



HAL
open science

Petri net control of a kanban loop

Philippe Thomas, André Thomas, Sana Belmokhtar

► **To cite this version:**

Philippe Thomas, André Thomas, Sana Belmokhtar. Petri net control of a kanban loop. International Conference on Industrial Engineering and Systems Management, IESM 2011, May 2011, Metz, France. pp.CDROM. hal-00601725

HAL Id: hal-00601725

<https://hal.science/hal-00601725>

Submitted on 20 Jun 2011

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.

Petri net control of a kanban loop *

Philippe THOMAS ^a, André THOMAS ^a, Sana BELMOKHTAR ^a

^a *Centre de Recherche en Automatique de Nancy (CRAN-UMR 7039), Nancy-Université, CNRS
ENSTIB 27 rue Philippe Séguin, B.P. 1041, 88051 Epinal cedex 9 France.*

Abstract

Just in Time (JIT) is a management philosophy that favors improvement through inventory reduction by producing only the right quantity at the right time. A kanban system is an important element of JIT philosophy. However, whereas kanban system is suitable when demand is stable, it should be adjusted when demand is a significantly variable to be efficient. In addition, some works have proved that Petri nets are useful to model a kanban loop. Then, the objective of the present work is to exploit the analytical relation of Petri net models in order to improve the control of kanban system by using the real data provided by the system or by the product. This approach is based on the use of a continuous approximation of a discrete Petri net model in order to exploit a strategy of command based on gradient descent. In the present paper, two continuous approximations are proposed and compared.

Key words: Kanban, Petri net, gradient, just in time

1 Introduction

The Just in Time (JIT) philosophy has appeared during the seventies in Japan [9], [18], [27]. In a JIT system, a good is produced with the right quantity at the right time [2]. In order to drive a JIT system, a kanban system is needed. Kanban means card in Japanese. In a kanban system, cards are used systematically to control the production within stations and the movement of parts between stations. A kanban card is attached to each product (or to each batch). Then, consumption of one product (or batch) frees the kanban card which can be attached to a new product (or batch) to product. This release of kanban card leads to a new manufacturing order. The kanban system is a pull control system, which can be represented by using stage notion [9]. Pull systems are most successful in production environments with stable demand and lead time [28]. In this case, a kanban system uses a constant number of cards which can be allocated to a product (or batch) or free. In an instable environment (high variation of demand), such a system is inefficient in such context leading to modifying and adjusting its parameters to avoid starving or high stock level. Usually, two approaches can be used. First, the number of kanban cards present in the system may be adjusted. Second, the number of products associated to a kanban card may be modified. This is an adaptive kanban system [25], [28].

Petri nets (PN) are useful for the discrete event systems (DES) and hybrid systems (HS) modelling [6], [17], [30] because they combine, in a comprehensive way, intuitive graphical representations and powerful analytic

* This paper was not presented at any other revue. Corresponding author P. Thomas. Tel. +33(0) 3 29 29 61 73. Fax +33(0) 3 83 68 44 37

Email addresses: philippe.thomas@cran.uhp-nancy.fr (Philippe Thomas), andre.thomas@cran.uhp-nancy.fr (André Thomas), sana.belmokhtar@cran.uhp-nancy.fr (Sana Belmokhtar).

expressions [4], [5], [20]. A lot of results have been established for the control of the DES or HS. The modelling of kanban system is one of the applications of the PN theory. In fact, numerous models of kanban systems by using PN [13], timed Petri nets [3], [9], [11], [22], [24], stochastic Petri nets [21] or coloured Petri nets [8], [18], [19], [29] have been proposed. When the dynamics of a physical system are numerous, the corresponding discrete PN (PN, TPN, CPN) model has too many marks which lead to a high complexity. Thus, David and Alla [7] have proposed to use continuous Petri nets with variable speed (VCPN). Moreover, they have proposed some methods of transformation of TPN into VCPN. A lot of results have been established with VCPN for the control of system [1], [10], [12], [16], [26]. This approach appears as an interesting alternative to the discrete control of DES [14].

The main contribution of this work is to propose and compare two VCPN approximation models of a TPN model of a kanban system. The use of VCPN approximations allows using control methods of VCPN. A gradient based controller for continuous Petri nets [16] is used in order to control the flows of products in the system. Now, the Petri nets theory is resumed by focusing on TPN and VCPN and the definition of notations is performed. In a third part, the TPN model of a kanban system used as reference is recalled and two VCPN approximations of this model are proposed. In the next part, we present the gradient based controller strategy used in the part five where the results are presented before to conclude.

2 Petri nets

A Petri net (PN) with n places and p transitions is defined as $\langle P, T, \text{Pre}, \text{Post}, M_0 \rangle$ where $P = \{P_i\}_{i=1, \dots, n}$ is a not empty finite set of places, $T = \{T_j\}_{j=1, \dots, p}$ is a not empty finite set of transitions, such that $P \cap T = \emptyset$ [4], [20]. IN is the set of integer numbers. $\text{Pre}: P \times T \rightarrow IN$ is the pre-incidence application: $\text{Pre}(P_i, T_j)$ is the weight of the arc from place P_i to transition T_j and $W_{PR} = (w_{ij}^{PR})_{i=1, \dots, n, j=1, \dots, p} \in IN^{n \times p}$ with $w_{ij}^{PR} = \text{Pre}(P_i, T_j)$ is the pre-incidence matrix. $\text{Post}: P \times T \rightarrow IN$ is the post-incidence application: $\text{Post}(P_i, T_j)$ is the weight of the arc from transition T_j to place P_i and $W_{PO} = (w_{ij}^{PO})_{i=1, \dots, n, j=1, \dots, p} \in IN^{n \times p}$ with $w_{ij}^{PO} = \text{Post}(P_i, T_j)$ is the post-incidence matrix. The PN incidence matrix W is defined as $W = W_{PO} - W_{PR} \in IN^{n \times p}$. Let us also define $M = (m_i)_{i=1, \dots, n} \in IN^n$ as the marking vector and $M_0 \in IN^n$ as the initial marking vector. \mathcal{T}_j (resp T_j^o) stands for the pre-set (resp. post-set) places of T_j . A firing sequence is an ordered set of transitions that are successively fired from marking M to marking M' . Such a sequence is represented by its characteristic vector $X = (x_j)_{j=1, \dots, p} \in IN^p$ where x_j stands for the number of T_j firings. The marking M' is related to the marking M and to the firing sequence X according to:

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

In autonomous PN without conflict, the enabling degree of each transition T_j , related to the marking M , is given by:

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

where $\text{fix}(\cdot)$ stands for the integer part of (\cdot) .

2.1 Timed Petri nets TPN

Two types of timed Petri nets occur, T-timed and P-timed Petri nets [7]. Here, only the T-timed Petri nets (TPN) are considered. A TPN is defined as $\langle \text{PN}, \text{Tempo} \rangle$ such as PN is an ordinary Petri net as defined above and Tempo is an application from the set T of transitions into IR^+ . Tempo (T_j) = d_j is the temporization associated to the transition T_j .

2.2 Continuous Petri nets VCPN

Continuous PN are a particular class of timed PN, deduced from TPN [23] to provide a continuous approximation of the system behaviour [7]. A continuous PN with n places and p transitions is defined as $\langle \text{PN}, X_{max} \rangle$ where PN is a Petri net and $X_{max} = (x_{maxj})_{j=1, \dots, p} \in IR^{+p}$ is the vector of maximal firing frequencies with IR^+ the set of non-negative real numbers. The marking $m_i(t)$ of each place P_i , $i = 1, \dots, n$, at time t has a non-

negative real value and each transition firing is a continuous flow in continuous PN. In fact, the transition T_j is fired with a frequency $x_j(t)$ less than the maximal frequency x_{max_j} . Let us define $X(t) = (x_j(t))_{j=1,\dots,p} \in \mathbb{R}^{+p}$ as the firing frequencies vector at time t . The marking evolution is given by the differential system (3):

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

Among the existing models of continuous PN, continuous PN with variable speeds (VCPN) are proved to give good approximations of TPN [7]. Components of the marking vector are continuous functions of the time; components of the firing frequencies vector $X(t)$ depend continuously on the marking of the places according to:

$$x_j(t) = x_{max_j} \cdot \mu_j(t) \quad \text{where} \quad \mu_j(t) = \min_{P_i \in {}^*T_j} (m_i(t)) \quad (4)$$

A self loop is usually associated to each transition T_j in order to limit the number of simultaneous firings.

2.3 Piecewise linear state space representation for VCPN

Due to the commutation function « min » and to the products between marking vector and maximal firing frequencies vector in (4), VCPN models are not linear but piecewise linear systems [1], [15], [16]. In order to bring VCPN models in the usual state space representation, let us introduce $U(t) \in \mathbb{R}^{+d}$ as the VCPN input vector at time t and $Y(t) \in \mathbb{R}^{+q}$ as the VCPN output vector at same instant. The input vector $U(t)$ is defined as the maximal firing frequencies of some transitions. For this purpose, the set of transitions T is divided into 2 disjoint subsets T_C , and T_{NC} such that $T = T_C \cup T_{NC}$. T_C is the subset of the controllable transitions, and T_{NC} is the uncontrollable transitions subset. As a consequence, let us define $X_C(t) = (x_j(t))_{T_j \in T_C} \in \mathbb{R}^{+d}$ and $X_{NC}(t) = (x_j(t))_{T_j \in T_{NC}} \in \mathbb{R}^{+p-d}$ according to:

$$D^{-1}.X(t) = \begin{pmatrix} X_C(t) \\ X_{NC}(t) \end{pmatrix} \quad (5)$$

with $D \in \mathbb{R}^{p \times p}$ a suitable permutation matrix (i.e. D is the matrix of a bijective mapping from T to T that clusters the set of transitions into controllable and uncontrollable ones). The controllable inputs vector $U(t) = X_{max_C}(t) \in \mathbb{R}^{+d}$ corresponds to the maximal firing frequencies of the transitions to be controlled. The input vector is constrained $0 \leq U(t) \leq U_{max}$ in order to limit the firing frequencies in a non negative bounded interval. The uncontrollable maximal firing frequencies $X_{max_{NC}}$ are supposed to be constant according to the VCPN models. The output vector $Y(t) = Q.M(t) \in \mathbb{R}^{+e}$ is composed of a selection of subnets marking that are observable. For this purpose, let us define $Q = (q_{ki})_{k=1,\dots,e, i=1,\dots,n} \in \mathbb{R}^{e \times n}$ as a positive observation matrix (i.e. Q is the matrix of a constant projector, each row corresponds to a positive weighted sum of the PN places marking). The goal of the controller is to drive $Y(t)$ according to some reference trajectories in the output space. Equation (3) can be rewritten as:

$$\frac{dM(t)}{dt} = W_{NC}.X_{NC}(t) + W_C.X_C(t) \quad Y(t) = Q.M(t) \quad (6)$$

with $W_C = (w_{Cij})_{i=1,\dots,n, j=1,\dots,d} \in \mathbb{R}^{n \times d}$ and $W_{NC} = (w_{NCij})_{i=1,\dots,n, j=1,\dots,p-d} \in \mathbb{R}^{n \times (p-d)}$ such that $(W_C / W_{NC}) = W.D$.

Several phases occur in the VCPN behaviour. Each phase φ is active between two successive commutations of the “min” operators in (4) and corresponds to a particular configuration of these operators characterised by the p clustering functions f_j :

$$f_j : \mathbb{R}^{+n} \rightarrow \{1, \dots, n\} \\ M(t) \rightarrow f_j(M(t)) \quad \text{such as} \quad m_{f_j}(t) = \mu_j(t) \quad (7)$$

Each function f_j specifies the place in the preset of T_j which has the minimal marking. During each phase φ , a constant relationship between the components of vectors $X_C(t)$ and $M(t)$ and also between $X_{NC}(t)$ and $M(t)$ occurs. This relation can be expressed under scalar form by using the functions f_j or under vectorial form by using the set

of vectors $A_j(\varphi) \in \{0,1\}^{1 \times n}$ and $B_j(\varphi) \in \{0,1\}^{1 \times n}$ which are constant during each phase but which may varied from one phase to another:

$$x_{C_j}(t) = u_j(t) \cdot A_j(\varphi) \cdot M(t), \quad j = 1, \dots, d \quad x_{NC_j}(t) = x_{\max NC_j} \cdot B_j(\varphi) \cdot M(t), \quad j = 1, \dots, p-d \quad (8)$$

Equation (6) can be rewritten under vectorial form:

$$\frac{dM(t)}{dt} = \left(\sum_{j=1}^d u_j(t) \cdot W_{C_j} \cdot A_j(\varphi) + \sum_{j=1}^{p-d} x_{\max NC_j} \cdot W_{NC_j} \cdot B_j(\varphi) \right) \cdot M(t) \quad Y(t) = Q \cdot M(t), \quad \varphi = 1, 2, \dots \quad (9)$$

where W_{C_j} denotes the j^{th} column of matrix W_C and W_{NC_j} denotes the j^{th} column of matrix W_{NC} . Equations (9) is piecewise bilinear representation of the VCPN (2). Each phase is characterised by a set of matrices $W_{C_j} \cdot A_j(\varphi) \in IN^{n \times n}$ associated to the controllable transitions and $W_{NC_j} \cdot B_j(\varphi) \in IN^{n \times n}$ associated to the uncontrollable ones.

3 Petri net models of kanban systems

As shown in the introduction, a lot of models of kanban systems based on Petri nets have been proposed. In the proposed work, a TPN model initially presented in [9] has been particularly studied because of its simplicity and its modularity. It can be noticed that the proposed approach can be applied to any model based on TPN and even to those based on CPN since an equivalent TPN could always be built. The considered kanban system has the following characteristics: each stage consists only of a single facility comprising one or more identical machines; each stage has, at most, one preceding stage and one succeeding stage; the manufacturing system produces only one type of part; the size of all kanbans is the same and is taken to be equal to one; when a kanban is removed from a part consumed by the downstream stage, it is immediately available to be attached to a part of the upstream stage.

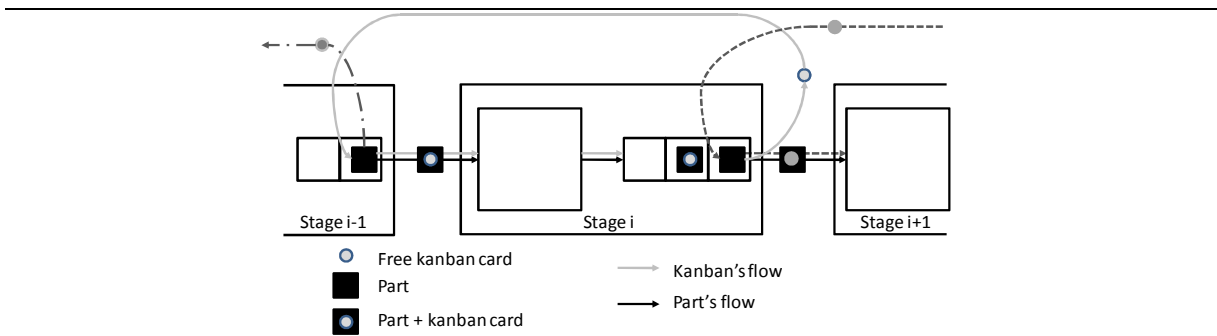


Fig. 1. Links between different stages

In a kanban system, the production system is decomposed in stages. Figure 1 presents the kanban system mechanism at the stage i and shows the flow (of parts and of kanban cards) between the stages $i-1$, i and $i+1$. This system is modeled by [9] by using the TPN model shown figure 2. When the stage i is considered (figure 1), each kanban card can be in three different states: free and waiting to be attached to a new part of stage $i-1$, or attached to a part that is in manufacturing process, attached to a finished part that is in the output buffer. So, these three states are represented by three places denoted respectively F_i , I_i and O_i . The number of tokens in place I_i represents the number of parts in the manufacturing process. The number of tokens in place O_i represents the number of parts in the output buffer when the number of tokens in F_i represents the number of free kanban cards. The number of tokens in the place N_i represents the number of identical machines of stage i . The transition T_i between the places I_i and O_i is a timed transition. Its delay d_i corresponds to the processing time. The transition τ_i represents the synchronization between stages i and $i+1$. This transition is an immediate transition. The TPN shown figure 2 corresponds to the case where the demand and the supply are supposed infinite. Di Mascolo *et al.* [9] have proposed some variations of this initial model in order to take into account, not infinite supply, or lost demands as examples. This paper focuses only on this first model.

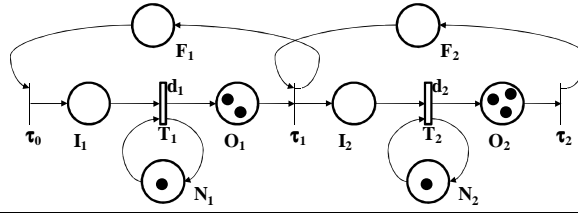


Fig. 2. TPN model for a two-stage kanban system

The goal of this work is to drive the considered kanban system by using its PN model and a control algorithm of the VCPN. So, a VCPN approximation of the TPN model shown figure 2 must be constructed. The transformation of a TPN into a VCPN is always feasible, but some modification must be applied to the model [7]. In order to do that, these different transformations must be applied: the discrete marking of places must be replaced by continuous marking; the delays associated to transitions of the TPN are used in order to determine the maximal speeds of firing of VCPN transitions. The maximal speeds of firing of the transitions are then obtained by using the relation:

$$x_{\max_i} = \frac{1}{d_i} \tag{10}$$

To do that, the main difficulty is the immediate transitions τ_0 , τ_1 and τ_2 of the TPN (figure 2). In fact, these transitions have a null delay which leads to a maximal firing speed of transition infinite. To solve this problem, two approaches can be used. The first one is to consider these immediate transitions as timed transitions but with a short delay comparing with the other dynamics of the system. This approach leads to the VCPN model shown figure 3. The resulting model is called here the complete VCPN model.

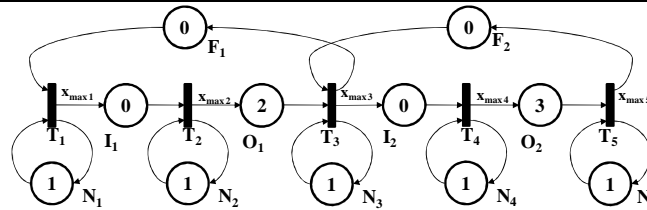


Fig. 3. Complete VCPN model for a two-stage kanban system

The transitions T_1 , T_3 and T_5 corresponds to the immediate transitions of the TPN (figure 2), and so, have a great maximal firing speed. The firing speeds T_2 and T_4 correspond to the processing speeds of the two stages. The marking of places N_2 and N_4 correspond to the number of machines constituting the two stages. The second approach is to consider that, because τ_0 , τ_1 and τ_2 are immediate transitions of the TPN, each token in place O_i implies that transition τ_i is immediately fired and the considered token is driven in the place I_{i+1} . So it is possible to aggregate these two places O_i and I_{i+1} in order to preserve uniquely the timed transitions. This approach leads to the VCPN model shown figure 4 which is called reduced VCPN model. These two approaches will be compared in the part 5.

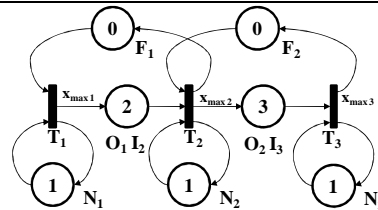


Fig. 4. Reduced VCPN model for a two-stage kanban system

It can be noticed that, in this reduced model, the processing of a part needs that some tokens are present in place F corresponding to free kanban cards. This fact is the main limitation of this model. However, because the main objective is to determine how many kanban cards must be introduced in the real system, the number of kanban cards must be oversized in order to allow the introduction of new cards. So, this limitation is not problematic.

4 Control design for VCPN

Flow control for VCPN was investigated by several authors [1], [10], [12], [14], [15], [16], [26]. The control algorithm used here is those proposed by Lefebvre *et al.* [15] which is inspired from the learning algorithm of the neural networks. This approach takes advantage of the gradient propagation through the PN nodes in order to minimise the square of instantaneous error between desired and measured outputs by modifying the maximal firing frequencies of controllable transitions. Gradient algorithms perform the minimisation of a scalar cost function that evaluates the distance between the desired output $Y_d(t)$ and the system output $Y(t)$. Let us assume that measures of the desired output are obtained with a sampling period Δt during the time horizon H . As a consequence, the proposed controllers will be worked out in discrete time. This algorithm is now summary.

4.1 Gradient based controller

For seek of simplicity, let us first consider the single output case. The instantaneous error is defined as $\varepsilon(k) = y_d(k) - y(k)$, where $y_d(k)$ stands for the desired output at time $t = k.\Delta t$, and $y(k)$ stands for the marking of the VCPN output y at time $t = k.\Delta t$ obtained from the marking $M(k-1)$ and the input vector $U(k-1)$ according to a first order numerical method:

$$y(k) = y(k-1) + \Delta t \cdot \left(\sum_{j=1}^d u_j(k-1) \cdot Q \cdot W_{C_j} \cdot A_j(\varphi) + \sum_{j=1}^{p-d} x_{\max NC_j} \cdot Q \cdot W_{NC_j} \cdot B_j(\varphi) \right) \cdot M(k-1) \quad k > 0 \quad (11)$$

Let us consider the scalar cost function $v(k)$:

$$v(k) = \frac{1}{2} \varepsilon^2(k) \in \mathbb{R} \quad (12)$$

Gradient-based methods result from the Taylor series expansion of the cost function $v(k)$ in order to work out the optimal value of the input vector $U(k)$. Second order terms are usually neglected in evaluation of hessian matrix, but an adaptive term $\lambda.I$ is added in order to approximate the inverse of the hessian matrix when it is not regular or badly conditioned. Thus, the actualisation of the control input is given by:

$$\delta U(k) = -(S(k) \cdot S(k)^T + \lambda.I)^{-1} \cdot S(k) \cdot \varepsilon(k), \quad k \geq 0 \quad (13)$$

with $\delta U(k) = U(k) - U(k-1)$. $S(k) \in \mathbb{R}^d$ is the gradient of the output y with respect to the input vector U , calculated at time $t = k.\Delta t$. When λ is great, the algorithm works as a gradient method. On the contrary, when λ is small, the algorithm works as a hessian method which is faster but may be instable. Different methods can be used to tuned λ [17]. Here, λ is tuned to 1 which corresponds to hessian comportment.

The previous controller can be generalised in the multi-outputs case, by considering the scalar cost function:

$$v(k) = \frac{1}{2} \sum_{\alpha=1}^e \varepsilon_{\alpha}^2(k) \in \mathbb{R} \quad (14)$$

that results in the following updating rule for the controller:

$$\delta U(k) = - \left(\sum_{\alpha=1}^e (S_{\alpha}(k) \cdot S_{\alpha}(k)^T + \lambda.I) \right)^{-1} \cdot \left(\sum_{\alpha=1}^e (S_{\alpha}(k) \cdot \varepsilon_{\alpha}(k)) \right) \quad (15)$$

where $S_{\alpha}(k) \in \mathbb{R}^d$ is the gradient of the output y_{α} considering the variations of the input vector U , calculated at time $t = k.\Delta t$. In order to converge quickly to the minimum of the cost function, the learning algorithms based on

the steepest descent of the gradient may be iterated several times on each sample [16]. In order to use this algorithm the value of the gradient $S_\alpha(k)$ used in (15) must be determined.

4.2 Gradient of the output

Gradient algorithms are based on the evaluation of sensitivity functions. Such functions are defined in continuous time for VCPN and will be worked out in discrete time according to the sampling period Δt in order to be implemented in numerical controllers. Considering $S_{\alpha\gamma}(k)$ the gradient of the output y_α with respect to input u_γ (the maximal firing frequency $x_{\max c_\gamma}$). The gradient $S_{\alpha\gamma}(k)$ is determined by using a first order method and the classification functions f_j given by (7):

$$\delta s_{\alpha\gamma}(k) = \sum_{i=1}^n q_{\alpha i} \cdot \left(w_{i\gamma} \cdot \mu_\gamma(k) + \sum_{\substack{j=1 \\ j \neq \gamma}}^p w_{ij} \cdot x_{\max c_j} \cdot s_{f_j\gamma}(k) \right) \cdot \Delta t \quad (16)$$

where $q_{\alpha i}$ is the cell of the i^{th} line and α^{th} column of the matrix Q. $\delta s_{\alpha\gamma}(k)$ represents the variation of the gradient $\delta s_{\alpha\gamma}(k) = s_{\alpha\gamma}(k) - s_{\alpha\gamma}(k-1)$ between times $k-1$ and k :

$$\begin{cases} \delta s_{\alpha\gamma}(k) = s_{\alpha\gamma}(k) - s_{\alpha\gamma}(k-1) \\ s_{\alpha\gamma}(0) = 0 \end{cases} \quad (17)$$

5 Example

Now, the control of the two-stage kanban system modelled thanks a TPN described figure 2 is investigated. As seen in part 3, two continuous approximations of this TPN has been built thanks VCPN theory. These two models called complete VCPN model and reduced VCPN model are described by figures 3 and 4. The incidence matrixes of these two VCPN are given by:

$$W_{\text{complete}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}^T \quad W_{\text{reduced}} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}^T \quad (18)$$

For the two VCPN models, Transition T_1 represents the arrival of parts in the system. The marking of the places N_i limit the number of simultaneous firing of the transitions. The output transitions (T_5 for the complete model and T_3 for the reduced one) represent the consumption of finished parts. In order to simulate this demand, the maximal firing frequency of these transitions (x_{\max_5} for the complete model and x_{\max_3} for the reduced one) is chosen equal to 2. The transitions T_2 and T_4 for the complete model and T_1 and T_2 for the reduced model allow to control the system. The command $U(t)$ corresponds to the maximal firing frequencies of these transitions and it is constraint between 0 and 5. The transitions T_1 and T_3 of the complete VCPN model which correspond to the immediate transitions in TPN have a great maximal firing speeds comparing to the other dynamics of the system. Their maximal firing speeds are fixed to 10.

Now the two VCPN models are completely defined. In order to control them, the objectives must be defined. The main objective of a kanban system is to supply the demand. In order to do that, the output buffer of the second stage should never be empty. So, for our VCPN models, this objective results in a constraint on the marking of the place which represent this buffer (O_2 for the complete model and O_2I_3 for the reduced one). So the measured output y of our system is the marking of the place O_2 for the complete model and O_2I_3 for the reduced one. This measured output must converge to the desired value $y_d = 1.2$. In a first time, the initial marking M_0 is given by:

$$M_{0_{\text{complete}}} = (1, 1, 1, 1, 1, 0, 2, 0, 3, 0, 0)^T \quad M_{0_{\text{reduced}}} = (1, 1, 1, 2, 3, 0, 0)^T \quad (19)$$

This marking corresponds to a system where all the kanban cards are associated to finished parts waiting to be consumed by the downstream system. The marking of the places F_1 and F_2 which correspond to the free kanban cards are both empty.

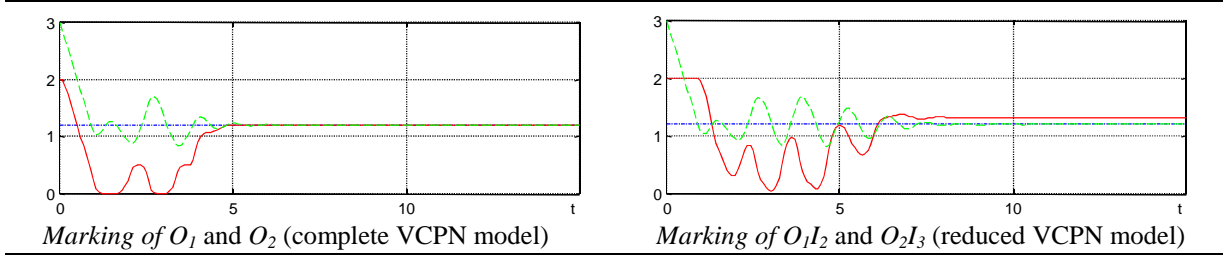


Fig. 5. Measured marking of places O_1 and O_{1I_2} (dashed line) and places O_2 and O_{2I_3}

The figures 5 show the marking of places corresponding to the output buffer during the time. This figure shows that the goal of 1.2 tokens in places O_2 (complete) and O_{2I_3} (reduced) is reached with the two VCPN models. The complete VCPN model is quicker than reduced model. The figures 6 show the evolutions of the commands of the system. These figures show that the command of complete VCPN stabilizes itself more quickly than reduced VCPN model.

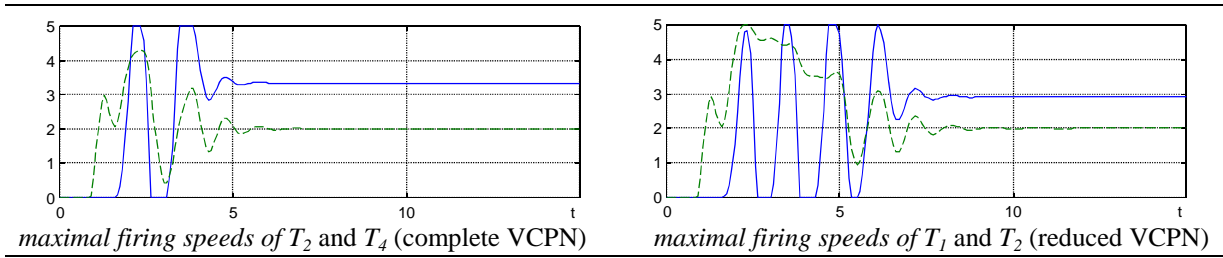


Fig. 6. Maximal firing speed of commands $U(t)$ (stage 1: continuous; stage 2: dashed)

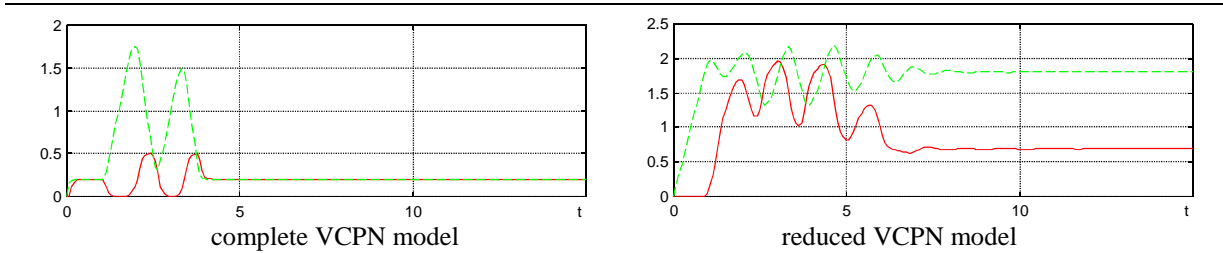


Fig. 7. Measured marking of places F_1 and F_2 (dashed line)

The figures 7 show the marking of places F_1 and F_2 which represents the number of free kanban cards in the two stages. These figures show a great difference of behavior between the two models. In the Complete model, only 0.2 cards are free in the two stages when 0.68 and 1.8 cards are free in the reduced model after stabilization. This fact implies that the complete model stocks parts in the input buffer of the two stages I_1 and I_2 which explain the speed of convergence of this model. However, the complete model is less parsimonious than the reduced one. Now, a second set of initial conditions described by a second vector of initial marking is used:

$$M_{01_{complete}} = (1, 1, 1, 1, 1, 0, 0, 0, 0, 2, 3)^T \quad M_{01_{reduced}} = (1, 1, 1, 0, 0, 2, 3)^T \quad (20)$$

This marking correspond to a system where all the kanban cards are free. The figures 8 show the marking of places corresponding to the output buffer. This figure shows that the goal of 1.2 tokens in places O_2 (complete) and O_{2I_3} (reduced) is reached with the two VCPN models and even quicker than in the preceding case.

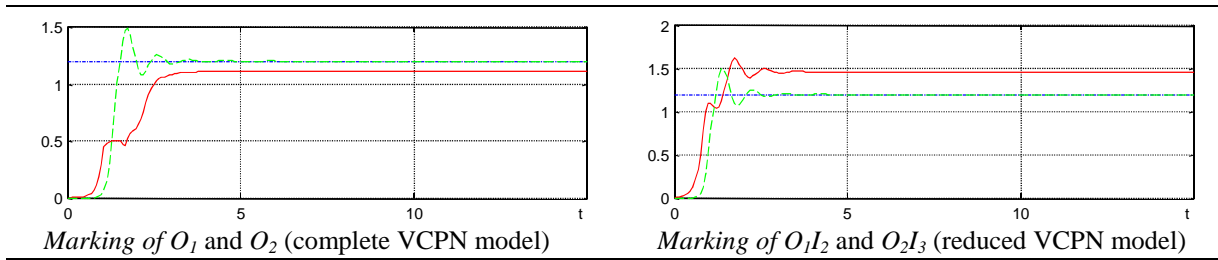


Fig. 8. Measured marking of places O_1 and O_{1I_2} (dashed line) and places O_2 and O_{2I_3} (marking M_{01})

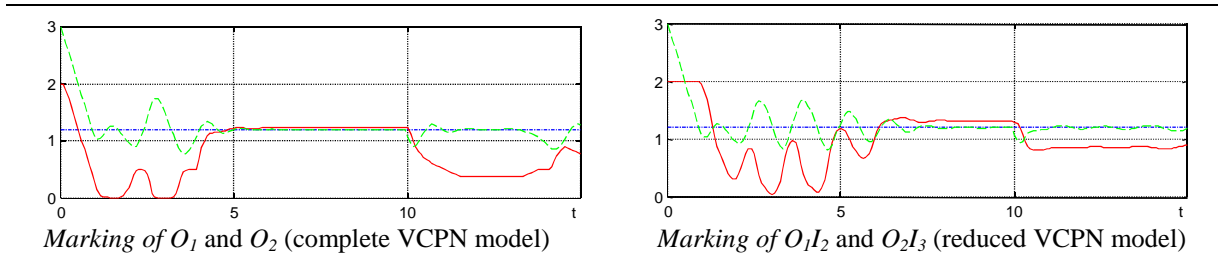


Fig. 9. Measured marking of places O_1 and O_{1I_2} (dashed line) and places O_2 and O_{2I_3} (evolution of demand)

Now, the initial marking is again the marking M_0 defined by (19). However, the demand is doubled at time 10. Figures 9 show the marking of places corresponding to the output buffer during the time. The variation of demand (demand is doubled at time 10) leads to a small variation of the marking of places O_2 (complete model) and O_{2I_3} (reduced model). However, the behavior of both models is different. In fact, for the complete model, the impact of variation of demand is mainly absorbed by consuming the stock of stage 1 (marking of place O_1). This leads to another variation of the marking of place O_2 at time 13. For the reduced model, the variation of demand is mainly absorbed by increasing the maximal firing speeds of the two transitions T_1 and T_2 .

6 Epilogue

A new approach of control of kanban system has been proposed based on Petri nets. A gradient based controller of VCPN model has been used and two VCPN models of kanban system have been compared on different experiences. From these different experiences, the main characteristics of the two models can be highlighted and resumed in table 1. The main advantages of complete model are speed and stability when the main advantages of reduced model are parsimony and adaptability.

Table 1
Characteristics of the two models.

	Speed	Stability	Parsimony	Adaptability
Complete	++	+	--	+
Reduced	+	-	++	+

This work shows that this approach can be used in order to adapt automatically the number of kanban cards in function of the demand which can be related to the adaptive kanban approach. The control of the Petri net model determines the needed number of kanban cards in function of the demand. So, it can be used to perform a fast simulation of the kanban system by considering an evolution of the demand to determine whether some kanban cards must be introduced or removed from the kanban system. The perspectives of this work are to use this approach in order to design a real kanban system. Moreover, the use of a continuous approximation of a TPN model introduces some mistakes. Thus an evaluation of the error performed during this approximation phase must be done. Furthermore, this method must be compared with concurrent methods of kanban system dimensioning.

7 References

- [1] Amrah, A., N. Zerhouni and A. El Moudni. Constrained State Feedback Control of a Class of Discrete Event Systems Modelled by Continuous Petri Nets. ICARV'96, Singapore, 979-984, 1996.
- [2] Beranger, P. Les Nouvelles Règles de la Production, Dunod Ed., Paris, 1987.
- [3] Bohez, E.L.J. A new Generic Timed Petri Net Model for Design and Performance Analysis of a Dual Kanban FMS. International Journal of Production Research, 42(4):719-740, 2004.
- [4] Brams, G.W. Réseaux de Petri, Vol I et II, Masson, Paris, 1983.
- [5] Brauer, W., W. Reisig and G. Rosenberg. Petri Nets: Central Models and their Properties. Lecture Notes in Computer Science, 254, 1986.
- [6] Cassandras, C.G. Discret Event Systems: Modeling and Performance Analysis, Asken Ass. Inc. Pub, 1993.
- [7] David, R. and H. Alla. Petri Nets and Grafcet – Tools for Modelling Discrete Events Systems, Prentice Hall, London, 1992.
- [8] Desrochers, A. and M. Pia Fanti. A Supply Chain Model Using Complex-Valued Token Petri Nets. Proc. of the 2005 IEEE Int. Conf. on Robotics and Automation, Barcelona, Spain, 3709-3714, 2005.
- [9] Di Mascolo, M., Y. Frein, Y. Dallery and R. David. A Unified Modeling of Kanban Systems Using Petri Nets. The International Journal of Flexible Manufacturing Systems, 3:275-307, 1991.
- [10] Ghabri, M. Sur la Modélisation et la Commande des Systèmes Flexibles de Production. Thèse de Doctorat, INPG, Grenoble, France, 1995.
- [11] Giua, A., A. Piccaluga and C. Seatzu. Firing Rate Optimization of Cyclic Event Graphs by Token Allocations. Automatica, 38:91-103, 2002.
- [12] Hanzalek, Z. Continuous Petri Nets and Polytopes. IEEE-SMC'03, Washington, USA, 1513-1520, 2003.
- [13] Jain, A., P.K. Jain and I.P.Singh. Performance Modelling of FMS with Flexible Process Plans – A Petri Net Approach. International Journal of Simulation Modelling, 5(3):101-113, 2006.
- [14] Lefebvre, D. and P. Thomas. Identification par phase des fréquences maximales de franchissement dans les réseaux de Pétri continus à vitesse variable. CIMASI'02, Casablanca, Maroc, 2002.
- [15] Lefebvre, D., E. Leclercq, F. Druaux and P. Thomas. Source and sink transitions controllers for continuous Petri nets: a gradient - based approach. ADHS'03, Saint Malo, France, 2003.
- [16] Lefebvre, D., E. Leclercq, F. Druaux and P. Thomas. Commande des flux dans les réseaux de Petri continus par propagation du gradient. CIFA'04, Douz, Tunisie, 2004.
- [17] Lefebvre, D. and P. Thomas. Parameters estimation for timed and continuous Petri nets: application to the identification and monitoring of hybrid systems. Cybernetics and Systems, 36(3):217-250, 2005.
- [18] Mohanty, R.P., S. Kumar and M.K. Tiwari. Expert Enhanced Coloured Fuzzy Petri Net Models of Traditional, Flexible and Reconfigurable Kanban System. Prod. Planning and Control, 14(5):459-477, 2003.
- [19] Moore, K.E. and S. M. Gupta. Stochastic Coloured Petri Net (SCPN) Models of Traditional and Flexible Kanban Systems. International Journal of Production Research, 37(9):2135-2158, 1999.
- [20] Murata, T. Petri Nets: Properties, Analysis and Application. Proceedings IEEE, 77(4):541-580, 1989.
- [21] Nakashima, K. and S.M. Gupta. Performance Evaluation of a Supplier Management System with Stochastic Variability. International Journal Manufacturing Technology and Management, 5(1/2):28-37, 2003.
- [22] Panayiotou, C.G. and C.G. Cassandras. Optimization of Kanban-based Manufacturing Systems. Automatica, 35:1521-1533, 1999.
- [23] Ramchandani, C. Analysis of Asynchronous Concurrent Systems by Timed Petri nets. PhD Thesis, MIT, USA, 1973.
- [24] Seeluangawat, R. and E.L.J. Bohez. Integration of JIT Flexible Manufacturing, Assembly and Disassembly System using Petri Net Approach. Journal of Manufacturing Technology Management, 15(7):700-714, 2004.
- [25] Shahabudeen, P. and G.D. Sivakumar. Algorithm for the Design of Single-stage Adaptive Kanban System. Computers & Industrial Engineering, 54:800-820, 2007.
- [26] Silva, M. and L. Recalde. On Fluidification of Petri Nets: from Discrete to Hybrid and Continuous models. IFAC-ADHS'03, Saint Malo, France, 9-20, 2003.
- [27] Sugimori, Y., K. Kusunoki, F. Cho and S. Uchikawa. Toyota Production System Materialization of Just-in-time and Respect-for-human System. International Journal of Production Research, 15(6):553-564, 1977.
- [28] Tardif, V. and L. Maaseidvaag. An Adaptive Approach to Controlling Kanban Systems. European Journal of Operation Research, 132:411-424, 2001.
- [29] Ullah, H. and E.L.J. Bohez. A Petri Net Model for Sequence Optimization and Performance Analysis of Flexible Assembly Systems. Journal of Manufacturing Technology Management, 19(8):985-1003, 2008.
- [30] Zaytoon, J., (ed.). Hybrid dynamical Systems. Journal Européen des Systèmes Automatisés, 32 :9-10, 1998.