



HAL
open science

Pilotage d'une boucle KANBAN modélisée par réseau de Petri

Philippe Thomas, André Thomas, Sana Belmokhtar

► **To cite this version:**

Philippe Thomas, André Thomas, Sana Belmokhtar. Pilotage d'une boucle KANBAN modélisée par réseau de Petri. 8ème Conférence Internationale de Modélisation et Simulation, MOSIM'10, May 2010, Hammamet, Tunisie. pp.CDRom. hal-00525891

HAL Id: hal-00525891

<https://hal.science/hal-00525891>

Submitted on 13 Oct 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

PILOTAGE D'UNE BOUCLE KANBAN MODELISEE PAR RESEAU DE PETRI

Philippe THOMAS, André THOMAS, Sana BELMOKHTAR

Centre de Recherche en Automatique de Nancy (CRAN-UMR 7039), Nancy-Université, CNRS
ENSTIB 27 rue du merle blanc, B.P. 1041
88051 Epinal cedex 9 France
Philippe.thomas@cran.uhp-nancy.fr

RESUME : *Le développement et l'utilisation des systèmes Kanban a contribué au succès des approches Juste à Temps (JàT) dans l'industrie. Cependant, si dans un environnement où la demande reste stable, il apparaît que le Kanban traditionnel est efficace, une variation de cette même demande va conduire à rendre l'utilisation d'une boucle Kanban insuffisante. D'autre part, dans le cadre du concept Système Contrôlé par le Produit (SCP), le produit permet de véhiculer et de transmettre de l'information qu'une boucle Kanban traditionnelle n'est pas capable d'exploiter. D'autre part, très rapidement, l'utilisation des Réseaux de Petri (RdP) pour modéliser le fonctionnement des boucles Kanban a semblé une approche intéressante. L'objectif de ce travail est d'étudier la possibilité d'exploiter les relations analytiques intrinsèques aux RdP afin d'améliorer le contrôle d'une boucle Kanban en exploitant des données temps réelles en provenance du processus ou du produit. Pour cela, une commande par descente du gradient est utilisée.*

MOTS-CLES : *boucle Kanban, réseau de Petri, descente du gradient, système contrôlé par le produit, Juste à Temps, contrôle.*

1 INTRODUCTION

L'origine du Juste à Temps (JàT) remonte au courant des années soixante, plus particulièrement dans l'industrie Japonaise (Sugimori *et al.*, 1977 ; Monden, 1981 ; Di Mascolo *et al.* 1991). Le principe de fonctionnement du JàT consiste à produire un bien ou un composant seulement à l'instant où l'on en a besoin (Béranger, 1987).

La mise en place d'une approche JàT est souvent réalisée par l'intermédiaire d'un système kanban. Dans un système kanban, une carte kanban est attachée à chaque produit. Dans ce cadre, l'utilisation d'une pièce produite libère la carte kanban qui lui est attachée et qui va être attachée à une nouvelle pièce qui reste à produire. Cette libération de carte kanban induit un nouvel ordre de fabrication. Nous nous trouvons alors dans un système de flux tiré. Un système kanban est généralement divisé en plusieurs étages ou mailles (Di Mascolo *et al.*, 1991).

Les systèmes JàT ont originellement été conçus pour des environnements de production déterministes présentant une demande stable et des temps de processus constants. Dans ce cadre, les systèmes JàT présentent des performances optimales (Moore et Gupta, 1996) et le système Kanban traditionnel possède un nombre de cartes qui circulent dans le système constant. Il est démontré qu'un tel Kanban est efficace dans le contexte d'un environnement à demande et délais de fabrication stables. Dans un environnement instable (variabilité importante de la demande par exemple), un tel système kanban devient inadapté et il est nécessaire de le modifier. Pour cela, deux grandes approches existent. Il est possible de modifier le

nombre de cartes kanban présentes dans le système ou encore de modifier leur taille. On obtient alors le kanban dynamique (Tardif et Maasiedvag, 2001 ; Shahabudeen et Sivakumar, 2007).

Les récents progrès dans le domaine des technologies d'identification automatique (Auto-ID), comme par exemple l'identification par radiofréquence (RFID-Radio Frequency Identification), rendent envisageable leur utilisation au profit de la production en permettant la collecte, le stockage et la communication d'informations précises et en temps réel (Klein, 2008). Le produit, ou un élément du système physique (le Kanban, par exemple) instrumenté à l'aide des technologies RFID deviendrait « actif ». Le concept de « système contrôlé par le produit » (SCP) est considéré comme une manière différente de voir le contrôle d'un atelier ou, d'une manière générale, des flux physiques de produits et de matières. Selon ce concept, le produit peut porter, de manière permanente et continue (ou discontinue), des informations. Il peut aussi en donner à et/ou en capter de son environnement immédiat. Il peut enfin sur la base de ces informations, et faisant suite à un traitement qu'il opérerait lui-même, prendre des décisions relatives à son propre devenir et les faire appliquer par les éléments constitutifs de son environnement (Thomas, *et al.*, 2009). Il semble donc envisageable qu'une telle approche permette de collecter des informations permettant de contrôler en temps réel le système kanban.

D'autre part, depuis longtemps, les réseaux de Petri (RdP) ont montré qu'ils étaient utiles pour l'étude des systèmes à événements discrets (SED) et même des systèmes dynamiques hybrides (SDH) (Cassandras, 1993, Zaytoon *et al.* 1998 ; Lefebvre et Thomas, 2005), du fait

qu'ils associent une représentation graphique intuitive à des expressions analytiques (Brams, 1983 ; Brauer *et al.*, 1986 ; Murata, 1989). De ce fait de nombreux algorithmes utilisant la théorie de RdP ont été construits pour le contrôle des SED et des SDH. La modélisation de systèmes Kanban à l'aide de RdP ne fait pas exception. En effet, de nombreux modèles de systèmes kanban exploitant les RdP (Jain *et al.* 2006), les RdP temporisés (Di Mascolo *et al.*, 1991 ; Panayiotou et Cassandras, 1999 ; Giua *et al.*, 2002 ; Bohez, 2004 ; Seeluangsawat et Bohez, 2004), les RdP Stochastiques (Nakashima et Gupta, 2003) ou les RdP colorés (Moore et Gupta, 1996 ; 1999 ; Mohanty *et al.*, 2003 ; Desrochers et Pia Fanti, 2005 ; Ullah et Bohez, 2008a,b) ont été proposés.

Lorsque les dynamiques d'un système physique sont nombreuses, sa modélisation par un RdP discret (ordinaire, temporisé ou coloré) est décrite par un nombre important de marques. Il s'ensuit un nombre des marquages accessibles trop grand. Pour pallier à cette difficulté, David et Alla (1992) ont proposé d'utiliser des RdP continus à vitesse variables (RdPCV). De surcroit, des méthodes de transformation des RdPT en RdPCV et vice versa ont été proposées (David et alla, 1992).

D'autre part, de nombreux résultats ont été établis avec les RdPCV pour la commande des systèmes de production (Ghabri, 1995 ; Amrah *et al.*, 1996 ; Hanzalek, 2003 ; Silva et Recalde, 2003 ; Lefebvre *et al.*, 2004). De par la portée des résultats obtenus, cette approche apparaît comme une alternative intéressante à la commande discrète des SED (Lefebvre et Thomas, 2002).

La contribution principale de ce travail va donc être d'étudier la possibilité de transformer des modèles RdPT de systèmes kanban en RdPCV et d'exploiter une stratégie de commande des flux dans les RdPCV basée sur la propagation du gradient dans le réseau (Lefebvre *et al.* 2004).

Nous allons maintenant rappeler succinctement le fonctionnement des RdPT et des RdPCV en définissant les notations utilisées. Dans une troisième partie, nous allons présenter le modèle de système kanban que nous allons utiliser et nous en construirons l'équivalent à l'aide d'un RdPCV. Nous poursuivrons en présentant succinctement la stratégie de commande que nous emploierons dans la partie 5 où nous présenterons les résultats avant de conclure.

2 RESEAUX DE PETRI

Un RdP avec n places et p transitions est défini par le quintuplet $\langle P, T, \text{Pre}, \text{Post}, M_0 \rangle$ où $P = \{P_i\}_{i=1, \dots, n}$ est un ensemble de places, $T = \{T_j\}_{j=1, \dots, p}$ est un ensemble de transitions, tels que $P \cap T = \emptyset$ (Brams 1983, Murata 1989). \mathbb{N} est défini comme l'ensemble des entiers naturels. $\text{Pre} : P \times T \rightarrow \mathbb{N}$ est l'application de pre-incidence: $\text{Pre}(P_i, T_j)$ est le poids de l'arc reliant la place P_i à la

transition T_j et $W_{PR} = (w_{ij}^{PR})_{i=1, \dots, n, j=1, \dots, p} \in \mathbb{N}^{n \cdot p}$ avec $w_{ij}^{PR} = \text{Pre}(P_i, T_j)$ est la matrice de pre-incidence. $\text{Post} : P \times T \rightarrow \mathbb{N}$ est l'application de post-incidence : $\text{Post}(P_i, T_j)$ est le poids de l'arc reliant la transition T_j à la place P_i et $W_{PO} = (w_{ij}^{PO})_{i=1, \dots, n, j=1, \dots, p} \in \mathbb{N}^{n \cdot p}$ avec $w_{ij}^{PO} = \text{Post}(P_i, T_j)$ est la matrice de post-incidence. La matrice d'incidence W du RdP est définie par $W = W_{PO} - W_{PR} \in \mathbb{N}^{n \cdot p}$.

Soit $M(t) = (m_i(t))_{i=1, \dots, n} \in \mathbb{N}^n$ le vecteur de marquage à l'instant t et $M_0 \in \mathbb{N}^n$ le vecteur de marquage initial. \mathcal{T}_j (resp T_j°) correspond à l'ensemble des places d'entrée (resp. l'ensemble des places de sortie) de T_j . Une séquence de franchissement est définie comme une suite ordonnée de transitions qui sont successivement franchies à partir du marquage M jusqu'au marquage M' . Une telle séquence est usuellement représentée par son vecteur caractéristique $X = (x_j)_{j=1, \dots, p} \in \mathbb{N}^p$ où x_j correspond au nombre de franchissements de la transition T_j .

$$M' = M + W.X \quad (1)$$

Le vecteur X est calculé en appliquant les règles de franchissement qui dictent la dynamique du RdP. Dans les RdP autonomes sans conflit, les composantes x_j de X sont calculées par l'équation suivante :

$$x_j = \min_{P_i \in \mathcal{T}_j} \left(\text{fix} \left(\frac{m_i}{w_{ij}^{PR}} \right) \right) \quad (2)$$

où $\text{fix}(\cdot)$ correspond à la partie entière de (\cdot) .

2.1 Réseaux de Petri temporisés RdPT

Deux types de RdP temporisés existent. Les RdP T-temporisés et les RdP P-temporisés (David et Alla, 1992). Dans ce papier, seul les RdP T-temporisés (RdPT) sont utilisés. Un RdPT est défini par le couple $\langle PN, \text{Tempo} \rangle$ tel que PN est un RdP ordinaire marqué et tempo est une application de l'ensemble T des transitions dans \mathbb{R}^+ (\mathbb{R}^+ représente l'ensemble des réels non négatifs). $\text{Tempo}(T_j) = d_j$ représente la temporisation associée à la transition T_j .

2.2 Réseaux de Petri continus RdPCV

Un RdP continu est une extension des RdP, déduite d'un réseau de Petri discret temporisé sur les transitions (RdPT) (Ramchandani, 1973). Les RdP continus ont été développés pour établir des approximations continues du comportement discret des SED (David et Alla, 1992). Un RdP continu avec n places et p transitions est défini par le couple $\langle PN, X_{max} \rangle$ ou PN est un réseau de Petri et $X_{max} = (x_{max,j})_{j=1, \dots, p} \in \mathbb{R}^p$ est le vecteur des fréquences maximales de franchissement. Le marquage $m_i(t) \in \mathbb{R}^+$ de chaque place P_i , $i = 1, \dots, n$, a une valeur réelle non négative et le franchissement de chaque transition est un

processus continu. Soit $X(t) = (x_j(t))_{j=1,\dots,p} \in \mathbb{R}^{+p}$ le vecteur des fréquences de franchissement à l'instant t . La dynamique du marquage est donnée par (3) :

$$\frac{dM(t)}{dt} = W.X(t). \quad (3)$$

Parmi les extensions de RdP continus, les RdP continus avec des vitesses variables (RdPCV) sont la meilleure approximation continue des RdPT (David et Alla, 1992). Les composantes du vecteur $M(t)$ sont des fonctions continues du temps, et les composantes du vecteur $X(t)$ dépendent continûment du marquage des places :

$$x_j(t) = x_{\max j} \cdot \mu_j(t) \quad (4)$$

avec :

$$\mu_j(t) = \min_{P_i \in {}^oT_j} (m_i(t)) \quad (5)$$

(Un bouclage Une self-loop) est usuellement associée à chaque transition T_j pour limiter le nombre de franchissements simultanés.

2.3 Représentation dans l'espace d'état

L'équation (5) présente des commutations dues à la fonction "min", et (4) fait apparaître des produits entre le vecteur d'état et le vecteur des entrées. Pour ces raisons, les RdPCV ne peuvent être décrits à l'aide d'un modèle d'état linéaire. En revanche, une représentation bilinéaire par morceaux peut en être donnée (Lefebvre *et al.*, 2003).

L'ensemble des transitions T est divisé en 2 sous ensembles disjoints T_C , et T_{NC} tels que $T = T_C \cup T_{NC}$. T_C est l'ensemble des transitions commandables et T_{NC} celui des transitions non commandables. On peut définir une matrice de permutation D telle que :

$$D^{-1}.X(t) = \begin{pmatrix} X_C(t) \\ X_{NC}(t) \end{pmatrix} \quad (6)$$

où $X_C(t) \in \mathbb{R}^{+d}$ et $X_{NC}(t) \in \mathbb{R}^{+p-d}$ sont respectivement les vecteurs des fréquences de franchissement des transitions commandables et non commandables. Le vecteur des entrées de commande $U(t) = X_{\max C}(t) \in \mathbb{R}^{+d}$ correspond aux fréquences maximales de franchissement des transitions commandables. (les fréquences maximales des transitions non commandables $X_{\max NC}$ sont supposées constantes). Le vecteur des entrées est contraint $0 \leq U(t) \leq U_{\max}$ pour limiter les fréquences de franchissement à un intervalle positif et borné. Les composantes du vecteur des sorties $Y(t) \in \mathbb{R}^{+e}$ sont définies comme des sommes pondérées de sous ensembles de

places qui sont supposées mesurables. Soit $Q = (q_{\alpha i})_{\alpha=1,\dots,e; i=1,\dots,n} \in \mathbb{R}^{e.n}$ une matrice de projection dont les termes sont tous non négatifs. L'équation (3) peut être réécrite :

$$\begin{aligned} \frac{dM(t)}{dt} &= W_{NC}.X_{NC}(t) + W_C.X_C(t) \\ Y(t) &= Q.M(t) \end{aligned} \quad (7)$$

Plusieurs phases apparaissent successivement dans la réponse temporelle du RdPCV. Chaque phase φ est active pendant la durée qui sépare deux commutations successives des opérateurs « min » dans (5) et correspond à une configuration particulière de ces opérateurs caractérisée par les p fonctions de classification f_j ($\forall T_j \in T^\circ$) :

$$\begin{aligned} f_j : \mathbb{R}^{+e} &\rightarrow \{1, \dots, n\} \\ M(t) &\rightarrow f_j(M(t)) \text{ telle que } m_{f_j}(t) = \mu_j(t) \end{aligned} \quad (8)$$

Chaque fonction f_j spécifie la place d'entrée de T_j de marquage minimal. Au cours de chaque phase φ il existe une relation constante entre les composantes des vecteurs $X_C(t)$ et $M(t)$ ainsi que $X_{NC}(t)$ et $M(t)$ que l'on peut exprimer sous forme scalaire à l'aide des fonctions f_j ou sous forme vectorielle à l'aide de familles de vecteurs lignes $A_j(\varphi) \in \{0,1\}^{1.n}$ et $B_j(\varphi) \in \{0,1\}^{1.p-d}$ constants sur chaque phase mais qui varie d'une phase à la suivante :

$$\begin{aligned} x_{C_j}(t) &= u_j(t).A_j(\varphi).M(t), \quad j = 1, \dots, d \\ x_{NC_j}(t) &= x_{\max NC_j}.B_j(\varphi).M(t), \quad j = 1, \dots, p-d \end{aligned} \quad (9)$$

L'équation (7) peut être transformée sous forme vectorielle :

$$\begin{aligned} \frac{dM(t)}{dt} &= \left(\sum_{j=1}^d u_j(t).W_{C_j}.A_j(\varphi) + \sum_{j=1}^{p-d} x_{\max NC_j}.W_{NC_j}.B_j(\varphi) \right).M(t) \\ Y(t) &= Q.M(t), \quad \varphi = 1, 2, \dots \end{aligned} \quad (10)$$

en désignant par W_{C_j} les colonnes de la matrice W_C et par W_{NC_j} celles de W_{NC} , (10) est une représentation bilinéaire par morceaux du RdPCV (3) (Lefebvre *et al.*, 2004). Chaque phase est caractérisée par une famille de matrices $W_{C_j}.A_j(\varphi) \in \mathbb{N}^{n.n}$ associée aux transitions commandables et $W_{NC_j}.B_j(\varphi) \in \mathbb{N}^{n.n}$ associée aux transitions non commandables.

3 MODÈLES RDP DE SYSTÈMES KANBAN

Comme présenté dans l'introduction, de nombreux modèles de systèmes kanban utilisant les réseaux de Petri

ont été proposés. Dans le cadre de ce papier, un modèle RdPT proposés par Di Mascolo *et al.* (1991) est plus particulièrement étudié. Il est cependant notable que l'approche proposée peut être appliquée à tout modèle utilisant des RdPT et même à la plupart des modèles utilisant des RdP Colorés car il est toujours possible, à partir d'un RdP coloré, d'en construire un modèle équivalent à l'aide de RdPT (Jensen, 1996).

Le système kanban considéré suit les hypothèses suivantes :

- chaque étage est constitué d'une installation unique comprenant une ou plusieurs machines identiques,
- chaque étage a, au plus, un étage prédécesseur et un étage successeur,
- le système produit un et un seul type de pièces,
- la taille de tous les kanbans est la même,
- quand un kanban est détaché d'une pièce utilisée par l'étage suivant, il est immédiatement disponible (pas de délai) pour être attaché à une pièce de l'étage précédent.

Le fonctionnement d'un tel système peut être décrit par la figure 1 qui se focalise sur l'étage i et présente les flux de produits et de kanban entre les étages $i-1$, i et $i+1$.

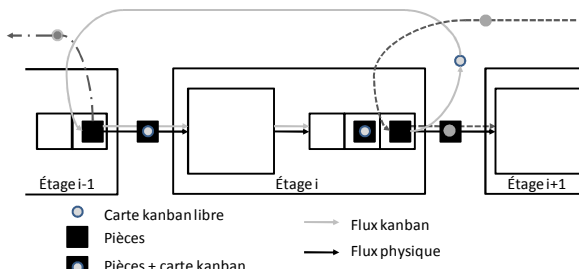


Figure 1 : Liens entre les différents étages

Un tel système kanban est modélisé par Di Mascolo *et al.* (1991) à l'aide du RdPT présenté figure 2.

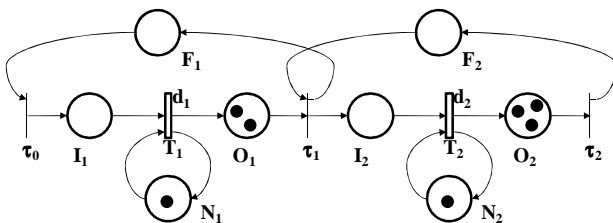


Figure 2 : Modèle RdPT pour un système kanban à 2 étages

Si on considère l'étage i (figure 1). Chaque kanban peut se trouver dans trois états distincts :

- il est libre et attend d'être attaché à une nouvelle pièce de l'étage $i-1$,
- il est attaché à une pièce qui se trouve dans le process de production,
- il est attaché à une pièce présente dans le stock de sortie.

De ce fait, dans le modèle RdPT (figure 2), ces trois états sont représentés par trois places dénotées respectivement F_i , I_i , et O_i . Le nombre de jetons dans la place I_i représente le nombre de pièces dans le process de production. Le nombre de jetons dans la place O_i représente le nombre de pièces présentes dans le stock de sortie quand celui dans F_i représente le nombre de cartes kanbans libres. La transition T_i placée entre les places I_i , et O_i est une transition temporisée dont le délai d_i représente le temps opératoire de l'étage i . Le nombre de jetons dans la place N_i correspond au nombre de machines identiques présentes dans l'étage i . La transition τ_i représente la synchronisation entre les étages i et $i+1$. Cette transition est une transition immédiate.

Le RdPT présenté figure 2 correspond au cas où la demande et l'approvisionnement sont supposés infinis. Di Mascolo *et al.* (1991) ont proposé diverses variations sur ce même modèle pour prendre en compte, une demande limitée, un approvisionnement limité ou encore des demandes perdues. Seul ce premier modèle sera exploité ici afin de simplifier la présentation et la compréhension du papier.

Notre objectif est ici de piloter le système kanban considéré en utilisant son modèle RdP et en exploitant des techniques de contrôles des RdPCV. Pour ce faire, il est nécessaire de convertir le modèle RdPT présenté figure 2 en son modèle équivalent RdPCV. Le passage d'un RdPT à un RdPCV est toujours possible et pour ce faire le modèle doit subir quelques transformations (David et Alla, 1992).

La première difficulté à résoudre pour transformer le RdPT de la figure 2 en un RdPCV est que les transitions τ_0 , τ_1 et τ_2 ne sont pas temporisées mais sont des transitions immédiates. Pour résoudre cette première difficulté deux approches peuvent être utilisées. La première est de considérer ces transitions non pas comme immédiates mais comme étant temporisées avec un délai court. L'autre approche est de considérer que puisque ces transitions sont immédiates, tout jeton dans la place O_i va immédiatement passé dans la place I_{i+1} . Il est donc possible de confondre ces deux places ce qui permet de ne conserver que les transitions temporisées. Cette deuxième approche, plus concise, sera employée ici.

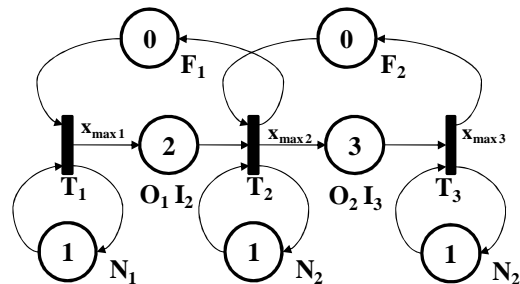


Figure 3 : Modèle RdPCV du système kanban considéré

Ceci fait, toutes les transitions présentes dans le modèle sont bien temporisées et il est possible d'appliquer les

méthodes de transformations. Dans ce cadre, tous les marquages discrets des places doivent être remplacés par un marquage continu. D'autre part les délais associés à chacune des transitions temporisées du RdPT permettent de déterminer les vitesses maximales de franchissement des transitions du RdPCV par la relation :

$$x_{\max_i} = \frac{1}{d_i} \quad (11)$$

Ces transformations nous permettent de construire de RdPCV de la figure 3 qui est une approximation du RdPT de la figure 2.

4 COMMANDE DES VCPN

La commande des flux avec les réseaux de Petri continus a été étudiée par de nombreux auteurs (Ghabri, 1995 ; Amrah *et al.*, 1996 ; Hanzalek, 2003 ; Lefebvre et Thomas, 2002 ; Silva et Recalde, 2003 ; Lefebvre *et al.*, 2003, 2004). La stratégie de commande employée ici est celle proposée par Lefebvre *et al.* (2003) qui est inspirée des algorithmes d'adaptation des réseaux de neurones. Elle tire profit de la propagation du gradient au travers du graphe orienté sous jacent au VCPN pour minimiser le carré de l'erreur instantanée entre les sorties désirées et les sorties mesurées en modifiant les fréquences maximales de franchissement des transitions commandables.

Les algorithmes d'adaptation basés sur la propagation du gradient sont généralement obtenus en minimisant un coût scalaire qui mesure la distance entre la sortie désirée $Y_d(t)$ et la sortie mesurée $Y(t)$ du système. Ces algorithmes ont été utilisés avec succès pour l'identification et la commande des systèmes continus (Ljung, 1987). Nous allons présenter succinctement l'algorithme utilisé.

4.1 Le régulateur

Dans un souci de simplification, nous présenterons dans un premier temps le régulateur dans le cas mono sortie. L'erreur instantanée est définie par $\varepsilon(k) = y_d(k) - y(k)$, où $y_d(k)$ correspond à la sortie désirée à l'instant $t = k.\Delta t$, et $y(k)$ correspond à la sortie mesurée du VCPN au même instant. Avec une méthode d'intégration numérique du premier ordre, on peut écrire :

$$y(k) = y(k-1) + \Delta t. \left(\sum_{j=1}^d u_j(k-1).Q.W_{C_j}.A_j(\varphi) + \sum_{j=1}^{p-d} x_{\max_{NC_j}}.Q.W_{NC_j}.B_j(\varphi).M(k-1) \right) \quad k > 0 \quad (12)$$

Soit le coût scalaire $v(k)$:

$$v(k) = \frac{1}{2} \varepsilon^2(k) \in \mathbb{R} \quad (13)$$

L'actualisation à chaque nouvel échantillon de mesures du vecteur $U(k)$ des entrées est obtenue en écrivant un développement en série de Taylor du coût v . Les termes de second ordre sont habituellement négligés dans l'évaluation de la matrice Hessienne, mais un petit terme positif $\lambda.I$ est ajouté pour permettre l'inversion de la matrice lorsqu'elle est singulière ou mal conditionnée. L'actualisation de l'entrée de commande est donnée par :

$$\delta U(k) = -(S(k).S(k)^T + \lambda.I)^{-1}.S(k).\varepsilon(k), \quad k \geq 0 \quad (14)$$

avec $\delta U(k) = U(k) - U(k-1)$. $S(k) \in \mathbb{R}^d$ est le gradient de la sortie par rapport aux variations des entrées, calculées à l'instant $t = k.\Delta t$.

L'algorithme de commande précédent peut être étendu au cas multi-sorties en considérant la fonction coût :

$$v(k) = \frac{1}{2} \sum_{\alpha=1}^e \varepsilon_{\alpha}^2(k) \in \mathbb{R} \quad (15)$$

en lieu et place de celui décrit par (13). Le développement en série de Taylor de (15) est utilisé pour actualiser le vecteur des entrées $U(k)$ à chaque nouvel échantillon de mesures :

$$\delta U(k) = - \left(\sum_{\alpha=1}^e (S_{\alpha}(k).S_{\alpha}(k)^T + \lambda.I) \right)^{-1} \cdot \left(\sum_{\alpha=1}^e (S_{\alpha}(k).\varepsilon_{\alpha}(k)) \right) \quad (16)$$

où $S_{\alpha}(k) \in \mathbb{R}^d$ est le gradient de la sortie y_{α} par rapport aux variations du vecteur U des entrées, calculé à l'instant $t = k.\Delta t$.

Pour converger efficacement vers un minimum de la fonction coût, les algorithmes d'adaptation basés sur le calcul du gradient peuvent être itérés plusieurs fois sur chaque nouvel échantillon de mesure (Lefebvre *et al.*, 2004). Pour pouvoir utiliser cet algorithme, il reste à déterminer la valeur du gradient $S_{\alpha}(k)$ utilisé dans (16).

4.2 Calcul du gradient

Le calcul du gradient revient à calculer la sensibilité de chaque sortie y_{α} par rapport aux variations des entrées de commande U . Le gradient doit être calculé en temps échantillonné pour pouvoir appliquer le résultat dans l'algorithme précédent.

Soit $S_{\alpha\gamma}(k)$ le gradient de la sortie y_{α} par rapport à l'entrée u_{γ} (i.e. la fréquence maximale de franchissement $x_{\max_{c_{\gamma}}}$). Le gradient $S_{\alpha\gamma}(k)$ est calculé par une mé-

thode du premier ordre en utilisant les fonctions de classifications f_j définies par (8) :

$$\delta s_{\alpha\gamma}(k) = \sum_{i=1}^n q_{\alpha i} \cdot \left(w_{i\gamma} \cdot \mu_{\gamma}(k) + \sum_{\substack{j=1 \\ j \neq \gamma}}^p w_{ij} \cdot x_{\max_j} \cdot s_{f_j\gamma}(k) \right) \cdot \Delta t \quad (17)$$

où $q_{\alpha i}$ est l'élément de la $i^{\text{ième}}$ ligne et $\alpha^{\text{ième}}$ colonne de la matrice Q. $\delta s_{\alpha\gamma}(k)$ représente la variation du gradient $\delta s_{\alpha\gamma}(k) = s_{\alpha\gamma}(k) - s_{\alpha\gamma}(k-1)$ entre les instants k-1 et k :

$$\begin{cases} \delta s_{\alpha\gamma}(k) = s_{\alpha\gamma}(k) - s_{\alpha\gamma}(k-1) \\ s_{\alpha\gamma}(0) = 0 \end{cases} \quad (18)$$

5 EXEMPLE

Nous nous intéressons ici au pilotage du système kanban dont un étage est présenté figure 1 et qui est modélisé par le RdPT de la figure 2 (système kanban à 2 étages). Nous avons vu dans la partie 3 qu'il est possible de construire une approximation continue de ce modèle RdPT en exploitant les RdPCV et le résultat de cette approximation est le modèle représenté figure 3. La matrice d'incidence W de ce RdPCV est :

$$W = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} \quad (19)$$

Les pièces entrent dans le système par l'intermédiaire de la transition T_1 . Les fréquences de franchissement sont limitées par le marquage des places N_i . La transition de sortie T_3 permet de représenter la consommation des pièces finies par le système aval. Les transitions T_1 et T_2 sont les deux transitions qui nous permettent de contrôler notre système quand la transition T_3 correspond à la contrainte demande qu'il faut satisfaire. Le paramètre de réglage de l'algorithme de contrôle λ est $\lambda = 1$ et on choisit une période d'échantillonnage $\Delta t = 0,0754$. En l'absence de commande, le système évolue librement selon :

$$\begin{aligned} \dot{m}_1(t) &= x_{\max_0} \cdot \min(m''_0(t), m'_1(t)) \\ &\quad - x_{\max_1} \cdot \min(m''_1(t), m_1(t), m'_2(t)) \\ \dot{m}_2(t) &= x_{\max_1} \cdot \min(m''_1(t), m_1(t), m'_2(t)) \\ &\quad - x_{\max_2} \cdot \min(m''_2(t), m_2(t)) \\ m'_i(t) &= C_i - m_i(t) = 3 - m_i(t) \quad i = 1, 2 \\ m''_j(t) &= m''_j(0) = 1 \quad j = 1, 2, 3 \end{aligned} \quad (20)$$

avec m_i représentant le marquage de la place $O_i I_{i+1}$, m'_i représentant le marquage de la place F_i et m''_i le marquage de la place N_i . C_i représente le nombre de cartes kanbans en service dans l'étage i ($C_i = m_i(0) + m'_i(0)$).

Considérant à présent le système (20) comme un système bilinéaire par morceaux avec une commande contrainte $U(t)$ (entre 0 et 10) correspondant aux fréquences maximales de franchissement des transitions commandables T_1 et T_2 (i.e. $U(t) = [x_{\max_1}(t), x_{\max_2}(t)]^T$), le RdPCV peut être représenté sous forme scalaire en utilisant les fonctions f_j (8).

La transition T_3 représentant la contrainte demande de notre système, cette transition est donc forcément commandable. Afin de simuler cette demande nous avons choisi de fixer la vitesse de franchissement de T_3 , x_{\max_3} à 2.

Une fois défini complètement le modèle RdPCV, il nous reste à définir l'objectif de son pilotage. Nous sommes en présence d'un système kanban, l'objectif principal d'un tel système est d'être toujours capable de répondre à la demande. Pour ce faire, notre objectif va être de faire converger le marquage m_2 de la place $I_2 O_3$ vers une valeur légèrement supérieure à 1 de manière à toujours être en mesure de répondre à une demande. La sortie mesurée y de notre système va donc être le marquage de la place $I_2 O_3$ que l'on va faire tendre vers la valeur de sortie désirée $y_d = 1,2$.

Dans un premier temps, le marquage initial M_0 est fixé à :

$$M_0 = (1, 1, 1, 2, 3, 0, 0)^T \quad (21)$$

Ce marquage correspond à un système où toutes les cartes kanbans sont associées à des pièces en attente d'être consommées par le système aval, ce qui fait que les marquages des places F_1 et F_2 sont tous les deux à 0 quand les marquages des places $O_1 I_2$ et $O_2 I_3$ sont respectivement à 2 et à 3, à l'état initial.

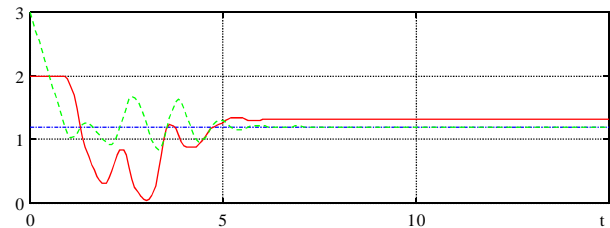


Figure 4 : marquages mesurés des places m_1 (continu) et m_2 (tiret)

La figure 4 présente le marquage des places $O_1 I_2$ et $O_2 I_3$ au cours du temps. On constate immédiatement que notre objectif de 1,2 jeton dans la place $O_2 I_3$ est atteint

très rapidement dès l'instant 5 et reste stable jusqu'à la fin de la période d'observation. D'autre part, le marquage de la place O_1I_2 se stabilise également très rapidement à une valeur très légèrement supérieure à 1,3.

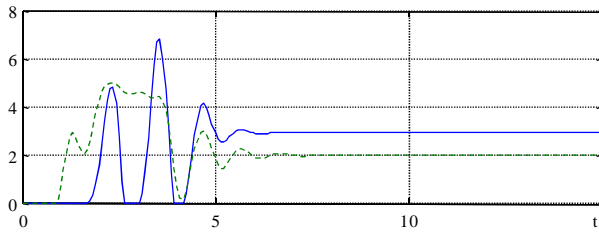


Figure 5 : Fréquences maximales de franchissement des transitions T_1 (continu) et T_2 (tiret)

La figure 5 montre l'évolution des commandes de notre système au cours du temps que sont les fréquences maximales de franchissement $[x_{\max_1}(t), x_{\max_2}(t)]^T$. Là aussi, on voit qu'après une période d'adaptation assez courte, ces commandes se stabilisent pour répondre à notre objectif.

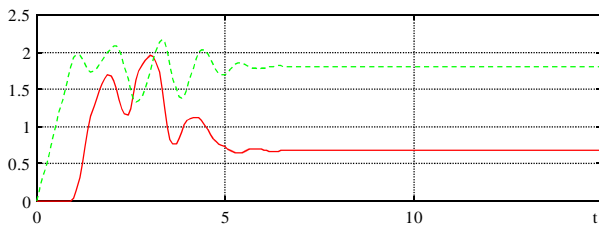


Figure 6 : marquages mesurés des places m'_1 (continu) et m'_2 (tiret)

La figure 6 présente sur le même exemple l'évolution du nombre de cartes kanbans libres dans le système. On constate immédiatement que la stratégie de pilotage proposée permet de limiter les stocks d'en cours en gérant de manière dynamique le nombre de cartes kanbans à utiliser dans le système.

Nous pouvons nous intéresser à un deuxième jeu de conditions initiales décrit par le nouveau vecteur de marquage initial M_{01} :

$$M_{01} = (1, 1, 1, 0, 0, 2, 3)^T \quad (22)$$

Ce marquage correspond à un système où toutes les cartes kanbans sont libres et donc, où tous les stocks d'en cours sont vides. La figure 7 présente le résultat de l'évolution du marquage des places O_1I_2 et O_2I_3 au cours du temps avec ce nouveau marquage initial.

Nous pouvons noter que notre objectif de 1,2 jeton dans la place O_2I_3 est atteint encore plus rapidement que dans le cas précédent et reste stable jusqu'à la fin de la période d'observation. Par contre, le marquage de la place O_1I_2 converge vers une valeur de 1,68 jeton à comparer au 1,3 précédemment trouvé. Ceci est du au

fait que le marquage de cette place ne fait pas partie de l'ensemble des sorties contrôlées ce qui fait qu'elle peut évoluer totalement librement.

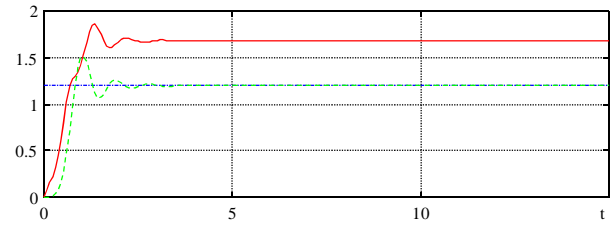


Figure 7 : marquages mesurés des places m_1 (continu) et m_2 (tiret) - marquage initial M_{01}

Maintenant, nous allons à nouveau considérer que le vecteur de marquage initial est le vecteur M_0 défini par (21). Par contre la demande représentée par la transition T_3 ne sera plus considérée comme une constante mais va doubler à partir de l'instant 10.

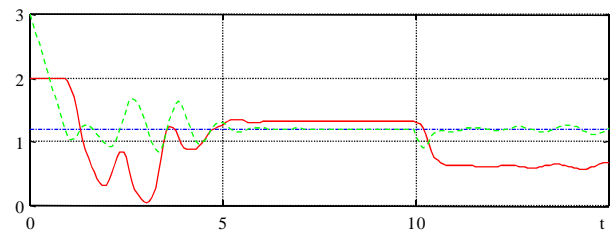


Figure 8 : marquages mesurés des places m_1 (continu) et m_2 (tiret) - évolution de x_{\max_3}

La figure 8 présente l'évolution du marquage des places m_1 et m_2 , ce qui revient à dire que la figure 8 représente l'évolution du stock d'en cours du système au cours du temps. Nous pouvons constater qu'à l'instant 10, l'évolution de la demande (x_{\max_3} est doublé) conduit à une légère diminution du nombre de jeton dans la place O_2I_3 mais qui reste très fugitive, puisque dès l'instant 10,6, le marquage de cette place tend à nouveau vers sa valeur désirée de 1,2. En fait, une grande partie de cette augmentation de demande est absorbée par le stock d'en cours représenté par la place O_1I_2 .

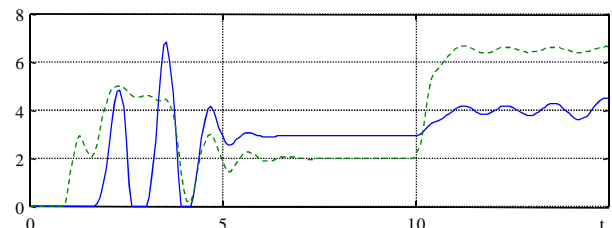


Figure 9 : Fréquences maximales de franchissement des transitions T_1 (continu) et T_2 (tiret) - évolution de x_{\max_3}

La figure 9 qui présente l'évolution de la commande au cours du temps, montre bien que l'essentiel de l'évolution de cette commande se fait en augmentant fortement la fréquence de franchissement x_{\max_2} qui passe de 2 à un peu plus de 6 tandis que cette augmentation est en partie associée à une augmentation de x_{\max_1} qui passe

de 3 à 4. Ce fonctionnement est confirmé par l'étude du marquage des places F_1 et F_2 présenté par la figure 10.

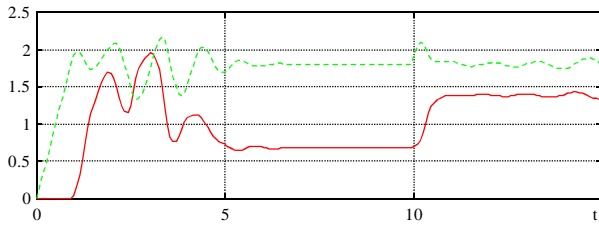


Figure 10 : marquages mesurés des places m'_1 (continu) et m'_2 (tiret) – évolution de $x_{max 3}$

L'évolution du marquage m'_1 de F_1 montre bien que c'est ce stock intermédiaire qui sert à compenser l'augmentation de la demande.

Nous pouvons maintenant nous intéresser à nouveau au cas simple où $x_{max 3}$ reste constant et où le marquage initial M_0 est défini par (21). Pour réaliser le contrôle de notre système, nous avons dû construire une approximation continue d'un modèle RdPT. Ce passage d'un RdPT à un RdPCV est tout à fait réversible et est tout à fait possible de retrouver les délais à affecter aux transitions T_1 et T_2 du RdPT de la figure 2 à partir des vitesses maximales de franchissement $x_{max 1}$ et $x_{max 2}$ présentées figure 5. Cette transformation peut se faire en exploitant à nouveau l'expression (11). Une autre approche est d'exploiter la structure du RdPCV présenté figure 11, dont la fréquence maximale de franchissement de la transition source T est tout simplement le signal de commande que l'on souhaite convertir en délais pour commander un RdPT. Pour obtenir ces délais, il suffit d'extraire les instants où le marquage m de la place P est un entier.

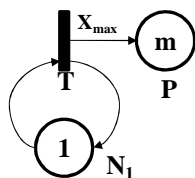


Figure 11 : RdPCV convertisseur de commande

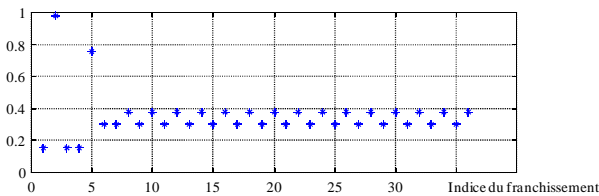


Figure 12 : séquence de délais pour la transition T_1 du RdPT

Lorsque l'on applique cette approche aux fréquences maximales de franchissement $x_{max 1}$ et $x_{max 2}$ on obtient les séquences de délais présentés par les figures 12 et 13.

Ces deux séquences de délais peuvent alors être appliquées aux transitions T_1 et T_2 du RdPT de la figure

2. Le marquage de la place O_2 que l'on cherche à contrôler est présenté figure 14.

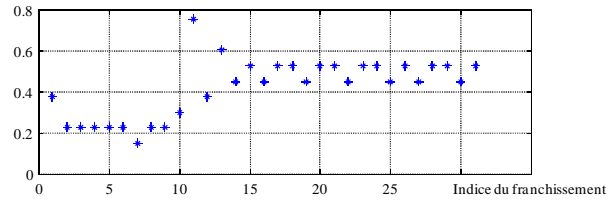


Figure 13 : séquence de délais pour la transition T_2 du RdPT

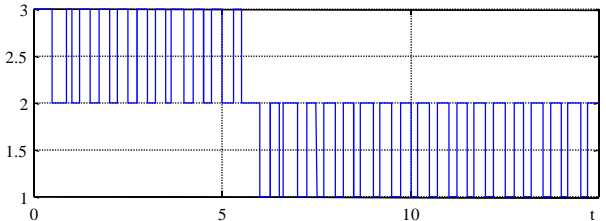


Figure 14 : marquage de la place O_2 du RdPT

L'évolution du marquage de cette place O_2 du RdPT est cohérente avec celle observée pour la place O_2I_3 du RdPCV présentée figure 4. Nous retrouvons bien un nombre de jetons présents dans cette place qui va tendre vers une valeur moyenne comprise entre 1 et 2 sachant que l'objectif que l'on s'était fixé dans le RdPCV était de 1,2.

6 CONCLUSION

Au cours de ce travail, nous avons proposé une nouvelle approche de pilotage des systèmes kanbans basée sur l'utilisation de modèles RdP. Nous avons montré comment transformer un modèle RdPT d'un système kanban en un modèle RdPCV avec lequel nous pouvons utiliser des stratégies de contrôle.

Nous avons testé une stratégie de contrôle particulière basée sur le principe de propagation du gradient dans le réseau. Nous avons vu que cette approche permet d'adapter automatiquement le nombre de cartes kanbans à utiliser dans le système en fonction de l'évolution de la demande. Cette approche semble donc une alternative intéressante aux stratégies de kanban dynamique.

Aux cours de travaux futurs, nous allons étudier la possibilité d'exploiter ce type d'approche pour dimensionner des systèmes kanbans. D'autre part, pour pouvoir passer à une application industrielle, il est nécessaire d'étudier des systèmes plus complexes et surtout pouvant traiter divers types de pièces. De tels systèmes vont nécessiter des modèles RdP plus complexes ce qui nécessitera vraisemblablement une adaptation de l'approche proposée.

REFERENCES

- Amrah, A., N. Zerhouni and A. El Moudni, 1996. Constrained State Feedback Control of a Class of Discrete Event Systems Modelled by Continuous Petri Nets. *ICARV'96*, Singapore, p. 979-984.
- Beranger, P., 1987. *Les Nouvelles Règles de la Production*, Dunod Ed., Paris.
- Bohez, E.L.J., 2004. A new Generic Timed Petri Net Model for Design and Performance Analysis of a Dual Kanban FMS. *International Journal of Production Research*, 42(4), p. 719-740.
- Brams, G.W., 1983. *Réseaux de Petri*, Vol I et II, Masson, Paris.
- Brauer, W., W. Reisig and G. Rosenberg, 1986. Petri Nets: Central Models and their Properties. *Lecture Notes in Computer Science*, 254.
- Cassandras, C.G., 1993. *Discret Event Systems: Modeling and Performance Analysis*, Asken Ass. Inc. Pub.
- David, R. and H. Alla, 1992. *Petri Nets and Grafset – Tools for Modelling Discrete Events Systems*, Prentice Hall, London.
- Desrochers, A. and M. Pia Fanti, 2005. A Supply Chain Model Using Complex-Valued Token Petri Nets. *Proc. of the 2005 IEEE Int. Conf. on Robotics and Automation*, Barcelona, Spain, p. 3709-3714.
- Di Mascolo, M., Y. Frein, Y. Dallery and R. David, 1991. A Unified Modeling of Kanban Systems Using Petri Nets. *The International Journal of Flexible Manufacturing Systems*, 3, p. 275-307.
- Ghabri, M., 1995. *Sur la Modélisation et la Commande des Systèmes Flexibles de Production*. Thèse de Doctorat, INPG, Grenoble, France.
- Giua, A., A. Piccaluga and C. Seatzu, 2002. Firing Rate Optimization of Cyclic Event Graphs by Token Allocations. *Automatica*, 38, p. 91-103.
- Hanzalek, Z., 2003. Continuous Petri Nets and Polytopes. *IEEE-SMC'03*, Washington, USA, p. 1513-1520.
- Jain, A., P.K. Jain and I.P.Singh, 2006. Performance Modelling of FMS with Flexible Process Plans – A Petri Net Approach. *International Journal of Simulation Modelling*, 5(3), p. 101-113.
- Jensen, K., 1996. *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use*, Vol I, seconde édition, Springer Verlag, Berlin.
- Klein, T., 2008. *Le Kanban Actif pour Assurer l'Interopérabilité Décisionnelle Centralisée / Distribuée*. Thèse de Doctorat, Université Henri Poincaré Nancy I, France.
- Lefebvre, D. and P. Thomas, 2002. Identification par phase des fréquences maximales de franchissement dans les réseaux de Pétri continus à vitesse variable. *CIMASI'02*, Casablanca, Maroc.
- Lefebvre, D., E. Leclercq, F. Druaux and P. Thomas, 2003. Source and sink transitions controllers for continuous Petri nets: a gradient - based approach. *ADHS'03*, Saint Malo, France.
- Lefebvre, D., E. Leclercq, F. Druaux and P. Thomas, 2004. Commande des flux dans les réseaux de Petri continus par propagation du gradient. *CIFA'04*, Douz, Tunisie.
- Lefebvre, D. and P. Thomas, 2005. Parameters estimation for timed and continuous Petri nets: application to the identification and monitoring of hybrid systems. *Cybernetics and Systems*, 36(3), p. 217-250.
- Ljung, L., 1987. *System Identification: Theory for the User*, Prentice Hall, Englewood Cliffs.
- Monden, Y., 1981. How Toyota Shortened Supply Lot Production Time, Waiting time and Conveyance Time. *Industrial Engineering*, 13(9), p. 22-30.
- Moore, K.E. and S. M. Gupta, 1996. Petri Net Models of Flexible and Automated Manufacturing Systems: a survey. *Industrial Journal Production Research*, 34(11), p. 3001-3035.
- Moore, K.E. and S. M. Gupta, 1999. Stochastic Coloured Petri Net (SCPN) Models of Traditional and Flexible Kanban Systems. *International Journal of Production Research*, 37(9), p. 2135-2158.
- Mohanty, R.P., S. Kumar and M.K. Tiwari, 2003. Expert Enhanced Coloured Fuzzy Petri Net Models of Traditional, Flexible and Reconfigurable Kanban System. *Production Planning and Control*, 14(5), p. 459-477.
- Murata, T., 1989. Petri Nets: Properties, Analysis and Application. *Proceedings IEEE*, 77(4), p. 541-580.
- Nakashima, K. and S.M. Gupta, 2003. Performance Evaluation of a Supplier Management System with Stochastic Variability. *International Journal*

- Manufacturing Technology and Management*, 5(1/2), p. 28-37.
- Panayiotou, C.G. and C.G. Cassandras, 1999. Optimization of Kanban-based Manufacturing Systems. *Automatica*, 35, p. 1521-1533.
- Ramchandani, C., 1973. *Analysis of Asynchronous Concurrent Systems by Timed Petri nets*. PhD Thesis, MIT, USA.
- Seelungsawat, R. and E.L.J. Bohez, 2004. Integration of JIT Flexible Manufacturing, Assembly and Disassembly System using Petri Net Approach. *Journal of Manufacturing Technology Management*, 15(7), p. 700-714.
- Shahabudeen, P. and G.D. Sivakumar, 2007. Algorithm for the Design of Single-stage Adaptive Kanban System. *Computers & Industrial Engineerin*, 54, p. 800-820.
- Silva, M. and L. Recalde, 2003. On Fluidification of Petri Nets: from Discrete to Hybrid and Continuous models. *IFAC-ADHS'03*, Saint Malo, France, p. 9-20.
- Sugimori, Y., K. Kusunoki, F. Cho and S. Uchikawa, 1977. Toyota Production System Materialization of Just-in-time and Respect-for-human System. *International Journal of Production Research*, 15(6), p. 553-564.
- Tardif, V. and L. Maaseidvaag, 2001. An Adaptive Approach to Controlling Kanban Systems. *European Journal of Operation Research*, 132, p. 411-424.
- Thomas, A., H. El Haouzi, T. Klein, S. Belmokhtar and C. Herrera, 2009. Architecture de Systèmes Contrôlés par le Produit pour un Environnement de Juste à Temps. *Journal Européen des Systèmes Automatisés*, 43, p. 513-535.
- Ullah, H. and E.L.J. Bohez, 2008a. A Petri Net Model for Sequence Optimization and Performance Analysis of Flexible Assembly Systems. *Journal of Manufacturing Technology Management*, 19(8), p. 985-1003.
- Ullah, H. and E.L.J. Bohez, 2008b. A Petri Net Model for the Design and Performance Evaluation of a Flexible Assembly System. *Assembly Automation*, 28(4), p. 325-339.
- Zaytoon, J., (éditeur), 1998. Hybrid dynamical Systems. *Journal Européen des Systèmes Automatisés*, 32, pp. 9-10.