: 10.1109/CDC.2015.7402225.

Gonçalves, V. M. (2015). *Tropical Algorithms for Linear Algebra and Linear Event-invariant Dynamical Systems.* PhD thesis, Universidade Federal de Minas Gerais, Brazil.

Gonçalves, V. M., Maia, C. A., and Hardouin, L. (2014). Avanços na teoria de controle para sistemas lineares na algebra tropical. In *Congresso Brasileiro de Automatica*, pages 1704–1711.

Gondran, M. and Minoux, M. (2010). *Graphs, Dioids and Semirings: New Models and Algorithms*. Springer Publishers.

Houssin, L., Lahaye, S., and Boimond, J. (2007). Just in time control of constrained (max ,+)-linear systems. *Discrete Event Dynamic Systems. Theory and Applications*, 17(2):159–178.

Houssin, L., Lahaye, S., and Boimond, J. L. (2013). Control of (max,+)-linear systems minimizing delays. In *Discrete Event Dynamic Systems*, volume 23, pages 261–276. Springer Publishers.

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

Maia, C. A. (2003). *Identificação e Controle de Sistemas a Eventos Discretos na Álgebra (max,+)*. Tese de doutorado, UNICAMP, Campinas, SP, Brasil.

Maia, C. A., Andrade, C. R., and Hardouin, L. (2011a). On the control of max-plus linear system subject to state restriction. *Automatica*, 47(5):988–992.

Maia, C. A., Hardouin, L., and Cury, J. E. R. (2013). Some results on the feedback control of max-plus linear systems under constraints. In *52th IEEE Conference Decision and Control*, number DOI:10.1109/CDC.2013.6760997, pages 6992–6997, Fiorence, Italy.

Maia, C. A., Hardouin, L., Mendes, R. S., and Loiseau, J. (2011b). A super-eigenvector approach to control constrained max-plus linear systems. In *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC'11)*, number DOI:10.1109/CDC.2011.6160411, pages 1136–1141, Orlando, Florida.

Maia, C. A., Luders, R., Mendes, R. S., and Loiseau, J. (2005). Estratégias de controle por modelo de referência de sistemas a eventos discretos max-plus lineares. *Controle e Automação*, 16(3):263–278. DOI: 10.1590/s0103-17592005000003000001.

Menguy, E., Boimond, J., Hardouin, L., and Ferrier, J. (2000). Just-in-time Control of Timed Event Graphs Update of Reference Input, Presence of Uncontrollable Input. *IEEE Trans. on Automatic Control*, 45(11):2155–2158.