EVOLUTION OF KANBAN SYSTEMS THANKS TO A (MAX,+)-ALGEBRA ANALYSIS

Bertrand Cottenceau * **Laurent Hardouin** *
Iteb Ouerghi *

* *LISA, university of Angers, France,*
bertrand.cottenceau@univ-angers.fr,
laurent.hardouin@univ-angers.fr,
iteb.ouerghi@univ-angers.fr

Abstract: This paper deals with a possible evolution of a Kanban system due to a (max,+)-algebra analysis. We show that for a given Kanban system, it is always possible to change the original Kanban policy by a (max,+)-linear policy which keeps the same quality of service but reduces the work in process. This new control policy contains a (max,+)-linear dynamic behavior for the recycling of kanban cards.

Keywords: (max,+)-linear systems, Kanban systems, feedback synthesis.

1. INTRODUCTION

This paper aims at showing that some recent tools from the (max,+) linear systems theory (Cohen *et al.*, 1989) (Baccelli *et al.*, 1992) allow to consider some classical production policies differently. This formal aspect gives analytical models of production management policies, which allows the performance analysis. It means considering the classical control policies under another aspect, and generally improving them. In order to provide a concrete aspect to this demonstration, the Kanban policy is analyzed. First the algebraic model of this policy is given, and it is shown that it can be compared to a feedback control ensuring a limitation of the work in process (WIP). The quality of service and the work in process are both expressed analytically. Thanks to these expressions, a controller's algebraic synthesis preserving the quality of service is proposed (the problem of controller synthesis is studied in (Cottenceau *et al.*, 2001) (Maia *et al.*, 2003) and (Lhommeau *et al.*, 2004)). The controller obtained reduces the WIP, which means that the corresponding policy has the same efficiency from the customer's point of view while reducing the internal stocks of the system. The simulations and the numerical examples allow to illustrate the synthesis and to evaluate the improvement.

2. LINEAR MODELS OF SOME PRODUCTION CELLS IN (MAX,+)-ALGEBRA

2.1 Timed Event Graphs

Among production systems, we are interested in the ones that we can model by linear recurrences in the (max,+) algebra. From a practical point of view, they correspond to *Discrete Event Systems* where the main phenomena are *time delays* (such as transportation times or processing times) and *synchronizations* (for instance, a processing can begin only when a raw part and a machine are available simultaneously).

These systems can also be described by some graphs called *Timed Event Graphs.* Timed Event Graphs (TEGs) are a subclass of timed Petri Nets where each place has exactly one upstream and one downstream transition. Therefore, the concurrency phenomena cannot be described. For a TEG (see Fig.1 for instance), a transition (bar) is "fired" once each upstream place (circle) contains one available token (small black circle). Moreover, a time delay can be associated to a place : a token must consume this delay before becoming available to fire the upstream transition.

Fig.1 represents the TEG model of a machine denoted M_1 which can process simultaneously up to 3 parts (machine capacity). The process time for each part is 2 time units (time delay associated to place $x_2 \rightarrow x_3$). Raw parts are taken from an unlimited upstream buffer (place denoted B1) and the finished parts are released in an unlimited downstream buffer (place denoted B2). The firing of transition x_1 models the input

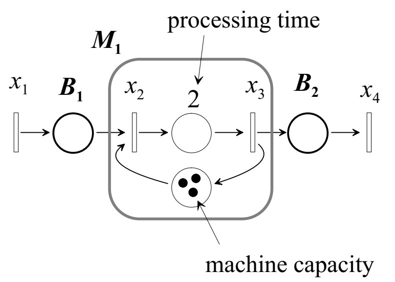

Fig. 1. Timed Event Graph model of a machine processing 3 parts each 2 time units

of a raw material in buffer B_1, and the firing of transition x_4 represents the departure of a part from buffer B_2.

2.2 Linear systems in (max,+)-algebra

The behavior of machine M_1 (with buffers B_1 and B_2) can be modelled by some recurrences. By denoting $x_i(k)$ the date of the $(k+1)^{\text{th}}$ firing of transition x_i - the function $k \mapsto x_i(k)$ is called the *dater function* associated to transition x_i - the behavior of machine M_1 is described by the following equations:
beginning of the $(k+1)^{\text{th}}$ processing

$$x_2(k) = \max(x_1(k), x_3(k-3))$$

end of the $(k+1)^{\text{th}}$ processing

$$x_3(k) = 2 + x_2(k).$$

The relation $x_2(k) = \max(x_1(k), x_3(k-3))$ models a synchronization phenomenon. The $(k)^{\text{th}}$ processing can begin only if raw material has entered in buffer B_1 and if the $(k-3)^{\text{th}}$ part [1] has been completely processed in machine M_1. The relation $x_3(k) = 2 + x_2(k)$ models the processing time of machine M_1.

If we denote by $\oplus$ the max operator and by $\otimes$ the classical sum $+$, the previous relations can be rewritten as :

$$\begin{aligned} x_2(k) &= x_1(k) \oplus x_3(k-3) \\ x_3(k) &= 2 \otimes x_2(k). \end{aligned} \tag{1}$$

Assuming that all firing dates and all time delays are some integers, it is shown in (Cohen *et al.*, 1989) and (Baccelli *et al.*, 1992) that the behavior of a TEG can always be described by linear recurrences over the (max,+)-algebra denoted $(\mathbb{Z}, \oplus, \otimes)$. This algebraic structure is an idempotent semiring [2] (see Def. 1 in Appendix). We can always represent the input-output behavior of a TEG by a linear state model over the (max,+) algebra such as :

$$\begin{cases} x(k) = A \otimes x(k-1) \oplus B \otimes u(k) \\ y(k) = C \otimes x(k) \end{cases} \tag{2}$$

where $x(k)$ is a vector of dater functions associated to internal events (such as transitions x_2 and x_3 in Fig.1), $u(k)$ is a vector of dater functions associated to input events (such as x_1 in Fig.1) and $y(k)$ is a vector of dater functions associated to output events (such as x_4 in Fig.1). In the sequel, the operator $\otimes$ will be sometimes omitted, as in classical algebra.

2.3 Transfer relation, γ-transform

As presented exhaustively in (Baccelli *et al.*, 1992) and (Cohen *et al.*, 1989), by introducing a backward shift operator in the event domain denoted γ, we can associate γ-series to dater functions of a TEG. The γ-transform of a dater function $k \mapsto x_i(k)$ is the formal series $x_i(\gamma) = \bigoplus_{k\in\mathbb{Z}} x_i(k)\gamma^k$. The set of γ-series is also an idempotent semiring denoted $\overline{\mathbb{Z}}_{max}[\![\gamma]\!]$. Thanks to the γ-transform, we obtain a new linear model for the behavior of a TEG. For instance, the state model (2) can be transformed into the following one over the semiring $\overline{\mathbb{Z}}_{max}[\![\gamma]\!]$:

$$\begin{cases} x(\gamma) = \gamma A\, x(\gamma) \oplus B\, x(\gamma) \\ y(\gamma) = C\, x(\gamma) \end{cases} \tag{3}$$

and finally, we can exhibit a direct input-output relation. First, the state equation can be solved thanks to Theorem 1 (see appendix), $x(\gamma) = (\gamma A)^* B\, u(\gamma)$. Then, $y(\gamma) = C(\gamma A)^* B u(\gamma)$. The transfer relation of the (max,+)-linear system given in (2) is

$$y(\gamma) = H\, u(\gamma), \tag{4}$$

where $H = C(\gamma A)^* B$ is the transfer matrix.

[1] If the $(k-3)^{\text{th}}$ part has been processed, it also means that at least one of the 3 resources of machine M_1 is available.

[2] An idempotent semiring is often called dioid in literature

Example 1. For the machine M_1 depicted on Fig. 1, the dater functions (1) can be transformed into

$$\begin{aligned} x_2(\gamma) &= x_1(\gamma) \oplus \gamma^3 x_3(\gamma) \\ x_3(\gamma) &= 2\, x_2(\gamma). \end{aligned}$$

Therefore, $x_3(\gamma) = 2(x_1(\gamma) \oplus \gamma^3 x_3(\gamma))$. By solving this implicit equation on $\overline{\mathbb{Z}}_{max}[\![\gamma]\!]$ (see Th.1), the transfer relation between $x_1(\gamma)$ and $x_3(\gamma)$ is given by:

$$\begin{aligned} x_3(\gamma) &= h_{M_1} \otimes x_1(\gamma) \\ &= 2(2\gamma^3)^* \otimes x_1(\gamma) \\ &= 2\gamma^0 \otimes (0\gamma^0 \oplus 2\gamma^3 \oplus 4\gamma^6 \oplus ...) \otimes x_1(\gamma) \end{aligned}$$

Remark 1. It is important to notice that all the dynamical characteristics of the machine M_1 are embedded in the transfer series $h_{M_1} = 2(2\gamma^3)^*$. For a given input trajectory $x_1(\gamma)$, the machine output is obtained by making the product of the series $x_1(\gamma)$ by the transfer series h_{M_1}.

Example 2. Let us consider $x_1(\gamma) = 1\gamma^0 \oplus 3\gamma^1 \oplus 3\gamma^2 \oplus 3\gamma^3 \oplus 3\gamma^4 \oplus +\infty\gamma^5$. This series models the following input trajectory: transition x_1 is fired once at date 1 (the first event is numbered 0), x_1 is fired 4 times at date 3 ($+\infty\gamma^5$ means that the 6-th firing never occurs). The corresponding output is

$$\begin{aligned} x_3(\gamma) &= h_{M_1} \otimes x_1(\gamma) \\ &= 3\gamma^0 \oplus 5\gamma^1 \oplus 5\gamma^2 \oplus 5\gamma^3 \oplus 7\gamma^4 \oplus +\infty\gamma^5, \end{aligned}$$

which means that transition x_3 is fired once at date 3, then it is fired 3 times at date 5 and finally 1 time at date 7.

This calculus can be computed under Scilab software thanks to the MinMaxGD package (SW2001, 2001). The appendix gives the script of this example.

Example 3. (Transfer of a production line). We can easily extend the previous example to a production line with several machines with some intermediate buffers. For instance, Fig. 2 describes a cell with two machines denoted M_1 and M_2. The intermediate buffer B_2 is an unlimited buffer and the time delay of 2 time units associated to the place represents a transportation time between machines M_1 and M_2. The main transfer relations are $h_{M_1} = 2(2\gamma^3)^*$, $h_{M_2} = 4(4\gamma^5)^*$.

The input-output transfer of this cell is the product of the transfer series, say $h = h_{M_1} \otimes h_{M_2} = 4(4\gamma^5)^* \otimes 2\gamma^0 \otimes 2(2\gamma^3)^*$. The computation of this product with the package MinMaxGD leads to:

$$h = (8\gamma^0 \oplus 10\gamma^3)(4\gamma^5)^*.$$

3. MODEL AND EVOLUTION OF KANBAN CELLS

3.1 The Kanban policy

In many manufacturing systems, production of parts proceeds in stages. Each stage may be seen as a production/inventory system with a single machine or a subnetwork of several machines. An important managerial concern is how to control the flow of parts through the stages. This can be done by implementing a pull control policy for which production is triggered by actual customer demands. Pull systems are motivated by the concept of Just-In-Time (JIT) whose aim is that products should be produced only when ordered and in the quantities needed.

The Kanban control system is the most well known pull control policy, for which a number of authorization cards, called Kanbans, is used to limit the Work-In-Process in each stage. Many works deal with modelling and performance analysis of these systems. The reader is invited to consult (Di Mascolo *et al.*, 1991), (Gaubert, 1992) and (Chaouiya and Dallery, 1997).

3.2 Block diagram of a Kanban cell

The input-output model obtained in the semiring $\overline{\mathbb{Z}}_{max}[\![\gamma]\!]$ allows to represent systems as *block diagrams* where a block represents a certain (max,+)-linear system with a given transfer function. In

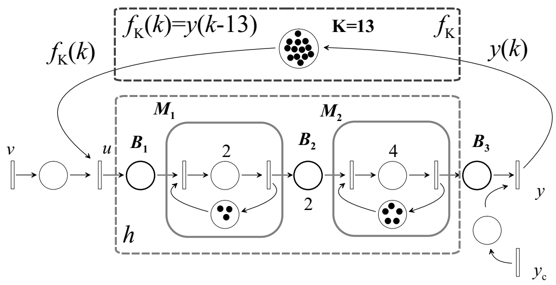

Fig. 2. TEG model of a production cell with 2 machines and 1 kanban loop

this framework, the Kanban policy can be seen as a system fed back to the production line, as modelled with Petri nets (see (Di Mascolo *et al.*, 1991)). The first (max, +)-linear model of Kanban systems is due to (Gaubert, 1992).

Let us consider the production cell depicted on Fig. 2. The production line is the forward system whose transfer h_c has been computed in Example 3. The Kanban loop (with 13 Kanban cards in Fig. 2) is a feedback system having the following transfer function [3]

[3] In $\overline{\mathbb{Z}}_{max}[\![\gamma]\!]$, the feedback of a Kanban system only induces some shifts in the event domain.

$$f_K = \gamma^K = \gamma^{13}.$$

If we express the Kanban loop as a (max,+)-linear system, one obtains

$$f_K(k) = y(k - K) = y(k - 13),$$

and the cell input (denoted u) has the following behavior

$$u(k) = v(k) \oplus f_K(k) = v(k) \oplus y(k - K).$$

Generically, we can consider a Kanban cell as a two-block system as depicted on Fig. 3 : a forward system h which contains the dynamics of the production line and a feedback system f_K the dynamics of which depends on the number of Kanban cards. In this generic model, the output

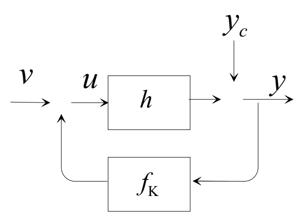

Fig. 3. Block diagram of a Kanban cell

y depends both on the customer demand y_c and on the raw parts availability v. From this point of view, a Kanban cell is a 2-input 1-output system whose transfer relation is formally given by

$$y = (h\, f_K)^* \, y_c \oplus (h\, f_K)^* h\, v \tag{5}$$

On the one hand, the transfer $y_c \to y$, *i.e.* $(h\, f_K)^*$, provides the dynamic behavior between a given demand y_c and the finish parts output y. It characterizes the customer's quality of service. On the other hand, the WIP is defined by the difference (in the event domain) between the trajectory u and y.

3.3 Evolution of the Kanban feedback

Different works in the (*max*,+) literature address the problem of control. In particular, it is shown that we can modify the behavior of a (max,+)-linear system thanks to another (max,+)-linear system called *controller* (Cottenceau *et al.*, 2001), (Maia *et al.*, 2003) (Lhommeau *et al.*, 2004). This approach is a transposition in the (max,+) framework of the classical problem of Model Reference Control.

For a Kanban system, the feedback loop (the loop which contains the Kanban cards) can be seen as a particular controller (as depicted on the block diagram of Fig. 3). We will show hereafter that the feedback loop of the Kanban system (the original controller) can be replaced by another controller which preserves the quality of service for the customer while reducing the Work-In-Process (WIP). In other words, we want to obtain a new feedback controller f such that 1) the transfer $y_c \to y$ remains unchanged (same quality of service) 2) the WIP is reduced (*i.e.* u is delayed).

Since the transfer relation of a Kanban cell is given by (5), one merely has to find a (max,+)-linear feedback f such that

$$(h\, f)^* = (h\, f_K)^* \quad \text{(same quality of service)} \tag{6}$$

$$f \succeq f_K \quad \text{(reduced WIP)}, \tag{7}$$

In one hand, if the controller f satisfies (6) then the transfer relation (5) is unchanged. On the other hand, the order relation $f \succeq f_K$ ($\succeq$ is defined in Def.2 in Appendix) means that we want to delay u in order to reduce the WIP.

The residuation theory (Baccelli *et al.*, 1992, Chap.4) is used to tackle this problem (see Th.2 in Appendix). It provides a pseudo-inverse for the $\otimes$ operator.

Proposition 1. The feedback system

$$\hat{f} = h \diamond\!\!\!\backslash (h\, f_K)^* \tag{8}$$

is the greatest feedback controller which preserves the quality of service, *i.e.* $(h\hat{f})^* = (h\, f_K)^*$.

Proof: Firstly, the equation $(h\, f)^* \preceq (h\, f_K)^*$ is considered. Thanks to Th. 3 and Th.2 (see appendix),

$$(h\, f)^* \preceq (h\, f_K)^* \iff h\, f \preceq (h\, f_K)^*$$
$$\iff f \preceq h \diamond\!\!\!\backslash (h\, f_K)^* = \hat{f}.$$

Secondly, $\hat{f}$ ensures equality since f_K is a solution of (6). □

Remark 2. The previous proposition means that we can replace the feedback loop f_K of a Kanban system by $\hat{f}$ given in (8). This controller keeps the same quality of service and reduces the WIP as much as possible (it is the greatest controller which satisfies (6)).

Example 4. Let us consider the system depicted on Fig. 2. The production cell transfer is $h = (8\gamma^0 \oplus 10\gamma^3)(4\gamma^5)^*$ and the feedback transfer is $f_K = \gamma^{13}$. Therefore, the transfer series of the feedback controller $\hat{f}$ is (see appendix for the script)

$$\hat{f} = h \diamond\!\!\!\backslash (h\, f_K)^* = (\gamma^{13} \oplus 2\gamma^{16})(4\gamma^5)^*.$$

We can also express this feedback as a (max,+)-linear system whose input is y

$$\hat{f}(k) = y(k - 13) \oplus 2y(k - 16) \oplus 4\hat{f}(k - 5),$$

and the cell input u has the following behavior

$$u(k) = v(k) \oplus \hat{f}(k).$$

Fig. 4 illustrates this new control policy. Clearly, the feedback loop $\hat{f}$ cannot be implemented only with Kanban cards. For this new control policy, a (max,+) dynamic must be added in the recycling of Kanban cards. This control policy requires to manage some "virtual Kanban cards" with a software assistance.

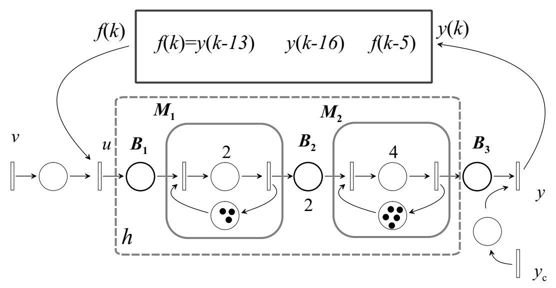

Fig. 4. Production cell with an evolution of a Kanban cell

3.4 Extension to Kanban multi-stage

We can extend this approach to production systems with several Kanban stages. The generic block diagram for several stages with different Kanban loops is depicted in Fig. 5.

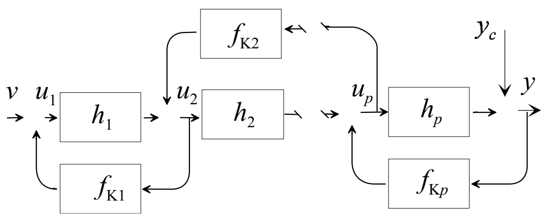

Fig. 5. Block diagram of a production line with several Kanban stages.

Proposition 2. (see (Cottenceau, 1999)). The transfer relation of a multi-stage (in series) Kanban system (with p stages) is given by the formal relation

$$y = (\alpha_p\, f_{\mathrm{K}_p})^* \, y_c \oplus (\alpha_p\, f_{\mathrm{K}_p})^* \beta_p \, v, \qquad (9)$$

with $\alpha_1 = \beta_1 = h_1$, $\alpha_{i+1} = h_{i+1}(\alpha_i f_{\mathrm{K}_i})^*$, and $\beta_{i+1} = \alpha_{i+1}\beta_i$.

Example 5. (Kanban with two stages (p=2)). The recurrence (9) leads to $\alpha_1 = \beta_1 = h_1$, $\alpha_2 = h_2(h_1\, f_{\mathrm{K}_1})^*$ and $\beta_2 = h_2(h_1\, f_{\mathrm{K}_1})^* h_1$. Then, the transfer relation is

$$\begin{aligned} y \;=\; & (h_2(h_1\, f_{\mathrm{K}_1})^* f_{\mathrm{K}_2})^* \, y_c \oplus \\ & (h_2(h_1\, f_{\mathrm{K}_1})^* f_{\mathrm{K}_2})^* h_2(h_1\, f_{\mathrm{K}_1})^* h_1 \, v. \end{aligned}$$

Proposition 3. The transfer relation of a multi-stage (with p stages) Kanban system remains unchanged by replacing each feedback f_{K_i} by the following controller

$$\hat{f}_i = \alpha_i \lcirclearrowright (\alpha_i\, f_{\mathrm{K}_i})^*. \qquad (10)$$

Proof: Proposition 1 shows that $\hat{f}_i$ is the greatest solution to $(\alpha_i f)^* = (\alpha_i\, f_{\mathrm{K}_i})^*$. Therefore, by replacing each feedback controller f_{K_i} by the controller $\hat{f}_i$, the transfer relation (9) remains unchanged. □

Example 6. The system depicted in Fig. 6 is a production line with 2 Kanban stages. Each stage contains two machines and 3 buffers. According to the generic block diagram given in Fig. 5, the transfer relations [4] are

$$\begin{aligned} h_1 &= (8\gamma^0 \oplus 10\gamma^3)(4\gamma^5)^* \\ f_{\mathrm{K}_1} &= \gamma^{13} \\ h_2 &= 14 \oplus (19\gamma^3 \oplus 22\gamma^4 \oplus 24\gamma^6)(8\gamma^4)^* \\ f_{\mathrm{K}_2} &= \gamma^8. \end{aligned}$$

We obtain,

$$\begin{aligned} \hat{f}_1 &= h_1 \lcirclearrowright (h_1\, f_{\mathrm{K}_1})^* \\ &= (\gamma^{13} \oplus 2\gamma^{16})(4\gamma^5)^* \\ \hat{f}_2 &= \alpha_2 \lcirclearrowright (\alpha_2\, f_{\mathrm{K}_2})^* \\ &= \gamma^8 \oplus (5\gamma^{11} \oplus 8\gamma^{12} \oplus 10\gamma^{14})(8\gamma^4)^*. \end{aligned}$$

These controllers can be expressed as two (max,+)-linear systems : $\hat{f}_1$ is a controller (computed in Example 4) whose input is u_2,

$$\hat{f}_1(k) = u_2(k-13) \oplus 2u_2(k-16) \oplus 4\hat{f}_1(k-5).$$

For controller $\hat{f}_2$, the corresponding (max,+)-linear system needs an internal state variable denoted $\hat{x}_{f_2}$

$$\begin{cases} \hat{x}_{f_2}(k) = 5y(k-11) \oplus 8y(k-12) \oplus \\ \qquad\qquad 10y(k-14) \oplus 8\hat{x}_{f_2}(k-4) \\ \hat{f}_2(k) = y(k-8) \oplus \hat{x}_{f_2}(k) \end{cases}$$

4. SIMULATION

The system depicted on Fig. 6 is simulated both for a classical Kanban policy ($f_{\mathrm{K}_1} = \gamma^{13}$ and $f_{\mathrm{K}_2} = \gamma^8$) and for the (max,+)-law computed in example 6 (feedback f_{K_1} (resp.f_{K_2}) is replaced by $\hat{f}_1$ (resp.$\hat{f}_2$)). The maximal production rate of this system is 0.5, it is the production rate of machine M_4 (4 parts/8 time units). These systems are compared for different utilization rates (demand rate/production rate) [5] .

We recall that for all demands y_c, the two policies always give the same output y, whereas the WIP is reduced for the $(\max,+)$ policy since the raw parts input u is delayed by a feedback $\hat{f}$ "slower than" the existing Kanban feedback f_{K}. The following table gives the WIP in cell 1 and in cell 2 for each policy and for different utilization rates.

---

[4] The Scilab scripts using the MinMaxGD package are given in appendix.

[5] For all the simulations, raw parts v are assumed to be always available

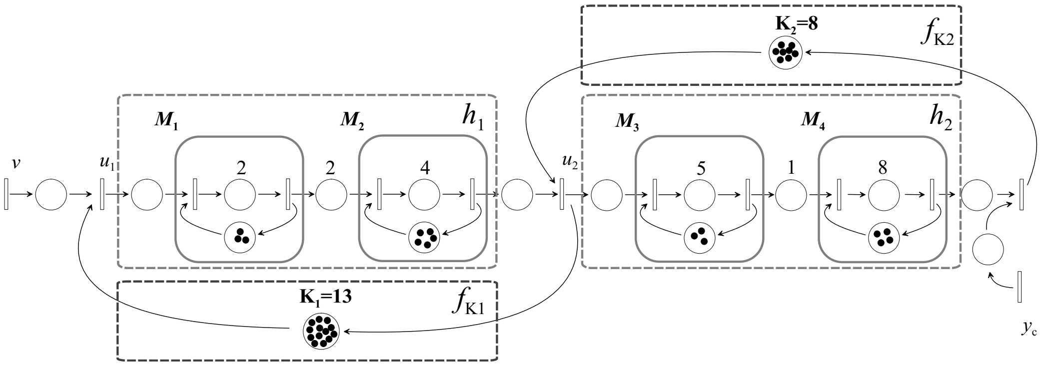

Fig. 6. TEG model of a production line with 2 Kanban stages in series.

Utilization	WIP Cell 1		WIP Cell 2	
	Kanban	(max,+)	Kanban	(max,+)
0.1	13	13	8	7.95
0.2	13	13	8	7.7
0.3	13	13	8	7.6
0.4	13	13	8	7.5
0.5	13	13	8	7.4
0.6	13	13	8	7.2
0.7	13	13	8	7.1
0.8	13	13	8	7
0.9	13	13	8	7
1	13	13	8	7

This simulation shows that the WIP is globally reduced. The main reduction due to the $(\max, +)$ control policy is in cell 2. For an utilization rate of 0.9, the $(\max, +)$ policy gives a global reduction of 5% of WIP (12% in cell 2).

5. CONCLUSION

This paper shows that the Kanban control policy can be described in the (max,+) algebra and that the algebraic model obtained allows the performances analysis of the controlled systems. Moreover, the $(\max, +)$ model can also be used to synthesize a control law which preserves the same quality of service as in the classical Kanban system, but which reduces the work in process. The $(\max, +)$ analysis allows to improve the Kanban policy.

The method proposed here to improve the Kanban policy, with the help of (max,+) tools, could be applied to other existing control policies. Indeed, such policies are often "tuned" from optimization procedures which do not take the intrinsic system' dynamic into account. The reader is invited to apply the methodology presented here to some well known methods such as base stock control (see (Dallery and Liberopoulos, 2000)) or generalized Kanban (see (Buzacott, 1989)) in order to persuade himself of this fact.

REFERENCES

Baccelli, F., G. Cohen, G.J. Olsder and J.P. Quadrat (1992). *Synchronization and Linearity : An Algebra for Discrete Event Systems.* Wiley and Sons.

Buzacott, J.A. (1989). Queuing models of kanban and MRP controlled manufacturing systems. *Engineering Cost and Production Economics* **17**, 3–20.

Chaouiya, C. and Y. Dallery (1997). Petri net models of pull control systems for assembly manufacturing systems. *Workshop on Manufacturing and Petri Nets, ICATPN,Toulouse.*

Cohen, G., P. Moller, J.P. Quadrat and M. Viot (1989). Algebraic Tools for the Performance Evaluation of Discrete Event Systems. *IEEE Proceedings: Special issue on Discrete Event Systems* **77**(1), 39–58.

Cottenceau, B. (1999). Contribution à la commande de systèmes à événements discrets : synthèse de correcteurs pour les graphes d'événements temporisés dans les dioïdes. Thèse. LISA - Université d'Angers.

Cottenceau, B., L. Hardouin, J. L. Boimond and J. L. Ferrier (2001). Model reference control for timed event graphs in Dioids. *Automatica* **37**, 1451–1458.

Dallery, Y. and G. Liberopoulos (2000). Extended kanban control system : combining kanban and base stock. *IIE Transactions on Design and Manufacturing* **32**, 369–386.

Di Mascolo, M., Y. Frein, Y. Dallery and R. David (1991). A unified modeling of kanban systems using petri nets. *International journal of Flexible Manufacturing Systems* pp. 275–307.

Gaubert, S. (1992). Théorie des Systèmes Linéaires dans les Dioïdes. Thèse. École des Mines de Paris.

Lhommeau, M., L. Hardouin, B.Cottenceau and L.Jaulin (2004). Interval analysis in dioid : Application to robust controller design for timed event graphs. *Automatica* **40**, 1923–1930.

Maia, C.A., L. Hardouin, R.S. Mendes and B.Cottenceau (2003). Optimal closed-loop control for timed event graphs in dioid. *IEEE Transaction on Automatic Control* **48**, 2284–2287.

SW2001 (2001). Software Tools for Manipulating Periodic Series. http://www.istia-angers.fr/~hardouin/outils.html.

6. APPENDIX

6.1 Algebraic results

Definition 1. (Idempotent semiring). An idempotent semiring $\mathcal{S}$ is a set endowed with two internal operations denoted $\oplus$ (addition) and $\otimes$ (multiplication), both associative and both having neutral elements denoted ε and e respectively, such that $\oplus$ is also commutative and idempotent (*i.e.* $a \oplus a = a$). The $\otimes$ operation is distributive with respect to $\oplus$, and ε is absorbing for the product (*i.e.* $\varepsilon \otimes a = a \otimes \varepsilon = \varepsilon, \quad \forall a$). When $\otimes$ is commutative, the semiring is said to be commutative. The symbol $\otimes$ is often omitted.

Definition 2. (Order relation). Idempotent semirings can be endowed with a natural order : $a \succeq b$ iff $a = a \oplus b$.

Definition 3. (Kleene star). On a complete semiring, the Kleene star operator is defined by $a^* = \bigoplus_{i \in \mathbb{N}} a^i$.

Theorem 1. (Implicit equation). The implicit equation $x = ax \oplus b$ defined on a complete semiring admits $x = a^*b$ as least solution.

Theorem 2. (Residuation). On a complete semiring, $ax \preceq b$ admits $a \backslash\!\!\circ b$ as greatest solution, for all a, b. Identically, $xa \preceq b$ admits a greatest solution denoted $b /\!\!\circ a$.

Theorem 3. On a complete semiring, the equation $x^* = a^*$ admits $x = a^*$ as greatest solution.

6.2 MinMaxGD package in Scilab

The MinMaxGD package (free to download on (SW2001, 2001)) allows to compute the operations of the semiring $\overline{\mathbb{Z}}_{max}[\![\gamma]\!]$. The series must be handled on a periodic form $s = p \oplus q(\tau\gamma^\nu)^*$, where p and q are polynomials, and r is a monomial.

Example 7. (Transfer series of M_1 Fig. 1). The transfer series of machine M_1 in Fig. 1 $h_{M_1} = 2(2\gamma^3)^*$, is a periodic series of $\overline{\mathbb{Z}}_{max}[\![\gamma]\!]$ where $p = \varepsilon$ (the null series), $q = 2\gamma^0$, and $r = 2\gamma^3$. With the MinMaxGD package, series h_{M_1} is defined as

```
-->h_M1=series([eps],[0 2],[3 2])
   h_M1  = (g^0d^2)[g^3d^2]*
```

Example 8. (Output trajectory). Example 2 can be computed as follows:

```
   x1=series([0 1;1 3;5 4],[6 %inf],[0 0]);
   h_M1=series([eps],[0 2],[3 2]);
   x4=h_M1*x1
-->x4=h_M1*x1
 x4  =g^0d^3+g^1d^5+g^4d^7+(g^6d^+oo)[g^0d^+oo]*
```

Let us remark that the semiring $\overline{\mathbb{Z}}_{max}[\![\gamma]\!]$ has a quotient structure in which $1 \oplus 3\gamma^1 \oplus 4\gamma^5$ is equivalent to $1 \oplus 3\gamma^1 \oplus 3\gamma^2 \oplus 3\gamma^3 \oplus 3\gamma^4 \oplus 4\gamma^5$ (Baccelli *et al.*, 1992).

Example 9. (Transfer calculus). For the system depicted in Fig. 6 we have $h_1 = 4(4\gamma^5)^* \otimes 2 \otimes 2(2\gamma^3)^*$. The periodic form of h_1 can be computed as follows :

```
   a=series([eps],[0 4],[5 4]);
   b=series([eps],[0 2],[0 0]);
   c=series([eps],[0 2],[3 2]);
   h1=a*b*c
   -->h1  =(g^0d^8+g^3d^10+)[g^5d^4]*
```

Example 10. (Controller calculus for the system in Fig. 6). For the system depicted in Fig. 6, the computation of the transfer series (see previous example) leads to $h_1 = (8\gamma^0 \oplus 10\gamma^3)(4\gamma^5)^*$ and $h_2 = 14 \oplus (19\gamma^3 \oplus 22\gamma^4 \oplus 24\gamma^6)(8\gamma^4)^*$.

Controllers $\hat{f}_1$ and $\hat{f}_2$ given in Proposition 3 can be computed as follows :

```
h1=series([eps],[0,8;3,10],[5,4]);
h2=series([0,14],[3,19;4,22;6,24],[4,8]);
f1=series([eps],[13,0],[0,0])
f2=series([eps],[8,0],[0,0]);
alpha2=(h2*stargd(h1*f1));
hatf1=h1\stargd(h1*f1)
hatf2=alpha2\stargd(alpha2*f2)
-->hatf1=(g^13d^0+g^16d^2+)[g^5d^4]*
-->hatf2=g^8d^0+(g^11d^5+g^12d^8+g^14d^10+)[g^4d^8]*
```

ACKNOWLEDGMENT

The authors are grateful to V. Reverdy for her valuable linguistic help.