Performance Control of Markovian Petri Nets via Fluid Models: A Stock-Level Control Example

C. Renato Vázquez Manuel Silva

Abstract— **Petri nets is a well-know formalism for studying discrete event systems. Applications include performance evaluation in communication networks, production systems, supply chains, and the implementation of sequence controllers. Timed Continuous Petri Net ($TCPN$) systems are continuous-state models that can approximate the dynamical behavior of *discrete* Markovian Petri nets (MPN). Based on this, an estimator-based control structure is introduced here for applying a control law designed for a $TCPN$ into the *original discrete system*. The result is a control policy for driving a MPN system in such a way that the *mean value* of its marking will reach a desired value, by applying additional delays to the controllable transitions. A stock level control of a Kanban-based automotive assembly line is synthesized as an application example.**

I. Introduction

Petri nets are a recognized paradigm useful for modeling and analyzing discrete event systems (DES). Applications include the analysis of communication protocols and manufacturing systems, the implementation of sequence controllers, validation in software development, and performance evaluation in multiprocessor systems, communication networks, production systems, supply chains, etcetera [1].

Several works can be found in the literature providing different control strategies for Petri nets. For instance, in [2] a state-feedback control that meets safety specifications in the form of mutual exclusions constraints is proposed (GMEC). A survey of control results in Petri nets can be found in [3]. Recalling from there, control policies can be classified into two different classes: the *state feedback control*, which has been mainly studied by means of a particular model called controlled Petri nets, and the *event feedback control* that has been mainly considered in a formal language setting and the corresponding models are called labeled Petri nets. Extensions to timed systems can also be found in the literature. Most control strategies are defined for the same control objective: disabling transitions for avoiding forbidden markings, in accordance with the Supervisory-Control Theory. A problem commonly found in the synthesis of controllers is the state explosion.

Fluidification constitutes a relaxation technique for studying discrete event systems through a continuous approximated model, thus avoiding the state explosion problem. In Petri Nets, fluidification has been introduced from different perspectives [4], [5]. In this work, *timed continuous* Petri net ($TCPN$) models under infinite server semantics are considered, since it has been found that such systems approximate interesting classes of DES. In fluid models more analytical techniques can be used for the analysis of some interesting properties, like controllability [6] and the synthesis of controllers, either for the optimal steady-state control problem [7] or dynamic control for reaching a desired marking in models in which all transitions are controllable [8], [9] or with some uncontrollable transitions [10], [11].

Coming back to the discrete event systems, in [12] it has been shown that a $TCPN$ system, with white noise added to the flow (leading to state perturbation), can approximate the mean value and covariance matrix of the marking of the corresponding Markovian Petri net (MPN, i.e., a Petri net in which transitions are timed with exponentially distributed random delays). For that, the probability that any transition is enabled is close to one. Based on such result, the goal in this work is to interpret an apply a control law designed for a $TCPN$ system into the corresponding MPN one. Such control interpretation will result in a control policy for driving a live MPN system in such a way that the mean value of its marking will reach a desired value, just applying additional delays to the controllable transitions. This constitutes an important difference with the control laws derived for DES in previous works, and represents the main contribution of this one. This result will be illustrated by means of a given application example: the stock level control of a Kanban-based automotive assembly line [13].

This paper is organized as follows: in Section 2 some basic concepts on $TCPN$ and MPN systems are provided, while in Section 3 controllability concepts and control laws for $TCPN$ are recalled from previous works. In Section 4, the interpretation of a control law designed for a $TCPN$ system into the corresponding MPN one is introduced. Finally, a given application example is studied in Section 5, and conclusions are present in Section 6.

II. Basic concepts of TCPN and MPN

We assume that the reader is familiar with Petri nets (PNs) (for notation we use the standard one, see for instance [14]). The structure $\mathcal{N} = \langle P, T, \mathbf{Pre}, \mathbf{Post}\rangle$ of *continuous Petri nets* (CPN) is the same as the structure of discrete PNs. That is, P is a finite set of places, T is a finite set of transitions with $P \cap T = \emptyset$, $\mathbf{Pre}$ and $\mathbf{Post}$ are $|P| \times |T|$ sized, natural valued, *pre- and post- incidence matrices*. We assume that $\mathcal{N}$

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no 224498.

This work was partially supported by projects CICYT and FEDER DPI2006-15390.

Authors are with Dep. de Informática e Ingeniería de Sistemas, Centro Politécnico Superior, Universidad de Zaragoza, E-50018 Zaragoza, Spain `{cvazquez,silva}@unizar.es`

This paper has been written in honor to Prof. L. Recalde, who passed away last December.

is connected and that every place has a successor, i.e., $|p^\bullet| \geq 1$. The usual PN system, $\langle \mathcal{N}, \mathbf{M}_0 \rangle$ with $\mathbf{M}_0 \in \mathbb{N}^{|P|}$, will be said to be *discrete* so as to distinguish it from a *continuous* PN system $\langle \mathcal{N}, \mathbf{m}_0 \rangle$, in which $\mathbf{m}_0 \in \mathbb{R}^{|P|}_{\geq 0}$. In the following, the marking of a CPN will be denoted in lower case $\mathbf{m}$, while the marking of the corresponding *discrete* one will be denoted in capital letter $\mathbf{M}$. The main difference between both formalisms is in the evolution rule, since in *continuous* PNs firing is not restricted to be done in integer amounts ([4], [5]). As a consequence the marking is not forced to be integer. More precisely, a transition t is *enabled* at $\mathbf{m}$ iff for every $p \in^\bullet t$, $\mathbf{m}(p) > 0$, and its *enabling degree* is $enab(t, \mathbf{m}) = \min_{p \in ^\bullet t}\{\mathbf{m}(p)/\mathbf{Pre}(p,t)\}$. The firing of t in a certain amount $\alpha \leq enab(t, \mathbf{m})$ leads to a new marking $\mathbf{m}' = \mathbf{m} + \alpha \cdot \mathbf{C}$, where $\mathbf{C} = \mathbf{Post} - \mathbf{Pre}$ is the token-flow matrix. As in discrete systems, right and left integer annullers of the token flow matrix are called *T-* and *P-flows*, respectively. When they are non-negative, they are called *T-* and *P-semiflows*. If there exists $\mathbf{y} > \mathbf{0}$ such that $\mathbf{y} \cdot \mathbf{C} = \mathbf{0}$, the net is said to be *conservative*, and if there exists $\mathbf{x} > \mathbf{0}$ such that $\mathbf{C} \cdot \mathbf{x} = \mathbf{0}$ the net is said to be *consistent*. Here, we assume that the initial marking marks all *P-semiflows*.

A *Markovian Stochastic Petri Net* system *(MPN)* is a *discrete* system in which the transitions fire at independent exponentially distributed random time delays (for a classical approach for the analysis of MPN [15]). Then, the firing time of each transition is characterized by its firing rate. In this way, a *MPN* is a tuple $\langle \mathcal{N}, \mathbf{M}_0, \boldsymbol{\lambda} \rangle$, where $\boldsymbol{\lambda} \in \mathbb{R}^{|T|}_{>0}$ represents the transition rates. Transitions (like stations in queueing networks) are the meeting points of clients and servers. In this paper, we will assume infinite-server semantics for all transitions. Then, the time to fire a transition t_i, at a given marking $\mathbf{M}$, is an exponentially distributed random variable with parameter $\lambda_i \cdot Enab(t_i, \mathbf{M})$, where the integer enabling degree is $Enab(t_i, \mathbf{M}) = \min_{p \in ^\bullet t_i}\{\lfloor \mathbf{M}(p)/\mathbf{Pre}(p, t_i) \rfloor\}$. $Enab(t_i, \mathbf{M})$ also represents the number of active servers of t_i at marking $\mathbf{M}$. We suppose that a unique steady-state behavior exists, and we restrict our study to bounded in average and reversible (therefore ergodic) MPN systems.

A *Timed Continuous Petri Net* $(TCPN)$ is a *continuous* PN together with a vector $\boldsymbol{\lambda} \in \mathbb{R}^{|T|}_{>0}$. Different semantics have been defined for timed *continuous* transitions, the two most important being *infinite server* or *variable speed*, and *finite server* or *constant speed* (see [4], [5]). Here *infinite server semantics* will be considered. Like in purely markovian *discrete* net models, under *infinite server semantics*, the flow through a timed transition t_i is the product of the rate, λ_i, and $enab(t_i, \mathbf{m})$, the instantaneous enabling of the transition, i.e., $\mathbf{f}_i(\mathbf{m}) = \lambda_i \cdot enab(t_i, \mathbf{m}) = \lambda_i \cdot \min_{p \in ^\bullet t_i}\{\mathbf{m}_p/\mathbf{Pre}(p, t_i)\}$. Observe that $Enab(t_i, \mathbf{M}) \in \mathbb{N}$ while $enab(t_i, \mathbf{m}) \in \mathbb{R}_{\geq 0}$. For the flow to be well defined, every transition must have at least one input place, hence in the following we will assume $\forall t \in T, |^\bullet t| \geq 1$. The "min" in the definition leads to the concept of *configurations* (see [7]): a configuration assigns to each transition one place that, for some markings, will control its firing speed. An upper bound for the number of configurations is $\prod_{t \in T} |^\bullet t|$. The reachability space is divided into *regions* according to the *configurations*. These *regions* are polyhedrons (in bounded systems), and are disjoint, except on the borders.

The flow through the transitions can be written in a vectorial form as $\mathbf{f}(\mathbf{m}) = \boldsymbol{\Lambda}\boldsymbol{\Pi}(\mathbf{m})\mathbf{m}$, where $\boldsymbol{\Lambda}$ is a diagonal matrix whose elements are those of $\boldsymbol{\lambda}$, and $\boldsymbol{\Pi}(\mathbf{m})$ is the configuration operator matrix at $\mathbf{m}$, which is defined such that the i-th entry of the vector $\boldsymbol{\Pi}(\mathbf{m})\mathbf{m}$ is equal to the enabling degree of t_i (see [7]). A similar representation can also be obtained for the *discrete* PN, i.e., $Enab(\mathbf{M}) = \lfloor \boldsymbol{\Pi}(\mathbf{M})\mathbf{M} \rfloor \simeq \boldsymbol{\Pi}(\mathbf{M})\mathbf{M}$ (the equality holds for ordinary PN's, but for weighted arcs there exists a relative error, decreasing with respect to $\mathbf{M}$, for rounding to the nearest lower integer). The state equation of a $TCPN$ system, which is linear inside each *region* ($\boldsymbol{\Pi}(\mathbf{m})$ is constant), is

$$\overset{\bullet}{\mathbf{m}} = \mathbf{C}\boldsymbol{\Lambda}\boldsymbol{\Pi}(\mathbf{m})\mathbf{m} \tag{1}$$

III. Controllability and Control in $TCPN$

Control action in $TCPN$ systems may only be a reduction of the flow through the transitions. That is, transitions (machines for example) cannot work faster than their nominal speed. Transitions in which a control action can be applied are called *controllable*. The effective flow through a transition which is being controlled can be represented as: $\mathbf{w}(t_i) = \lambda_i \cdot enab(t_i, \mathbf{m}) - u(t_i)$, where $0 \leq u(t_i) \leq \lambda_i \cdot enab(t_i, \mathbf{m})$. The control vector $\mathbf{u} \in \mathbb{R}^{|T|}$ is defined such that $\mathbf{u}_i$ represents the control action on t_i. If t_i is not controllable then $\mathbf{u}_i = 0$. The forced flow vector is expressed as $\mathbf{w}(\mathbf{m}, \mathbf{u}) = \boldsymbol{\Lambda}\boldsymbol{\Pi}(\mathbf{m})\mathbf{m} - \mathbf{u}$. The set of all controllable transitions is denoted by T_c, and the set of uncontrollable transitions is $T_{nc} = T - T_c$. The behavior of a TCPN forced system is described by the state equation:

$$\begin{array}{c} \overset{\bullet}{\mathbf{m}} = \mathbf{C}\boldsymbol{\Lambda}\boldsymbol{\Pi}(\mathbf{m})\mathbf{m} - \mathbf{C}\mathbf{u} \\ \mathbf{0} \leq \mathbf{u} \leq \boldsymbol{\Lambda}\boldsymbol{\Pi}(\mathbf{m})\mathbf{m} \end{array} \tag{2}$$

A control action that fulfills the constraints $\mathbf{0} \leq \mathbf{u} \leq \boldsymbol{\Lambda}\boldsymbol{\Pi}(\mathbf{m})\mathbf{m}$ and $\forall t_i \in T_{nc}$ $\mathbf{u}_i = 0$ is called *suitable bounded* (s.b). If an input is not s.b. then it cannot be applied. A marking $\mathbf{m}$ for which $\exists \mathbf{u}$ s.b. at $\mathbf{m}$ such that $\dot{\mathbf{m}} = \mathbf{C}[\boldsymbol{\Lambda}\boldsymbol{\Pi}(\mathbf{m})\mathbf{m} - \mathbf{u}] = \mathbf{0}$ is called an *equilibrium marking*.

A. Controllability

Regarding to control in TCPN systems, it is important to keep in mind that these are not controllable in the classical sense ([7]). If $\mathbf{y}$ is a *P-flow*, then for any reachable marking $\mathbf{m}$, $\mathbf{y}^T\mathbf{m} = \mathbf{y}^T\mathbf{m_0}$. So, whenever a TCPN system has *P-flows*, linear dependencies between marking variables appear, introducing token conservation laws, a class of state invariants. However, we are interested in the controllability "over" this invariant, which is called $Class(\mathbf{m_0})$. Notice that every reachable marking belongs to $Class(\mathbf{m_0})$, however, the reverse is not true for timed models.

Controllability has been studied in [6], where a local controllability definition, which considers the input constraints, was introduced. It was proved that if all the transitions are controllable then *consistence* (i.e., $\exists \mathbf{x} > \mathbf{0}$ s.t. $\mathbf{Cx} = \mathbf{0}$) is sufficient and necessary for controllability over $Class(\mathbf{m_0})$. For systems with uncontrollable transitions, controllability was studied over sets of *equilibrium markings*, which represent "the stationary operating points". The set of all equilibrium markings in $Class(\mathbf{m_0})$ is denoted as $E_qS = \{m \in Class(\mathbf{m_0}) | \exists \mathbf{u}$ s.b., such that $\mathbf{C}(\mathbf{\Lambda \Pi}(\mathbf{m})\mathbf{m} - \mathbf{u}) = 0\}$.

This set is divided according to *regions*. Then, for each particular configuration $\mathbf{\Pi}_i$ there is its corresponding *region* $\Re_i = \{\mathbf{m} \in Class(\mathbf{m_0}) | \mathbf{\Pi}(\mathbf{m}) = \mathbf{\Pi}_i\}$ and the corresponding set of equilibrium markings $E_i = \{\mathbf{m} | \mathbf{m} \in E_qS \cap \Re_i\}$. The $Class(\mathbf{m_0})$, the *regions* and the equilibrium sets E_i are convex, and most of the cases, the union of sets E_i are connected. Inside each *region* the state equation (2) is linear ($\mathbf{\Pi}(\mathbf{m})$ is constant), then the controllability was studied first over each E_i and next, over their union [6].

B. Control Laws

The problem of synthesizing control laws for the $TCPN$ system is beyond the scope of this work. Nevertheless, some references are provided in this subsection.

Continuous Petri nets have three main characteristics, which must be considered during the design of a control law: 1) the model is piecewise linear (PWL), 2) the input must be nonnegative and upper bounded by a function of the state (constrained), and 3) models with some real meaning are high-order systems (with tens or even hundreds of state-variables). Different approaches and techniques have been developed for systems with input constrains and PWL ones.

Control laws for $TCPN$ systems have been proposed by using different techniques. The optimal steady-state control problem has been addressed and solved in [7]. In [8] a solution based on Model Predictive Control was proposed, obtaining thus robust control laws. Nevertheless, this technique becomes prohibitive when the number of places is large. In [9] a tracking control approach was introduced, considering step and ramp references and low-and-high gain controllers, local stability and input boundedness were proved for a class of PNs. In those papers all transitions are assumed to be controllable. Uncontrollable transitions were considered in [10], where a gradient-based controller was proposed for driving the output towards the desired value. In [11], uncontrollable transitions were also considered. There, the classical linear feedback control structure was adapted (computing a gain matrix for each *region*), avoiding computational complexity problems and providing feasibility and effectiveness.

In any case, in order to interpret a control law into the corresponding MPN, it is required the input to be s.b. and robust "enough" (remember that the original system is a stochastic one). In the sequel the control law will be expressed in general form as $\mathbf{u} = f(\mathbf{m})$, where f is a function $f: Class(\mathbf{m_0}) \to \mathbb{R}^{|T|}$ such that the input is s.b..

IV. Implementation of control to MPN via TCPN

In this section, the implementation of the control law designed for a $TCPN$ system to the corresponding MPN is described. It requires an interpretation of such control input in terms of the MPN, and of the marking of the MPN in terms of the $TCPN$. The second one is based on the approximation of the mean value of the marking of the MPN by means of the corresponding of the $TCPN$ system, which is detailed in the following subsection.

A. Approximation of MPN via TCPN

The approximation of the MPN by means of the $TCPN$ was studied in [12]. There, the $TCPN$ was analyzed in discrete-time, obtaining thus ($\Delta\tau$ is the sampling period):

$$\mathbf{m}_{k+1} \simeq \mathbf{m}_k + \mathbf{C\Lambda\Pi}(\mathbf{m}_k)\mathbf{m}_k\Delta\tau - \mathbf{C}\Delta\tau\mathbf{u}_k \tag{3}$$

There, it was proved that given $\mathbf{m}_0 = \mathbf{M}_0$, the marking of a $TCPN$ system $\langle \mathcal{N}, \boldsymbol{\lambda}, \mathbf{m}_0 \rangle$, whose evolution is described by (3), but without the input, approximates the average marking of the MPN $\langle \mathcal{N}, \boldsymbol{\lambda}, \mathbf{M}_0 \rangle$ during the time interval $(\tau_0, \tau_0 + n\Delta\tau)$, if the following conditions are fulfilled at $\mathbf{M}(\tau_0 + k\Delta\tau)$ for any time step k in the interval: 1) the probability that the transitions of the MPN are all enabled is near one, 2) the probability that the marking is outside the *region* of $\mathbf{M}_0$ is near zero.

Even if the quality of the approximation decreases when a change of *regions* occurs (i.e., Condition 2 does not hold during certain time) and/or the transitions are not enabled during certain time period (Condition 1), the approximation could be good enough for the analysis and control purposes.

In order to improve the approximation when condition 2 does not hold, a noise vector $\mathbf{v}_k$ is added to the flow of the $TCPN$ model, leading to a *Markovian continous* Petri net ($MCPN$). The noise has as elements independent normally distributed random variables with mean and covariance:

$$\begin{aligned} E\{\mathbf{v}_k\} &= \mathbf{0} \\ \mathbf{\Sigma}_{\mathbf{v}_k} &= diag[\mathbf{\Lambda\Pi}(\mathbf{m}_k)\mathbf{m}_k\Delta\tau] \end{aligned} \tag{4}$$

The $MCPN$ model is defined as:

$$\mathbf{m}_{k+1} = \mathbf{m}_k + \mathbf{C\Lambda\Pi}(\mathbf{m}_k)\mathbf{m}_k\Delta\tau + \mathbf{C}\mathbf{v}_k \tag{5}$$

By analyzing the moments of this system and the MPN one, and using the Central Limit Theorem, it was shown that the first two moments (mean value and covariance) of the marking of the $MCPN$ system *approximate* those of the marking of the corresponding MPN during a time interval $(\tau_0, \tau_0 + n\Delta\tau)$, if the initial conditions of both coincide and Condition 1 is fulfilled. Then, in this work only live PN systems will be considered (it is required for Condition 1).

B. Control Architecture

The application of the control law designed for the TCPN to the MPN is described in the Block Diagram of fig. 1. It represents a typical structure of a closed-loop system with an estimation-based control being applied.

Blocks in the upper dashed box (*Plant*) represent the original system (modeled by a MPN). There are different ways for simulating a MPN, but that is beyond the scope of this work. However, in this case it is only necessary to keep in mind that the future marking of the MPN depends on

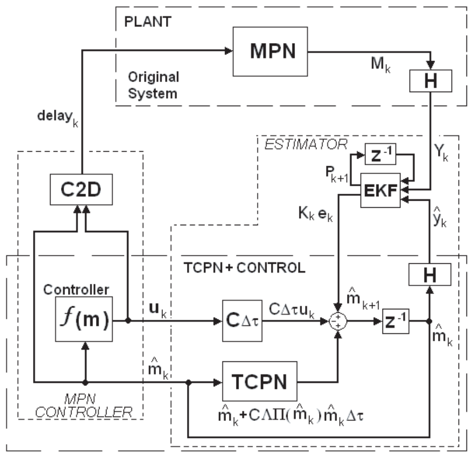


Fig. 1. Block Diagram of the closed-loop system

the current marking and some information about the times to fire each transition, commonly called *clocks*. In the Block diagram it is considered a linear output function, i.e., the information that we have about the state of the MPN is:

$$\mathbf{Y}_k = \mathbf{H} \cdot \mathbf{M}_k \tag{6}$$

It is assumed that the output has enough information to determine the current configuration and reconstruct the marking. The lower dashed box (*TCPN+Control*) represents the $TCPN$ system (3). The same output function is also applied, so, the output of the $TCPN$ is:

$$\widehat{\mathbf{y}}_k = \mathbf{H} \cdot \widehat{\mathbf{m}}_k \tag{7}$$

Notice that, if blocks *C2D* and *EKF* were eliminated, only the MPN and the $TCPN$ blocks would be present (i.e., blocks in dashed boxes *Plant* and *TCPN+Control*). In such case, two independent systems would be obtained, whose outputs would be linear functions on particular realizations (or marking trajectories) of both systems, but no interaction between them would occur. Then, blocks *C2D* and *EKF* play the role of an interface between both systems. *C2D* transfers the information from the $TCPN$ system to the MPN one, while *EKF* do the same in the opposite direction.

C. Interface blocks, C2D and EKF

First, let us consider the Block Diagram in fig. 1 without the block *EKF*. As it was pointed out in subsection IV-A, the expected value of the marking of the MPN can be approximated by the marking of corresponding $TCPN$ if Conditions 1 and 2 are fulfilled. So, let us suppose at this moment that both conditions are fulfilled.

Now, assume that a s.b. control law is being applied to the $TCPN$ system. Consider the state equation of the continuous model as in (2). Given a controllable transition t_j, the controlled flow is equal to $\mathbf{w}(t_j) = \lambda_j \cdot enab(t_j, \mathbf{m}) - \mathbf{u}(t_j)$. However, since the input is s.b., there exists a function $\alpha(\mathbf{u}(t_j), \mathbf{m})$ that takes values in the interval $[0, 1]$ such that $\mathbf{u}(t_j) = \alpha(\mathbf{u}(t_j), \mathbf{m})\lambda_j \cdot enab(t_j, \mathbf{m})$, then $\mathbf{w}(t_j) = [1 - \alpha(\mathbf{u}(t_j), \mathbf{m})]\lambda_j \cdot enab(t_j, \mathbf{m})$. This last equality means that each active server of t_j fires with a mean time delay of $([1 - \alpha(\mathbf{u}(t_j), \mathbf{m})]\lambda_j)^{-1}$ in the controlled $TCPN$ system, instead of the mean time delay of λ_j^{-1} that it would have without control. Then, the control law *imposes* to each active server of t_j an additional delay of:

$$delay(t_j) = \frac{1}{[1 - \alpha(\mathbf{u}(t_j), \mathbf{m})]\lambda_j} - \frac{1}{\lambda_j} \tag{8}$$

If additional delays are defined for all the controllable transitions in the same way, and they are added to the corresponding mean time delays of the MPN system, then the mean value of its marking will still be approximated by the marking of the $TCPN$ in the closed-loop. Block *C2D* computes such additional delays, so, according to (8) and substituting α, the output of *C2D* is defined as:

$$delay_k(t_j) = \frac{enab(t_j, \mathbf{m}_k)}{\lambda_j \cdot enab(t_j, \mathbf{m}_k) - \mathbf{u}_k(t_j)} - \frac{1}{\lambda_j} \tag{9}$$

Notice that it is only necessary to compute the additional delays for the controllable transitions T_c. In order to exemplify the application of these additional delays into the MPN system, suppose that at some time step k the controllable transition t_j in the MPN is newly enabled, then the time to fire t_j in the open-loop system would be given by a random variable having an exponential p.d.f. with parameter $(1/\lambda_j) \cdot (1/Enab(t_j, \mathbf{M}_k))$, but, in order to apply the control law the parameter of the exponential p.d.f. is considered as $(1/\lambda_j + delay_k(t_j)) \cdot (1/Enab(t_j, \mathbf{M}_k))$. In this way, t_j will fire with the required mean time delay, in agreement with the input applied to the $TCPN$ system. This control interpretation is a particular one of many that can be defined, however, this is used for simplicity and because it has shown positive results.

Block *C2D* may be enough for applying the control law into the MPN if Conditions 1 and 2 are always fulfilled. However, notice that the MPN does not receive any feedback in this way (remember that at this point, block *EKF* is not considered), so, in order to improve the accuracy, an *Extended Kalman Filter* (EKF) is added in the Block Diagram of fig. 1 (for a detailed introduction to Kalman Filter see, for instance, [16]). This new block also allows to consider several marking *regions*.

In order to analyze block *EKF*, suppose that no control law is being applied to both systems. Now, as it was pointed out in subsection IV-A, the MPN can be approximated by the corresponding $MCPN$, i.e., $E\{\mathbf{m}_k\} \simeq E\{\mathbf{M}_k\}$ and $\sum_{\mathbf{m}_k} \simeq \sum_{\mathbf{M}_k}$. Then, defining the approximation error $\boldsymbol{\varepsilon}_k = \mathbf{M}_k - \mathbf{m}_k$, the evolution of the output of the MPN is:

$$\begin{aligned} \mathbf{m}_{k+1} &= [\mathbf{I} + \mathbf{C}\boldsymbol{\Lambda}\boldsymbol{\Pi}(\mathbf{m}_k)\Delta\tau]\, \mathbf{m}_k + \mathbf{C}\mathbf{v}_k \\ \mathbf{Y}_{k+1} &= \mathbf{H} \cdot \mathbf{M}_{k+1} = \mathbf{H} \cdot \mathbf{m}_{k+1} + \mathbf{H} \cdot \boldsymbol{\varepsilon}_{k+1} \end{aligned} \tag{10}$$

Notice that $E\{\boldsymbol{\varepsilon}_k\} \simeq \mathbf{0}$ by definition. Previous system is actually the corresponding $TCPN$ system plus zero-mean at the state ($\mathbf{C}\mathbf{v}_k$, which is also uncorrelated in time) and the output ($\mathbf{H}\boldsymbol{\varepsilon}_{k+1}$), then, it seems obvious to use and EKF, in order to obtain a noise-free estimation of the underlying

$TCPN$ model. In this way, an estimator is defined as:

$$\begin{aligned}\widehat{\mathbf{m}}_{k+1} &= \left[\mathbf{I} + \mathbf{C}\boldsymbol{\Lambda}\boldsymbol{\Pi}(\widehat{\mathbf{m}}_k)\Delta\tau\right]\widehat{\mathbf{m}}_k + \mathbf{K}_k\mathbf{e}_k \\ \widehat{\mathbf{y}}_{k+1} &= \mathbf{H}\cdot\widehat{\mathbf{m}}_{k+1}\end{aligned} \tag{11}$$

where $\mathbf{K}_k$ is Kalman gain matrix and $\mathbf{e}_k = \mathbf{Y}_k - \widehat{\mathbf{y}}_k$ is the output error. In order to ensure convergence, it is assumed that the output $\mathbf{Y}_k$ has enough information to determine the current configuration, and that the pair $(\mathbf{H}, \mathbf{C}\boldsymbol{\Lambda}\boldsymbol{\Pi}_i)$ is observable for all the visited configurations $\boldsymbol{\Pi}_i$.

The gain introduced by the Kalman Filter ($\mathbf{K}_k\mathbf{e}_k$) is computed in the block *EKF* as:

$$\begin{aligned}\mathbf{P}'_{k+1} &= \left[\mathbf{I} + \mathbf{C}\boldsymbol{\Lambda}\boldsymbol{\Pi}(\widehat{\mathbf{m}}_k)\Delta\tau\right]\cdot\mathbf{P_k}\cdot\left[\mathbf{I} + \mathbf{C}\boldsymbol{\Lambda}\boldsymbol{\Pi}(\widehat{\mathbf{m}}_k)\Delta\tau\right]^T \\ &\quad + \mathbf{Q}_k \\ \mathbf{K}_k &= \mathbf{P}'_{k+1}\cdot\mathbf{H}^T\cdot\left[\mathbf{H}\cdot\mathbf{P}'_{k+1}\cdot\mathbf{H}^T + \mathbf{R}_k\right]^{-1} \\ \mathbf{P}_{k+1} &= \left[\mathbf{I} - \mathbf{K}_k\cdot\mathbf{H}\right]\mathbf{P}'_{k+1} \\ \mathbf{K}_k\mathbf{e}_k &= \mathbf{K}_k\cdot(\mathbf{Y}_k - \widehat{\mathbf{y}}_k)\end{aligned} \tag{12}$$

Matrix $\mathbf{Q}_k$ represents the covariance of the state perturbation, which according to the $MCPN$ approximation and the definition of $\mathbf{v}_k$, it should be close to $\mathbf{C}\cdot diag[\boldsymbol{\Lambda}\boldsymbol{\Pi}(\widehat{\mathbf{m}}_k)\widehat{\mathbf{m}}_k\Delta\tau]\cdot\mathbf{C}^T$. Matrix $\mathbf{R}_k$ represents the covariance of the output perturbation, i.e., the covariance of ε_k. A reasonable estimation for such covariance is given by $\mathbf{R}_k = 0.5\cdot\mathbf{I}$ (assuming that the discrete marking follows a normal-multivariate distribution, such value for $\mathbf{R}$ means that the error between markings is less that 1.5 with probability close to 0.95). Since the covariance matrix of the error ($\mathbf{P}_{k+1}$) is used for the next time step, a feedback-loop with the unit delay z^{-1} is added in the Block Diagram.

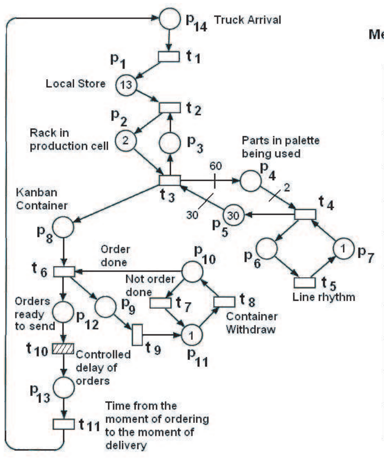
Then, block *EKF* computes the gain $\mathbf{K}_k\mathbf{e}_k$, with which the estimation for the next time step $\widehat{\mathbf{m}}_{k+1}$ can be obtained by using (11). In this way, with the output of the MPN system and block *EKF* it is possible to obtain an estimation for the state as if it were a $TCPN$ system, i.e., it is obtained $\widehat{\mathbf{m}}$ that evolves like the $TCPN$ system but approximates the mean value of the MPN one in agreement to its output values $\mathbf{Y}$.

Finally, since the evolution of the system is linear by *regions* and time invariant, according to the *Separation principle*, it is reasonable to integrate the *EKF* and the control law, obtaining thus the closed-loop system shown in fig. 1.

V. Application example

In this section, the control scheme considered through this paper is applied to a given application example: the stock level control of a Kanban-based automotive assembly line, which was introduced in [13].

Authors in that paper proposed an stochastic Petri net model for an existing assembly line that produces cars. The production is based on Kanban process. The assembly line is a self-moving transporter, which carries the car bodies through a number of quite similar production cells. The time that the car body spends in every production cell is equal for all productions cells and is given by the line rhythm, which is constant. Each production cell has some small stores (racks) where palettes with all parts, specific to the particular production cell, are to be found. In every cell there is a space



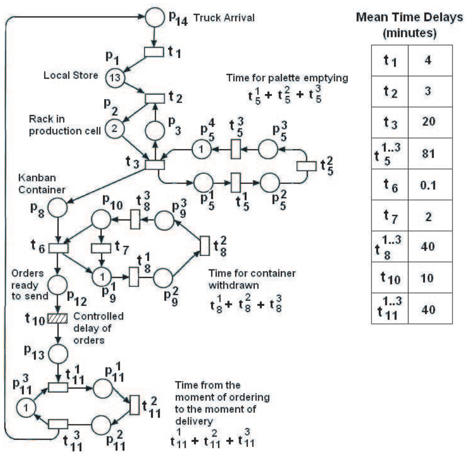
Mean Time Delays (minutes)	
t_1	4
t_2	3
t_3	20
t_4	0.1
t_5	8
t_6	0.1
t_7	2
t_8	120
t_9	0.1
t_{10}	10
t_{11}	120

Fig. 2. Petri net model of one part assembly [13].

to accommodate at maximum two palettes of each part type used there. One palette contains only the same kind of parts.

Fig. 2 shows the model proposed for describing the assembly of one part. Tokens in p_1 represents the Kanban-tickets in the local store. Tokens in p_2 represents the Ktickets in the space close to the production cell. Such number is limited by a conservative law imposed by p_3 ($\mathbf{M}(p_2)+\mathbf{M}(p_3) = 2$). Place p_4 represents the number of parts available in the palette that is being used for production. The number of parts in one palette is 60 (arc (t_3, p_4)), while the number of parts of the same kind required for one car production is 2 (arc (p_4, t_4)). Transition t_5 represents the assembly operation. Its delay is equal to the time interval between the production of two consecutive cars (i.e., the production speed or line rhythm). Place p_5 enables t_3 when the marking in p_4 is null, i.e., when no more parts are available in the palette that is being used. The container withdraw is described by the subnet defined by $\{p_9, p_{10}, p_{11}\}$, which works in the following way: transition t_8 models the waiting time before an order (orders are done just at some hours), after its firing p_{10} enables t_6 and t_7. A Kanban in p_8 means that an order must be done, in such case t_6 fires (its delay is considerably lower than that of t_7) and a token is transferred from p_8 to p_{12}, meaning that a supply order is ready to be sent. On the other hand, if there is a token in p_{10} but not in p_8 then t_7 fires, meaning that the Kanban container is withdrawn. Transition t_{11} represents the time from the moment of ordering to the moment of delivery, while a token in p_{14} represents the truck arrival. Transition t_{10} does not appear in the original model in [13]. In this work it is added for control purposes: its delay will be controlled meaning that orders (i.e., tokens in p_{12}) must be delayed before being sent.

Only some of the mean time delays are reported in [13] (for one part). However, only three transitions exhibit significant, and almost constant, delays: t_5, t_8 and t_{11}. Furthermore, t_6 must fire faster than t_7 whenever both are enabled. For our purpose, mean time delays are defined as in fig. 2, transitions t_5, t_8 and t_{11} fire with Erlang-3 p.d.f. (for reducing their coefficients of variation), transition t_6 and t_7



Mean Time Delays (minutes)

t_1	4
t_2	3
t_3	20
$t_5^{1..3}$	81
t_6	0.1
t_7	2
$t_8^{1..3}$	40
t_{10}	10
$t_{11}^{1..3}$	40

Fig. 3. Timed continuous Petri net model

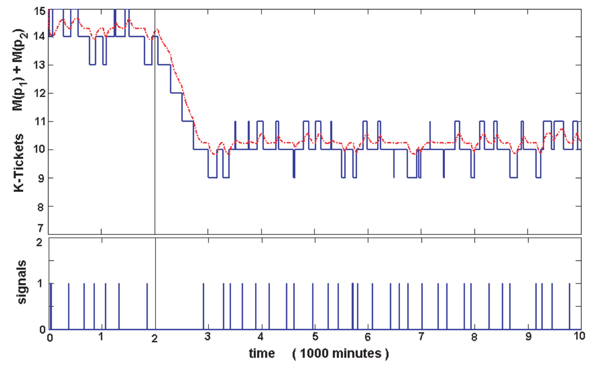


Fig. 4. Number of Kanban-Tickets in the local store and racks $(\mathbf{M}(p_1) + \mathbf{M}(p_2))$ in the closed-loop systems, and firing signals of the controllable transition t_{10} (an impulse means a firing). The control law is applied after time 2000 min.

fire with constant delays, while the other transitions fire with exponential p.d.f. and infinite server semantics.

Fig. 3 shows the proposed $TCPN$ model, in which some modifications w.r.t the *original* PN are introduced for obtaining a better approximation. In this model, the component representing the parts in the palette that is being used is substituted by the component of places $\{p_5^1, ..., p_5^4\}$, a token in p_5^4 means that the palette in use is empty. The container withdraw is modeled in a similar way, the difference is that, in order to reduce the coefficient of variation, t_8 of fig. 2 is now splitted in three transitions $\{t_8^1, t_8^2, t_8^3\}$ (classical simulation of an Erlang-3 by 3 exponentials). In a similar way, t_{11} of fig. 2 is splitted into $\{t_{11}^1, t_{11}^2, t_{11}^3\}$. Notice that places $\{p_1, p_2, p_3, p_8, p_{10}, p_{12}, p_{14}\}$ keep the same meaning, in the same way that their corresponding output transitions do. All transitions fire with exponential p.d.f. and infinite server semantics (t_6 and t_7 fire with constant delays in the original model, but their delays are very small w.r.t. the others so they can be well approximated by exponential delays). Mean time delays are defined as in fig. 3. Notice that the delays of $\{t_5^1, t_5^2, t_5^3\}$ sum the total time required for emptying the palette that is being used, i.e., the sum of the delays of t_4 and t_5 of the *original* model (fig. 2) multiplied by 30. In the same way, the sum of delays of $\{t_8^1, t_8^2, t_8^3\}$ and $\{t_{11}^1, t_{11}^2, t_{11}^3\}$ of the $TCPN$ are equal to the delays of t_8 and t_{11} of the *original* model, respectively.

The goal in [13] is to propose a methodology for optimizing the stock reserves of each part, i.e., to control the sum of ($\mathbf{M}(p_1)$+$\mathbf{M}(p_2)$) in fig. 2. Having a large number of Kanbans in the store (i.e., $\mathbf{M}(p_1)$) implies unnecessary costs, however, missing Kanbans might stop the whole production. Such Kanbans missing can occur for unexpected delays in truck arrivals or lost orders. In that paper, the optimum number of K-Tickets (i.e., the optimum value for $\mathbf{M}(p_1)$+$\mathbf{M}(p_2)$) is computed based on simulation data. However, no solution for making the system to keep the optimum number of K-Tickets is described. In this example, the control scheme introduced in the previous section is used for that purpose.

PN of fig. 2 represents the *original* model or the *Plant*, i.e., the upper dashed box MPN in the block diagram of fig. 1, while fig. 3 represents the model for the $TCPN$ system (*TCPN+Control* in the block diagram), for which the control law is designed. Output functions are defined as $[\mathbf{M}(p_8)$, $\mathbf{M}(p_{12})$, $\mathbf{M}(p_{14})$, $\mathbf{M}(p_1)$, $\mathbf{M}(p_2)$, $\mathbf{M}(p_{10})$, $\mathbf{M}(p_5)/30]$ for the original system and $[\mathbf{m}(p_8)$, $\mathbf{m}(p_{12})$, $\mathbf{m}(p_{14})$, $\mathbf{m}(p_1)$, $\mathbf{m}(p_2)$, $\mathbf{m}(p_{10})$, $\mathbf{m}(p_5^4)]$ for the estimator. Notice that they are equal for both systems excepting the last output, which is required for knowing which arc is constraining t_3. Now, following [13] let us suppose that the optimum number for the sum $\mathbf{M}(p_1)+\mathbf{M}(p_2)$ is computed as 10, then, our control law must impose additional delays in t_{10} such that the mean value of the sum in the *original* system be 10.

By using the techniques introduced in [11], a control law was designed for the $TCPN$ system (fig. 3), and then interpreted and applied to the *original* one, according to the control scheme described in the previous section (fig. 1). The results are shown in fig. 4. Control law is applied after 2000 min. Dashed line in fig. 4 corresponds to the *estimator* ($\widehat{\mathbf{m}}_k$), while the other one represents the discrete *original* system. As it can be seen, the control law successfully drives the discrete system for obtaining the desired mean value of the marking at the local store and racks. Fig. 4 also shows the firing signals for the unique controllable transition t_{10} in the closed-loop system. A unit impulse means that t_{10} is fired, i.e., that an order is released and sent to the parts supplier.

It was necessary to adjust the values for the covariance matrices $\mathbf{Q}_k$ and $\mathbf{R}_k$ of the Kalman Filter, in order to obtain a good closed-loop performance. If the values provided in the previous section are used, the estimated marking $\widehat{\mathbf{m}}_k$ will be close to the state of the MPN ($\mathbf{M}_k$). Therefore the control input $\mathbf{u}_k$ will be computed using the noisy signal $\widehat{\mathbf{m}}_k$, but the control law was designed for the $TCPN$ without noise, so, such input signal may result in a bad performance. On the other hand, decreasing $\mathbf{Q}_k$ will make that the trajectory of $\widehat{\mathbf{m}}_k$ be soft enough, so applying the computed input to the $TCPN$ system will lead to the expected results. Nevertheless, the behavior of the continuous system could be different from that of the MPN, because with a low value for $\mathbf{Q}_k$ the EKF will not ensure the approximation.

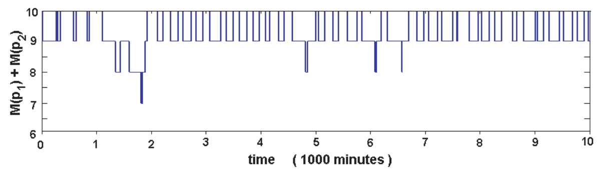


Fig. 5. Number of Kanban-Tickets in the local store and racks $(\mathbf{M}(p_1)+\mathbf{M}(p_2))$ with the $GMEC$ control.

The best performance is obtained by decreasing $\mathbf{Q}_k$ from its theoretical value (obtaining thus a soft estimated trajectory), but decreasing the entries of $\mathbf{R}_k$ corresponding to the outputs whose approximation must be improved.

Finally, for comparing purposes, let us show the results obtained by using a control feedback with Generalized Mutual Exclusion Constraints (GMEC), introduced in [2]. Such approach is defined for safety specifications, according to which a weighted sum of markings must be limited. In our case, the specification could be defined with the following GMEC $\mathbf{M}(p_1)+\mathbf{M}(p_2) \leq 10$. The controller that force the GMEC consists in the addition of a new place, called Monitor, having as input transition t_3 and output one t_1, however, t_1 is not controllable, then the Monitor must have t_{10} as output transition, in this way the GMEC is fulfilled. For the initial marking, p_1 should have 8 tokens (i.e., $10-\mathbf{M}_0(p_2)$) and the new place, the Monitor, must have 5 tokens (i.e., total K-Tickets 15, minus $\mathbf{M}_0(p_1)+\mathbf{M}_0(p_2)$), while the other places remain marked as in fig. 2. The results are shown in fig. 5. As it can be seen, the GMEC control approach guarantees that the combined marking $(\mathbf{M}(p_1)+\mathbf{M}(p_2))$ is not larger than 10. Notice that the sum is not always close to the desired mean value, because the $GMEC$ is defined for imposing upper bounds to the marking but not for enforcing a desired mean value.

In this example the GMEC control approach still provides a good mean value for $\mathbf{M}(p_1)+\mathbf{M}(p_2)$ (an average value of 9.46 was obtained), but it is not always the case. For instance, consider the same system (fig. 2) but with $\mathbf{M}(p_1) = 4$ at the initial marking. Suppose that a mean value of 4 is desired for the sum of $\mathbf{M}(p_1)+\mathbf{M}(p_2)$. After simulating $30,000$ minutes, using a GMEC $\mathbf{M}(p_1)+\mathbf{M}(p_2) \leq 5$ an average value (for the sum) of 4.59 was obtained, while a value of 3.60 resulted with GMEC $\mathbf{M}(p_1)+\mathbf{M}(p_2) \leq 4$. They represent 14.75 and 9.9 percent error, respectively. On the other hand, the control strategy introduced in this paper provides an average value of 4.02. Therefore, if a good accuracy for the average value is required, the method proposed in this paper is more suitable. On the other hand, if just safety specifications must be fulfilled, the GMEC control approach is a better choice. In any case, both methods can be combined obtaining the best properties of each one, for instance, if we would like that $\mathbf{M}(p_1)+\mathbf{M}(p_2) \leq 5$ but having a mean value of 4.

VI. Conclusions

In this work, a scheme has been provided for the interpretation of a control law designed for a $TCPN$ system into the corresponding MPN one. The resulting scheme constitutes a tool for controlling the mean value of a MPN system by means of applying additional delays to the controllable transitions, i.e., for controlling a performance index of the *original* stochastic Petri net.

This control strategy has been applied to an application example: the stock level control of a Kanban-based automotive assembly line. The results obtained are positive, showing the feasibility of the control scheme proposed. However, it is required that the control law designed for the $TCPN$ system be robust enough, since the MPN system is interpreted as a $TCPN$ with state and output perturbations. Furthermore, the covariance matrices of the EKF need to be suitably adjusted in order to obtain a good closed-loop performance.

References

[1] R. Zurawski, M.C. Zhou, Petri nets and industrial applications: A tutorial, *IEEE Transsactions on Industrial Electronics*, vol. 41(6), 1994, pp 567-583.

[2] A. Giua, F. DiCesare, M. Silva, Generalized mutual exclusion constraints on nets with uncontrollable transitions, *In: 1992 IEEE Int. Conf. on Systems, Man, and Cybernetics, Chigago, USA*, 1992, pp 974979.

[3] L.E. Holloway, B.H. Krogh, A. Giua, A Survey of Petri Net Methods for Controlled Discrete Event Systems, *Discrete Event Dynamic Systems*, vol. 9(2), 1997, pp 151–190.

[4] R. David and H. Alla, *Discrete, Continuous and Hybrid Petri Nets*, Springer, 2005.

[5] M. Silva, L. Recalde, L., Petri nets and integrality relaxations: A view of continuous Petri nets, *IEEE Trans. on Systems, Man and Cybernetics*, vol. 32(4), 2002, pp 314–327.

[6] C.R. Vázquez, A. Ramírez-Trevino, L. Recalde, M. Silva, M., On controllability of timed continuous Petri nets, *11th Int. Workshop Hybrid Systems: Computational and Control, (Egerstedt M., Mishra B. eds.), Lecture Notes in Computer Sciences*, Springer Verlag, vol. 4981, 2008, pp 528–541.

[7] C. Mahulea, A. Ramírez-Trevino, L. Recalde, M. Silva, Steady state control reference and token conservation laws in continuous Petri net systems, *IEEE Trans. on Automation Science and Engineering*, vol. 2(5), 2008, pp 726–741.

[8] C. Mahulea, A. Giua, L. Recalde, C. Seatzu, M. Silva, Optimal model predictive control of Timed Continuous Petri nets, *IEEE Transsactions on Automatic Control*, vol. 53(7), 2008, pp 1731–1735.

[9] J. Xu, L. Recalde, M. Silva, Tracking control of join-free timed continuous Petri net systems under infinite servers semantics, *Discrete Event Dynamic Systems*, vol. 18(2), 2008, pp 263–283.

[10] D. Lefebvre, D. Catherine, E. Leclerq, F. Druaux, Some contributions with petri nets for the modelling, analysis and control of HDS, *Proccedings of the International Conference on Hybrid Systems and Applications, LA, USA*, vol. 1(4), 2007, pp 451–465.

[11] C.R. Vázquez, L. Recalde, M. Silva, M., Piecewise Constrained Control for Timed Continuous Petri Net Systems, *Internal Report, Universidad de Zaragoza*, 2008.

[12] C.R. Vázquez, L. Recalde, M. Silva, Stochastic Continuous-State Approximation of Markovian Petri Nets Systems, *In 47th IEEE Conference on Decision and Control, Cancun, Mexico*, 2008, pp 901–906.

[13] M. Dub, G. Pipan, Z. Hanzálek, Stock optimization of a kanban-based assembly line, *Proceedings of the 12th International Conference on Flexible Automation&Intelligent Manufacturing, Dresden, Germany*, 2002, pp 258–267.

[14] M. Silva, *Introducing* Petri *Nets, In Practice of Petri Nets in Manufacturing*, Chapman & Hall, 1993, pp 1–62.

[15] M.K. Molloy, Performance Analysis Using Stochastic Petri Nets, *IEEE Transactions on Computers*, vol. 31(9), 1982, pp 913-917.

[16] M.S. Grewal, A.P. Andrews, A.P.,*Kalman Filtering, Theory and Practice Using MatLab*, Wiley-Interscience Publication USA, Second edition 2001.