Computation of Performance Bounds for Real-Time Systems Using Time Petri Nets

Simona Bernardi⋆, Javier Campos♯
⋆Dipartimento di Informatica, Università di Torino, Italy
♯Dpto. de Informática e Ingeniería de Sistemas, Universidad de Zaragoza, Spain
bernardi@di.unito.it, jcampos@unizar.es

***Abstract*—Time Petri Nets (TPNs) have been widely used for the verification and validation of real-time systems during the software development process. Their quantitative analysis consists in applying enumerative techniques that suffer the well known state space explosion problem. To overcome this problem several methods have been proposed in the literature, that either provide rules to obtain equivalent nets with a reduced state space or avoid the construction of the whole state space. In this paper, we propose a method that consists in computing performance bounds to predict the average operational behavior of TPNs by exploiting their structural properties and by applying operational laws. Performance bound computation was first proposed for Timed (Timed PNs) and Stochastic Petri nets (SPNs). We generalize the results obtained for Timed PNs and SPNs to make the technique applicable to TPNs and their extended stochastic versions: TPN with firing frequency intervals (TPNFs) and Extended TPNs (XTPNs). Finally, we apply the proposed bounding techniques on the case study of a robot-control application taken from the literature.**

***Index Terms*—D.2.8.b. Performance Measures, D.4.8.b. Modeling and Prediction, D.4.8.d. Operational Analysis, G.1.6.g. Linear Programming, I.6.4. Model Validation and Analysis, J.7.g. Real time.**

I. Introduction

In the verification and validation activities of real-time systems, task completion time is the basic metric of reference and the main goal is to give *guarantees* about the worst and best case completion time, before such systems are put into use [1]. Guarantees are related to the computation of upper and lower bounds of the task completion time. Bounds can be calculated, despite of exact values, also at the early stages of the software development process, often characterized by uncertainties due to the lack of complete knowledge of the whole system and of the external factors.

Place/Transition (P/T) Petri nets [2] have been extended in the literature with suitable time interpretations for the modelling and timing prediction of real-time systems. A time interpretation consists in specifying the behaviour in time in such a way that [3]: (1) the new model is compatible with the original P/T model, (2) part of the non-determinism present in the P/T model is reduced in order to take the timing constraints into account, and (3) the behaviour of the system is specified precisely enough to be able to check or compute the temporal properties under study. Time Petri Nets (TPNs), as defined in [4], reduce the non-determinism in the duration of activities of P/T Petri nets by associating a time interval with each transition. Interval limits define the earliest and the latest firing time of the transition, relative to the instant at which it was last enabled.

The same kind of reduction could be applied to the non-determinism involved in the choice among several conflicting transitions. Let us consider the case of a free-choice between two immediate transitions t_1 and t_2. One might specify, as an additional interpretation of the net system, that the firing ratio between t_1 and t_2 either is constant (as in Generalized Stochastic Petri Nets [5]) or falls into a given interval. This latter interpretation was introduced in [3] to illustrate the basic timing concepts of P/T Petri net models and it has not been elaborated later in the literature. We return to that interpretation in this paper, and we give to it the name "TPN with firing frequency intervals" (TPNF), as a particular case where new analytical techniques can be derived to compute temporal properties. One more step in the reduction of the non-determinism in the duration of activities, in addition to the time interval approach proposed by [4], is to introduce a stochastic measure for the duration within the given interval. In this sense, Extended TPNs (XTPNs) [6], [7], reduce the non-determinism by associating a probability density function to each transition firing time that takes a non-null value only within a given firing interval.

The quantitative analysis of such kind of nets consists basically in applying enumerative techniques, i.e., techniques based on the construction of the state classes graph [8], [9] and the discrete reachability graph [10] for TPNs, and techniques based on the generation of the randomized state graph [7] for XTPNs. Such techniques suffer the well known state space explosion problem even in case of bounded nets. To tackle this problem, alternative methods have been proposed in the literature, such as reduction methods [11], that provide rules to obtain nets with a reduced state space in which the timing and concurrency properties are preserved, and parametric descriptions of transition firing sequences [12], that avoid the construction of the whole state space.

In this paper, we consider TPNs as well as the extended formalisms of TPNFs and XTPNs. We propose an efficient method to compute performance bounds for the throughput of transitions and for the mean marking of places. Bounds are computed through the solution of linear programming (LP) problems derived from the structure of the net, the initial marking, and the time interpretation. The proposed technique can be applied under the *average operational behavior* assumption, that is the usual *job flow balance* assumption of operational analysis of queueing models [13]. Basically, this balance assumes that, during the observation period, the number of arrivals to each station (or place, with Petri Net terminology) is equal to the number of completions. It holds only in some observation periods, in particular, in arbitrary large observation periods. Nevertheless, it is an easily testable assumption. It is the operational counterpart of the steady-state hypothesis in stochastic models. An important advantage of the operational approach with respect to the stochastic one is that a steady-state stochastic theorem is a statement about a collection of possible infinite behaviour sequences, but it is not guaranteed to apply to a particular finite behaviour sequence. On the other hand, an operational theorem is guaranteed to apply to every behaviour sequence in the collection [13], given that in the considered observation period the job-flow balance assumption is fulfilled.

The method extends the applicability of the existing well established performance bound computation techniques for Timed Petri nets [14] (Timed PNs) and Stochastic Petri nets [15] (SPNs). Indeed, from one hand, the Timed PN/SPN bounding techniques are not applicable to TPNs or TPNFs anymore, while they can still be used for XTPNs. In the latter case, they provide better performance bounds (i.e., closer to the min/max values) w.r.t. the bounds computed with the TPN bounding technique. On the other hand, if the TPN bounding technique is applied to Timed PNs or SPNs, the performance bounds obtained are equal to the ones computed with the Timed PN/SPN bounding techniques.

This work builds on the previous proposal [16] and improves it from both theoretical and application perspectives. From the theoretical point of view, the relationship between the proposed TPN bounding technique and the existing Timed PN/SPN bounding techniques is given. From the application point of view, a performance bound solver for TPNs [17] has been implemented and used for the flexible manufacturing system running example and the robot-control application case study.

The paper is organized as follows. In Section II basic definitions and notations are given, illustrating the concepts through an example of flexible manufacturing system. Section III includes the technique for the computation of bounds for TPNs. Firstly, basic observational quantities and the average operational behavior assumption for a TPN are given. Then, we present a set of linear equations and inequalities that are used as constraints of LP problems stated to compute the performance bounds. Finally, we apply the bounding technique to the running example. In Section IV, the relationship between the TPN bounding technique and the Timed PN/SPN bounding techniques are explained. In Section V we apply the bounding techniques to the robot-control application case study. Concluding remarks are given in Section VI.

Related work

TPNs have been extensively used for the validation of real-time systems, e.g., industrial robotic arm controller systems [18], assembly systems [19], control and command systems [20], avionic mission computing applications [21] and virtual reality systems [22], both from the correctness and the quantitative point of view.

The works [18], [19] and [20] propose enumerative techniques, alternative to the original proposal [8], to support schedulability analysis. In particular, the enumerative technique presented in [18] computes tight bounds on the maximum and minimum execution time of feasible traces. Wang et al. [19] introduce the concept of clock-stamped class (CS-class) and verify timing requirements on the reachability tree of CS-classes using on-the-fly techniques to avoid the complete generation of the state space of the TPN model. Xu et al. [20] integrate the concept of absolute firing domain, beside to the relative one, in the generation of the state class graph of a TPN and propose a compositional technique to manage the complexity of schedulability analysis.

In [23] and [24] extensions of TPN formalism are proposed in order to support the modelling of preemptive real-time systems. Bucci et al. [23] introduce Preemptive Time Petri Nets (PTPNs) and a state space-based technique for their analysis. The schedulability analysis of PTPNs is carried out in two stages: first an approximate representation of the state space of a PTPN model is derived, that is used to check necessary condition for feasibility of traces and to compute upper and lower bounds on their durations. Then, in the second stage, maximum execution times are computed for the critical traces by regenerating the actual set of state classes visited by the traces. Lime and Roux [24] define Scheduling Extended Time Petri Nets (SETPN) and propose an approximation method for computing the state space of a SETPN model as a stopwatch automaton. Stochastic TPN (sTPN) [25] is a suitable formalism for performance assessment of real-time system. The sTPN solution technique proposed in [26] is based on the construction of the *stochastic class graph*. Unlike the XTPNs, the firing times of concurrently enabled transitions of a sTPN are not independent.

In [27] a bounding technique is proposed for the computation of the time elapsed between two given events in uniprocessor concurrent systems. The modelling formalism considered in [27] is timed finite state automata and the approach consists in the formulation of integer LP problems, where the constraints represent necessary conditions that must be satisfied by all the execution timed paths of the model. Although the integer linear programming is NP-complete, authors argue that the obtained integer LP problems can be reduced to LP problems. The set of limitations introduced in [27], on the type of real-time systems that can be considered, are quite restrictive. In particular, the concurrent processes have to be aperiodic and sequential and the events representing time-out cannot be modeled.

Several works in the literature propose interval-based analysis for handling the uncertainties associated with parameter values and output metrics of performance models. Majumdar [28] surveys those existing techniques that exploit interval-based analysis for the performance evaluation of computing systems, modelled with queueing network formalisms. Such techniques basically consists in solving a set of linear as well as non-linear equalities/inequalities which express the relationships among the input parameters of the performance model and the performance measures. The main drawback of using interval arithmetic is the so-called *dependency problem* that may lead to the computation of loose performance bounds. In [29] a brute force interval splitting approach is presented to overcome the dependency problem, that unfortunately is characterized by exponential time complexity. Nevertheless, efficient interval splitting solutions have been proposed, such as in [30].

II. Basic definitions and notation

A. Time Petri Net

A Time Petri Net (TPN) is a tuple $\mathcal{T} = (P, T, B, F, I, M_0, I_0)$, where P is the set of places, T is the set of transitions, $B : T \times P \rightarrow \mathbb{N}$ is the backward incidence function and $F : T \times P \rightarrow \mathbb{N}$ is the forward incidence function (in matrix form, denoted as $\mathbf{B}$ and $\mathbf{F}$, respectively). The input sets of $p \in P$ and $t \in T$ will be denoted as ${}^\bullet p = \{t \in T : \ F(t,p) \geq 1\}$ and ${}^\bullet t = \{p \in P : \ B(t,p) \geq 1\}$, respectively; the output set of $p \in P$ will be denoted as $p^\bullet = \{t \in T : \ B(t,p) \geq 1\}$. The function $I : T \rightarrow \mathbb{Q}_0^+ \times (\mathbb{Q}_0^+ \cup \{\infty\})$ assigns to each transition $t \in T$, a time interval $I(t) = (a[t], b[t]), a[t] \leq b[t]$, where $a[t]$ is the *static earliest firing time* and $b[t]$ is the *static latest firing time*. The initial state of the TPN is specified by the pair (M_0, I_0), where $M_0 : P \rightarrow \mathbb{N}$ is the initial marking function ($\mathbf{M_0}$ in vector form) and I_0 is the function that assigns, to each transition enabled in marking M_0, the corresponding initial dynamic firing interval:

$$I_0(t) = \begin{cases} I(t) & \text{if } M_0(p) \geq B(t,p), \forall p \in P \\ (-,-) & \text{otherwise} \end{cases}$$

A state of a TPN is defined as a pair $S = (M, I_d)$, where M is the marking function and I_d is a firing interval function that associates to each transition the time (dynamic) interval $a_d[t], b_d[t]$ in which the transition is allowed to fire. Times $a_d[t]$ and $b_d[t]$ are relative to the instant at which the transition t was last enabled. So that, if t has been last enabled at time τ then it may not fire before $\tau + a_d[t]$ and it must fire at most at $\tau + b_d[t]$, unless it is disabled by the firing of a conflicting transition (that is a transition having some input places in common with t).

A transition t is *enabled* in marking M at time τ if the following inequality (in vector form) holds: $\mathbf{M} \geq \mathbf{B^T}[\cdot, t]$, where $\mathbf{B^T}[\cdot, t]$ is the row of $\mathbf{B}$ corresponding to t. The transition is *firable* in marking M at time $\tau + \theta$, if: 1) it is enabled in M at time τ and 2) $\theta \in [a_d[t], aLFT]$, where $aLFT = min_k\{b_d[t_k]\}$ is the *actual latest firing time*, that is the smallest of the latest firing times of all the transitions enabled in marking M.

The firing of t at time $\tau + \theta$ from a state $S = (M, I_d)$ leads to a state $S' = (M', I'_d)$. Firing itself does not consume time and the new marking M', in vector form, is equal to $\mathbf{M'} = \mathbf{M} + \mathbf{F^T}[\cdot, t] - \mathbf{B^T}[\cdot, t]$, where $\mathbf{F^T}[\cdot, t]$ is the row of $\mathbf{F}$ corresponding to t.

The function I'_d assigns the new firing intervals to the transitions. Each transition t' *concurrent* with t, i.e., transition that was enabled in M and is still enabled in M' after the firing of t, is characterized by the remaining firing time interval $(\max(0, a_d[t'] - \theta), b_d[t'] - \theta)$. Each *newly enabled* transition t', i.e., transition that was disabled in M and becomes enabled in M', is characterized by the static firing time interval $(a[t'], b[t'])$. Each *disabled* transition is characterized by a null firing time interval.

A transition t is *multiple enabled* in a marking M, if the greater integer K such that $\mathbf{M} \geq K \ \mathbf{B^T}[\cdot, t]$ is greater than one: K is called *enabling degree* of t in M. In absence of multiple enabledness, the memory policy associated to transitions of a TPN corresponds to the enabling policy defined for SPNs [31], since only transitions concurrent with t take into account of their enabling time from their last enabling instant. In presence of multiple enabledness, we will assume the *extended firing rule with non-deterministic strategy* [32]. According to this rule, the multiple enabled transitions of a TPN are characterized by more than one time interval. That is, a transition t enabled in a marking M with enabling degree K, has K time intervals associated $I_d^1, \ldots, I_d^K$. Each I_d^i, $i = 1, \ldots, K$ represents the time interval of the i-th enabled instance of t, and the different instances of t are independent and run concurrently.

Figure 1 depicts a TPN modelling the behavior of a flexible manufacturing system [33], where the manufacturing process of three different *types of products*, 1,2 and 3, is modelled. *Flexibility in the production process* can be seen, for instance, for the product type 3, where two different operating sequences are allowed (from *use3_M4* to *work3_M4* and from *use3_M6* to *work3_M6*). Six different *shared resources* are modelled in the system (*Mi*, $i = 1, \ldots, 6$): they represent the machines. For instance, a token in place *M1* models the availability of the machine $M1$ that can be used both for the operation on product type 1 and the operation on product type 2. Constraints on the *transport resources* for a particular production process are represented using places whose initial marking represents the capacity of the transport resource. An example of this type of constraint is place *palletsA*, marked with $\mathbf{MA}$ tokens representing the number of pallets. Global transport capacity constraints for several production processes can also be modelled (like place *palletsB*, marked with $\mathbf{MB}$ tokens). The initial marking $\mathbf{N}$, assigned to place *start*, represents instead the *system workload*. Firing time intervals $[a, b]$, associated to transitions, are shown in the Figure 1. Min/max firing delays are assigned to transitions *workX_MY*, representing the timed activity of machine *MY* within the manufacturing process of product *X*. The other transitions of the TPN may represent either logical choices (e.g., *typeX*) or resource acquisition (e.g., *acquireX_MY*) and they have zero firing delays, i.e., $[0, 0]$.

B. Time Petri Net with firing frequency intervals

Let us consider the equal conflict relation [34]:

$$t_i \ EQ \ t_j \quad \text{iff} \quad \mathbf{B^T}[\cdot, t_i] = \mathbf{B^T}[\cdot, t_j] \neq \mathbf{0}.$$

EQ is an equivalence relation that partitions the set of transitions of the net into equivalence classes ECS_j, called *equal conflict sets*. Transitions belonging to a given equal conflict

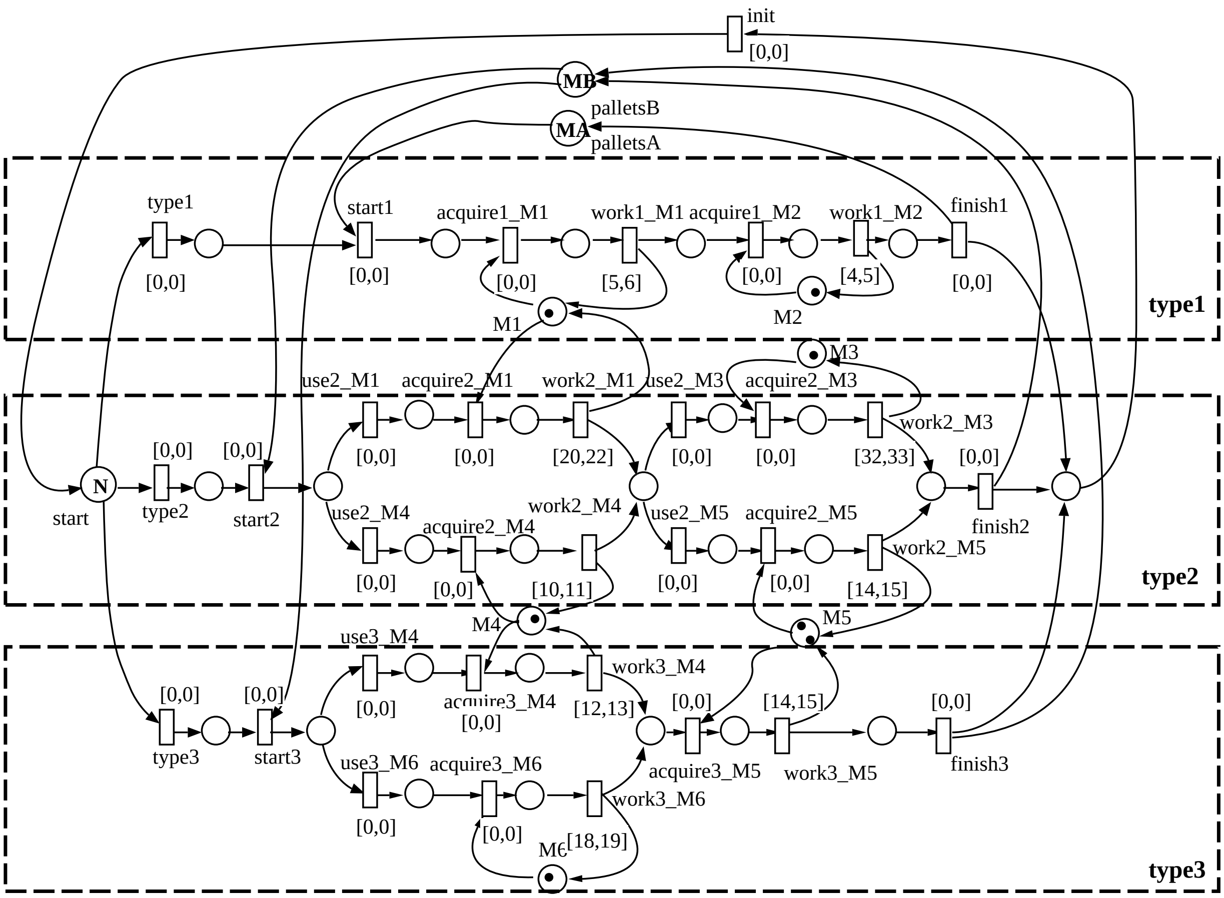

Fig. 1. The TPN model of the flexible manufacturing system.

set are in *extended free-choice conflict*. A TPN with firing frequency intervals (TPNF) is a TPN in which transitions in extended free choice conflict behave according with frequency interval constraints.

Examples of TPN models that behave as TPNF are those for which extended free choice conflicts are solved in a probabilistic manner, verifying the frequency interval constraints, or those with deterministic schedulers modelled as weighted regulation circuits, that solve the extended free choice conflicts preserving the frequency interval constraints. In this work, we define TPNFs as an extended stochastic version of TPNs. In particular, for each enabled extended free choice conflict, the ratio between the firing probability associated to a transition $t \in ECS$ and to a reference transition $t_0 \in ECS$ falls in a fixed interval. The enabling and firing rules of a TPNF are the ones of its underlying TPN.

Formally, a TPNF is a pair $\mathcal{TF} = (\mathcal{T}, R)$, where $\mathcal{T}$ is the underlying TPN model and $R : T \to \mathbb{R}_0^+ \times \mathbb{R}_0^+$ is the frequency interval function that assigns an interval $(r^i[t], r^s[t])$, $r^i[t] \leq r^s[t]$, to each transition. Without loss of generality, we assume that the transitions belonging to an equal conflict set ECS, where $| ECS | > 1$, are immediate (i.e., $a[t] = b[t] = 0$). Then, for each equal conflict set $ECS \subseteq T$, the function R has to satisfy the following constraint: there exists a transition $t_0 \in ECS$, where $R(t_0) = (1, 1)$.

When the ECS is enabled (i.e., all the $| ECS | = n$ transitions $t \in ECS$ are enabled in a given marking M), the conflict among $t \in ECS$ is resolved in a probabilistic manner by a discrete random variable (d.r.v.) from the family $\mathcal{F} = \{X_{\mathbf{p}}\}_{\mathbf{p} \in \mathcal{R}}$, where each d.r.v. $X_{\mathbf{p}}$ specifies which transition $t_j \in ECS$ will fire once enabled, as follows:

$$Pr\{X_{\mathbf{p}} = t_j \mid ECS \text{ enabled}\} = \begin{cases} p_0 > 0 & \text{if } t_j = t_0 \\ p_j & \text{otherwise} \end{cases}$$

since $X_{\mathbf{p}}$ is a d.r.v. $\sum_{j=0}^{n} p_j = 1$, and

$$\begin{aligned} \mathcal{R} \quad = \quad & \{\mathbf{p} = (p_0, \ldots, p_n) \in [0,1]^n \subseteq \mathbb{R}^n : \\ & r^i[t_j] \leq \frac{p_j}{p_0} \leq r^s[t_j], t_j \in ECS\}. \end{aligned}$$

Observe that the selection of the d.r.v. from the family $\mathcal{F}$ is carried out when the ECS becomes enabled and it is not deterministic. Moreover, the transition firing probabilities are marking and time independent.

Considering the TPN model in Figure 1, we can make several assumptions in order to reduce the non-determinism of the immediate free-choice conflicts, then transforming it into a TPNF. Firstly, we can assume that the production ratios for types 2 and 3, w.r.t. type 1, are constant and equal to two. Then, we can assign the firing frequency intervals $(2, 2)$, $(1, 1)$, and $(2, 2)$ to the immediate transitions *type1, type2* and *type3*, respectively[1]. Observe that the non-determinism of the

[1]Firing frequency intervals are not shown in Figure 1.

free-choice conflict has been eliminated in this case, by fixing the firing probability ratios between the conflicting transitions. On the other hand, within a manufacturing process of a type of product, assumptions on the execution ratio between two conflicting process steps can be made. For example, within the process of product 2, the process step which uses the machine $M3$ can be carried out, w.r.t. the one which uses the machine $M5$, with a minimum execution ratio of $\frac{3}{4}$ and a maximum execution ratio $\frac{9}{7}$. This model assumption is specified by assigning the firing frequency intervals $(1,1)$ and $(\frac{3}{4},\frac{9}{7})$ to the immediate transitions *use2_M5* and *use2_M3*, respectively. In this second case, the non-determinism of the free-choice conflict is not eliminated, however it has been reduced by bounding the firing probability ratios of the conflicting transitions to min/max values.

It is worth to note that the TPNF definition does not change when timed free-choice conflicts are considered, provided that the conflict is *well-behaved*, that is all the conflicting transitions $t \in ECS$ are eventually firable once the ECS is enabled (e.g., $\forall t \in ECS : a[t] \leq min_{t \in ECS}\{b[t]\}$). Indeed, a well-behaved timed free-choice conflict of a TPN can be always transformed into a TPN subnet characterized by an immediate free-choice conflict by preserving the timing behavior [16], as shown in Figure 2 without considering the specification of the R function. The two TPNF subnets of Figure 2 are also stochastically equivalent since the frequency intervals, $R(t_k)$, associated to the conflicting timed transitions $t_k (k = 1, \dots, n)$ are equal to the frequency intervals $R(t_{0k})$, associated to the corresponding conflicting immediate transitions t_{0k}.

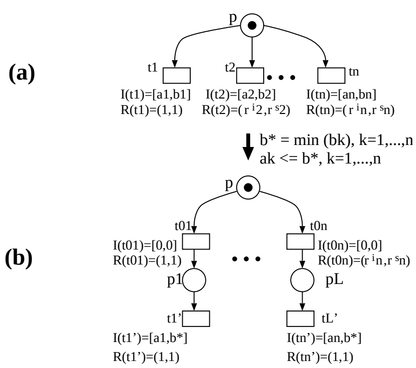

Fig. 2. Preselection policy in free-choice TPNFs.

C. Extended Time Petri Net

Extended Time Petri Net (XTPN) [6], [7] is a stochastic extension of TPN. Formally, a XTPN is defined as a pair $\mathcal{XT} = (\mathcal{T}, F_0)$, where $\mathcal{T}$ is the underlying TPN model and F_0 is a functional that assigns to each transition $t \in T$ an initial firing probability density function (pdf) defined over its static firing time interval $f_t(x), x \in I(t)$. The pdfs can be either continuous, discrete or mixed; in case of several discrete (or mixed) pdfs, it is assumed that concurrently enabled transitions have null probability of firing simultaneously [7].

The state of a XTPN is a triplet $S = (M, I_d, F_d)$, where (M, I_d) is the state of the associated TPN model $\mathcal{T}$ and F_d is a functional that defines the firing probability density function to each transition with non-empty firing interval. A transition t is firable in state S at time θ at the latest, iff 1) it firable in the underlying TPN, and 2) the probability of firing transition t before or at time θ is not zero. The new state reached by the firing of t is a state $S' = (M', I'_d, F'_d)$, where (M', I'_d) is the state reached in the underlying TPN. The functional F'_d defines the new pdfs of the transitions enabled in marking M', as follows:

$$F'_d(t_k) = \begin{cases} f_{t_k}(x), x \in I(t_k) & \text{if } t_k \text{ is newly enabled} \\ \frac{f_{t_k}(x+\theta)}{1-\Gamma_{t_k}(\theta)}, x \in I'_d(t_k) & t_k \text{ is concurrent with } t \end{cases}$$

where, $\Gamma_{t_k}(\theta)$ is the cumulated probability of $f_{t_k}(x)$ in the interval $[0, \theta]$. Concretely, the newly enabled transitions are characterized by their initial pdfs, and the transitions concurrent with t are characterized by the pdfs of their remaining times to fire. When all the enabled transitions are newly enabled (or their firing time is exponentially distributed over $[0, \infty]$) their times to fire are independent. Then, considering continuous distributions, the probability of firing t in a state $S = (M, I_d, F_d)$ is defined as:

$$P_S[t] = \int_0^{aLFT} f_t(y) \Big[\prod_{t \neq t'} \int_y^{\infty} f_{t'}(x) dx \Big] dy \qquad (1)$$

where, $aLTF$ is the actual latest firing time, $F_d(t) = f_t(y)$ and $F_d(t') = f_{t'}(x)$ are the pdfs of the transitions t' enabled with t in marking M. We will not consider, in this paper, the general case in which a non-exponentially distributed transition is continuously enabled despite the firing of another non-exponentially distributed transition.

III. Performance bound technique for TPNs

The technique presented in this section can be used to compute upper and lower bounds for performance metrics of a TPN model, such as transition throughput and place average marking. As in [15], bounds are computed from the solution of proper LP problems (max-LP problem for the upper bound and min-LP problem for the lower bound), therefore they can be obtained in polynomial time on the size of the net model. We remark that the solution of the LP problem is significant only if the TPN is live.

The idea is to compute vectors $\mathbf{x}$ and $\bar{\mathbf{M}}$ that maximize (or minimize) the throughput of a transition, or the average marking of a place, among those verifying a set of linear operational laws that are imposed as constraints. The latter can be derived from the net structure, the initial marking, the static timing specification and, in case of the TPN stochastic extensions considered in this paper (i.e., TPNF or XTPN), the stochastic information associated to transitions in free-choice conflict.

We will introduce, first, the operational quantities that are used in the formulation of the LP constraints and the *average operational behavior* assumption for a TPN model. Then, we will present the sets of linear equations and inequalities that are used as constraints of LP problems stated for the performance bound computation. Finally, we apply the proposed bounding technique to the flexible manufacturing system example.

A. Basic observational quantities

Let us consider the following quantities that can be collected during the period $(0, \Gamma), \Gamma \in \mathbb{R}^+$, by observing the behavior of a TPN model:

$$\bar{M}[p] = \frac{1}{\Gamma} \int_0^\Gamma M[p](\tau) d\tau \qquad (2)$$

the average marking of $p \in P$, where $M[p](\tau)$ is the number of tokens in p at time $\tau \in (0, \Gamma)$;

$$\bar{e}[t] = \frac{1}{\Gamma} \int_0^\Gamma e[t](\tau) d\tau \qquad (3)$$

the average enabling degree of $t \in T$, where $e[t](\tau) = min_{p \in {}^\bullet t}\{\frac{M[p](\tau)}{B(t,p)}\}$ is the number of instances of t enabled at time $\tau \in (0, \Gamma)$;

$$\theta_j[t] = \int_0^\Gamma e_j[t](\tau) d\tau \qquad (4)$$

the enabling time for the j-th instance of transition t, where $e_j[t](\tau)$ is the characteristic function that evaluates to 1 iff the j-th instance is enabled at time $\tau \in (0, \Gamma)$ (i.e., $e[t](\tau) \geq j$);

$$\bar{S}[t] = \frac{\sum_{j=1}^\infty \theta_j[t]}{\sum_{j=1}^\infty \Phi_j[t]} \qquad (5)$$

the mean service time of $t \in T$, where $\Phi_j[t]$ is the number of firings of the j-th instance of t;

$$x[t] = \frac{\Phi[t]}{\Gamma} \qquad (6)$$

the throughput of $t \in T$, where $\Phi[t] = \sum_{j=1}^\infty \Phi_j[t]$ is the number of firings of t.

B. The average operational behavior assumption

Average operational behavior assumption in TPNs corresponds to the *job flow balance assumption* in the operational analysis of a single-server queueing system. According to [13], a system satisfies the *job flow balance* during an observation period $(0, \Gamma)$, when the number of job arrivals is equal to the number of job completions. When considering a TPN, the number of jobs in a queue at a given instant τ, corresponds to the number of tokens in a place p. Tokens are produced (consumed) by the transitions in the input (output) sets of p, through their firings. In particular, when a transition in the input-set of p fires produces $F(t, p)$ tokens and when a transition in the output-set of p fires consumes $B(t, p)$ tokens. Then, a place $p \in P$ satisfies the *average operational behavior assumption* during $(0, \Gamma)$, when the number of tokens produced in p is equal to the number of tokens consumed from p:

$$\Phi_i = \sum_{t \in {}^\bullet p} \Phi[t] F(t, p) = \sum_{t \in p^\bullet} \Phi[t] B(t, p) = \Phi_o. \qquad (7)$$

As for the *job flow balance*, such assumption holds only in some periods. In particular, when 1) the number of tokens at the beginning of the observation period is equal to the number of tokens at the end of the observation period, or 2) the observation period is long enough that the difference between the produced tokens and the consumed ones is small w.r.t. to the consumed tokens:

$$\lim_{\Gamma \to \infty} \frac{\mid \Phi_i - \Phi_o \mid}{\Phi_o} = 0. \qquad (8)$$

When the *average operational behavior* assumption is not satisfied (e.g., due to the choice of Γ), it is always possible to calculate the error made assuming it holds. Indeed in [35], a necessary and sufficient condition for job flow imbalance is given (Prop.6). In the following, we reformulate such condition in TPN terms: it will be used to get an error approximation formula when the *average operational behavior* assumption does not hold.

Proposition 3.1: The condition $\mid \Phi_i - \Phi_o \mid = (M_{max} - M_{min})$ is true iff $M[p](0) = M_{max}$ and $M[p](\Gamma) = M_{min}$, or vice-versa $M[p](\Gamma) = M_{max}$ and $M[p](0) = M_{min}$.

C. Constraints for the LP problem

The set of linear equations and inequalities, that will be introduced in the following, defines the solution domain of the LP problem for the bound computation of performance metrics associated to a TPN model. First, we will present a set of linear equations (i.e., *structural constraints*) that can be derived from the net structure and, at most, on the initial marking. Then, we will give a set of linear inequalities, namely the *enabling operational law constraints*, that stem from the application of the utilization law for queueing systems [13] on Petri Nets. Such inequalities consider the static timing information of the TPN model. Finally, we will provide additional constraints (i.e., *routing constraints*) that hold for the TPN stochastic extensions considered in this paper, and take into account the stochastic information associated to transitions in extended free-choice conflict.

1) Structural constraints: A first set of constraints is derived by considering that for all markings reachable at instant $\tau \in (0, \Gamma)$, denoted in vectorial form as $\mathbf{M_r}(\tau)$, we have that $\mathbf{M_r}(\tau) = \mathbf{M_0} + (\mathbf{F} - \mathbf{B})^\mathbf{T} \sigma_\mathbf{r}(\tau)$, where $\sigma_\mathbf{r}(\tau)$ is a feasible firing count vector until instant τ, $\mathbf{M_0}$ is the initial marking vector and $(\mathbf{F} - \mathbf{B})^\mathbf{T}$ is the incidence matrix. A second set of constraints is derived from the *average operational behavior* assumption.

Proposition 3.2: (Reachability) The average marking vector $\bar{\mathbf{M}}$ satisfies the linear equality:

$$\bar{\mathbf{M}} = \mathbf{M_0} + (\mathbf{F} - \mathbf{B})^\mathbf{T} \sigma \qquad (9)$$

where $\sigma = \frac{1}{\Gamma} \int_0^\Gamma \sigma_\mathbf{r}(\tau) d\tau$ is the average firing count vector.

Proof: From definition (2), written in vectorial form, and from the reachability equation:

$$\begin{aligned}\bar{\mathbf{M}} &= \frac{1}{\Gamma}\int_0^{\Gamma}\mathbf{M_r}(\tau)d\tau = \\ &\frac{1}{\Gamma}\int_0^{\Gamma}\mathbf{M_0} + (\mathbf{F}-\mathbf{B})^{\mathbf{T}}\ \sigma_{\mathbf{r}}(\tau)d\tau = \\ &\mathbf{M_0} + (\mathbf{F}-\mathbf{B})^{\mathbf{T}}\frac{1}{\Gamma}\int_0^{\Gamma}\sigma_{\mathbf{r}}(\tau)d\tau = \\ &\mathbf{M_0} + (\mathbf{F}-\mathbf{B})^{\mathbf{T}}\sigma. \quad \blacksquare\end{aligned}$$

Proposition 3.3: (Token flow) Let us consider a place $p \in P$ that satisfies the *average operational behavior* assumption, then:

$$\sum_{t\in{}^{\bullet}p} x[t]F(t,p) = \sum_{t\in p^{\bullet}} x[t]B(t,p). \tag{10}$$

Proof: The equalities are immediate considering the equalities (7) and the throughput definition (6). ■

It is worth to note that, when the *average operational assumption* is not satisfied for a place, the corresponding token flow equation is not finally included in the set of LP constraints. As an alternative, we can state a token flow constraint characterized by a correction term, which is derived from Proposition 3.1, as follows:

$$\sum_{t\in{}^{\bullet}p} x[t]F(t,p) - \sum_{t\in p^{\bullet}} x[t]B(t,p) = \begin{cases}\frac{M[p](\Gamma)-M[p](0)}{\Gamma} & \text{if } M[p](\Gamma) > M[p](0)\\ \frac{M[p](0)-M[p](\Gamma)}{\Gamma} & \text{otherwise}\end{cases}$$

Finally, the average marking vector, the average firing count vector and the transition throughputs are, by definition, always non negative values.

Proposition 3.4: (Non negativity)

$$\bar{\mathbf{M}}, \sigma \geq 0, \quad x[t] \geq 0 \quad \forall t \in T. \tag{11}$$

2) Enabling operational law constraints: This set of constraints consider the static interval function I of a TPN model. Concretely, the static earliest firing times are used to define the first set of linear inequalities, while the second set of linear inequalities exploits the static latest firing times. The first set of constraints can be applied to all transitions of a TPN model and for every observation period. The second set of constraints holds only for persistent transitions, i.e., once enabled they eventually fire, and for particular observation periods.

Proposition 3.5: (Throughput upper bound inequality)

$$\forall t \in T \text{ and } \forall p \in {}^{\bullet}t: \quad x[t] \leq \frac{\bar{M}[p]}{a[t]B(t,p)}. \tag{12}$$

Proof: Let us consider a transition $t \in T$ with $a[t]$ its earliest static firing time. If the j-th instance of t becomes enabled at $\tau \in (0,\Gamma)$, it cannot fire before $a[t]+\tau$: that is its minimum firing waiting time is $a[t]$. Then, the maximum number of firings of the j-th instance of t during the observation period is given by $\left\lfloor \frac{\theta_j[t]}{a[t]} \right\rfloor$, so that:

$$\Phi_j[t] \leq \Phi_j^{max}[t] = \left\lfloor \frac{\theta_j[t]}{a[t]} \right\rfloor \leq \frac{\theta_j[t]}{a[t]}.$$

Summing over all the instances of t and dividing by Γ the first and the last member of the above inequalities, we obtain:

$$x[t] \leq \frac{\sum_{j=1}^{\infty}\theta_j[t]}{\Gamma}\frac{1}{a[t]}.$$

Replacing $\theta_j[t]$ with its definition and exchanging the integral and the sum we get:

$$x[t] \leq \frac{\int_0^{\Gamma}\sum_{j=0}^{\infty} e_j[t](\tau)d\tau}{\Gamma}\frac{1}{a[t]}.$$

The equalities $e[t](\tau) = \sum_{j=0}^{\infty} e_j[t](\tau)$ and $e[t](\tau) = min_{p\in{}^{\bullet}t}\{\frac{M[p](\tau)}{B(t,p)}\}$ hold for all τ, then we obtain:

$$x[t] \leq \frac{\int_0^{\Gamma} M[p](\tau)d\tau}{a[t]B(t,p)\Gamma} = \frac{\bar{M}[p]}{a[t]B(t,p)}, \forall p \in {}^{\bullet}t. \quad \blacksquare$$

Proposition 3.6: (Throughput lower bound inequalities) Let us consider a persistent transition $t \in T$ and an observation period $(0,\Gamma)$ that satisfies one of the following properties: 1) $\Gamma \gg 0$ (sufficiently large), or 2) there exists $p \in {}^{\bullet}t$ such that $M[p](\Gamma) = 0$.

- Let $t \in T: {}^{\bullet}t = \{p\}$. Then:

$$x[t]b[t] \geq \frac{\bar{M}[p] - B(t,p) + 1}{B(t,p)}. \tag{13}$$

  If $\exists N[p]\ \forall\tau \in (0,\Gamma) : M[p](\tau) \leq N[p]$ we have the further constraint:

$$x[t]b[t] \geq k\frac{\bar{M}[p] - kB(t,p) + 1}{N[p] - kB(t,p) + 1} \tag{14}$$

  where $k \in \mathbb{N}: \ kB(t,p) \leq N[p] < (k+1)B(t,p)$.

- Let $t \in T : {}^{\bullet}t = \{p_1, p_2\}$ and $\exists N[p_1], N[p_2]\ \forall\tau \in (0,\Gamma) : M[p_1](\tau) \leq N[p_1]$, $M[p_2](\tau) \leq N[p_2]$, and $N[p_1] \leq N[p_2]$. Then:

$$x[t]b[t]B(t,p_1) \geq \bar{M}[p_1] - B(t,p_1) + 1 - N[p_1]f_2 \tag{15}$$

  where $f_2 = \left(1 - \frac{\bar{M}[p_2]-B(t,p_2)+1}{N[p_2]-B(t,p_2)+1}\right)$.

Proof: The above constraints are derived from similar constraints defined in [15] - see the inequalities (13'-15') in Table II - by proving that $\bar{S}[t] \leq b[t]$. Indeed, being t persistent, for each j-th instance of t, once enabled at time instant τ_{ji}, eventually fires at a time instant always less than or equal to $\tau_{ji} + b[t]$. So that its firing waiting time $S_{ji} \leq b[t], \forall i$. Then, the enabling time for the j-th instance of t during the observation interval $(0,\Gamma)$ can be written as:

$$\theta_j[t] = \sum_{i=1}^{\Phi_j[t]} S_{ji} + \delta_j \leq \Phi_j[t]b[t] + \delta_j,$$

where S_{ji} represents the firing waiting time of the j-th instance enabled at $\tau_{ji} : \tau_{ji} + b[t] < \Gamma$, and $\delta_j \in [0, b[t])$ represents a possible not complete firing waiting time because of the choice of Γ. Summing over all the instances of t and dividing by number of firings of t both the members of the above inequality, we get: $\bar{S}[t] \leq b[t] + \frac{\sum_{j=0}^{\infty}\delta_j}{\Phi[t]}$, where $\frac{\sum_{j=0}^{\infty}\delta_j}{\Phi[t]}$ represents the residual service time of t due to the choice of Γ. If the observation period is sufficiently large (i.e., property

1 holds), the residual service time of t approaches to zero. On the other hand, if there exists an empty input place of t at the end of the observation interval (i.e., property 2 holds) then $\delta_j = 0, \forall j$. ∎

Observe that it is always possible to apply a *preselection policy* to timed transitions in extended free-choice conflict to make them persistent [16], that preserves the timing behavior of a TPN. The *throughput lower bound inequalities* (13-15) can be stated then for such transitions and added to the set of LP constraints.

3) Routing constraints for TPN stochastic extensions: Additional constraints can be defined for transitions that are in extended free-choice conflict, when a TPNF (XTPN) model is considered. Indeed, such constraints exploit the stochastic information of the net, that is the frequency interval function (TPNF) or the firing probabilities (XTPN) associated to the conflicting transitions.

Proposition 3.7: (TPNF routing inequalities) Let us consider a TPNF $\mathcal{TF} = (\mathcal{T}, R)$ and a *well-behaved* equal conflict set $ECS \subseteq T$. Then, for each pair of transitions $t_j, t_k \in ECS$, the following inequalities hold:

$$r^i[t_j]x[t_k] \leq r^s[t_k]x[t_j], \quad r^i[t_k]x[t_j] \leq r^s[t_j]x[t_k]. \tag{16}$$

Proof: Since the conflict is well-behaved, all the transitions $t \in ECS$ are eventually firable once the ECS is enabled. From definition of TPNF, the following inequalities hold for the ratio between the firing probability of t and $t_0 \in ECS$, where $R(t_0) = (1, 1)$: $r^i[t] \leq \frac{p[t]}{p[t_0]} \leq r^s[t]$. On the other hand, the firing probability of a transition $t \in ECS$ can be expressed as $p[t] = \frac{\Phi[t]}{\sum_{t_k \in ECS} \Phi[t_k]}$, so that $\frac{\Phi[t]}{\Phi[t_0]} = \frac{p[t]}{p[t_0]}$. By definition (6), we obtain the same inequalities for the throughputs:

$$r^i[t] \leq \frac{x[t]}{x[t_0]} \leq r^s[t] \quad \text{and} \quad \frac{1}{r^s[t]} \leq \frac{x[t_0]}{x[t]} \leq \frac{1}{r^i[t]}.$$

Considering that $\frac{x[t_j]}{x[t_k]} = \frac{x[t_j]}{x[t_0]} \frac{x[t_0]}{x[t_k]}$, it is straightforward to obtain inequalities (16). ∎

Proposition 3.8: (XTPN routing equalities) Let us consider a XTPN $\mathcal{XT} = (\mathcal{T}, F_0)$ and a *well-behaved* equal conflict set $ECS \subseteq T$. If:

C1) $\forall t' \in T \setminus ECS$ cannot become enabled concurrently with $t \in ECS$, and

C2) $\exists t \in ECS \; \exists p \in {}^\bullet t : B(t, p)$-bounded

then $\forall t_j, t_k \in ECS$, the following equality holds:

$$\frac{x[t_i]}{P[t_i]} = \frac{x[t_j]}{P[t_j]} \tag{17}$$

where $P[t]$ is the firing probability of t, defined as follows (continuous case):

$$P[t] = \int_{a[t]}^{b_{min}} f_t(y) \Big[\prod_{t' \in ECS, t' \neq t} \int_y^\infty f_{t'}(z) dz \Big] dy. \tag{18}$$

being $F_0(t) = f_t(\cdot)$ the initial firing pdf of $t \in ECS$ and $b_{min} = min_{t \in ECS} b[t]$.

Proof: By definition of XTPN, concurrently enabled transitions have null probability of firing simultaneously, then the free-choice conflicts are always resolved in a probabilistic manner. The routing constraints stated for Timed/SPNs - see constraints (17') in Table II - can be applied to XTPNs as well, where the routing rates r_i are replaced by the firing probabilities $P[t_i]$, provided that the $P[t_i]$ are marking and time independent.

Constraint **C1** requires that transitions $t' \notin ECS$ and transitions $t \in ECS$ are never enabled in the same marking. The firing probability for $t \in ECS$ can be computed then by considering the local behavior of the net. Constraint **C2** guarantees that transitions belonging to the ECS have enabling degree at most equal to one. We can then associate only a single pdf for each $t \in ECS$, since there is always only one instance of t enabled.

Then, the firing probability $P[t]$ of a transition $t \in ECS$ is marking and time independent and can be computed without generating the state space of the net. The firing pdf associated to the conflicting transitions can be defined either in the continuous or discrete domain, so to compute $P[t]$ we should use a different formula. In case of continuous pdf, $P[t]$ can be calculated using formula (18). ∎

Observe that restrictions **C1** and **C2** can be verified by applying sufficient structural conditions, e.g., structural mutual exclusion condition based on P-invariants for **C1** and structural marking bound computation for **C2**. Actually, similar conditions were formulated in [36] to identify a class of regenerative Stochastic Petri Nets without the generation of the state space.

D. Application to the flexible manufacturing system example

We can apply the bounding technique on the flexible manufacturing system (FMS) example of Figure 1, where the TPN model is converted into a TPNF by assigning firing frequency intervals to transitions, according to the specification given in [33]. Table I shows the firing frequency intervals of conflicting transitions (non conflicting transitions have $R(t) = (1, 1)$). The system workload is set to $\mathbf{N} = 50$ and the number of pallets A and B is set to $\mathbf{MA} = \mathbf{MB} = 10$.

Transitions	Firing frequency intervals
type1, type2, type3	(2,2), (1,1), (2,2)
use2_M4, use2_M1	(1,1), ($\frac{2}{3}$, $\frac{2}{3}$)
use2_M5, use2_M3	(1,1), ($\frac{3}{4}$, $\frac{9}{7}$)
use3_M4, use3_M6	(1,1), (1,1)

TABLE I
FIRING FREQUENCY INTERVALS OF CONFLICTING TRANSITIONS

The metric of interest is the minimum cycle time of the system in an arbitrary large observation interval, that can be evaluated by taking the inverse of the throughput of transition *init*. The value obtained in [33] is $X(init) = 1/3.6$. We then state the max-LP problem to compute the upper bound throughput of transition *init*. The max-LP problem generated has as the objective function $f(x) = x[init]$ and it is characterized by 1223 variables and 1000 constraints. The set of constraints includes structural, enabling operational law and routing constraints. Solving the LP problem took 0.69 $sec.$ of CPU time and the upper bound of the throughput is equal to 0.27778 which is the same value obtained by Ohl.

Observe that the joint use of sensitivity analysis and the bounding technique is an efficient method to size the set of resources of the FMS for a given performance goal. In the example, we have carried out sensitivity analysis by varying the $\mathbf{MA}, \mathbf{MB}$ parameters in the range $[1, 10]$, in order to analyse the effect of the number of pallets on the minimum cycle time of the system. The analysis took few seconds, indeed the time to solve the LP problem does not depend on the initial marking of the TPNF. The result is that the minimum cycle time decreases as the number of pallets increases, until it reaches the minimum value of 3.6 when $\mathbf{MA} = \mathbf{MB} = 6$. A further increment of the number of pallets does not improve the system performance.

IV. Comparison between bounding techniques

The non-determinism present in the TPNs that we consider in this paper, is eliminated in Timed PNs or SPNs by introducing either constants or random variables. Timing associated to each transition is either a constant duration or a random variable with a given probability distribution function (PDF). Conflicts resolution is done, instead, using either race policy between stochastically timed conflicting transitions or random routing based on the weights associated with conflicting immediate transitions. In the general approach for Timed PNs/SPNs [15], bounds are computed from the solution of LP problems. Bounds depend on the mean values of service time $\overline{S}[t]$, associated to the firing of transitions, and on the routing rates r_i, associated with transitions in free conflict. They do not depend, instead, on the higher moments of the PDFs of the random variables that describe the timing of the system.

When TPNs are considered, the set of LP constraints stated for Timed/SPNs is not valid anymore. In the previous section, we presented a new formulation of those constraints (i.e., *enabling operational law* and *routing* constraints) that can be still useful. Concretely, the new formulation takes into account that, for each timed transition t, a time interval $(a[t], b[t])$ is defined instead of deterministic or stochastic duration with average $\overline{S}[t]$. Moreover, routing constraints can be also given in terms of either intervals (for TPNFs) or marking and time independent firing probabilities (for XTPNs).

The resulting LP problems for TPNs and for Timed/SPNs are shown in Table II, where the set of linear constraints include both the equation/inequalities defined for TPNs (in Section III) and the ones stated and proved for Timed/SPNs [15]. In particular, the constraints applicable to each Petri Net class are emphasized with a checkmark and, possibly, the restriction on their applicability is given. The set of constraints is not *closed* in the sense that can be extended for particular net classes or using additional information (like higher moments of the involved random variables [37] for SPNs). We remark that the solution of a LP problem can be computed in polynomial time [38]. Thus the theoretical time complexity of the solution techniques (using constraints in Table II) is polynomial on the net size, since the number of variables and constraints of the derived LP problem is linear on the net size.

We also remark that even if the observational quantities, introduced and used in the previous section to derive linear constraints, are defined for a period $(0, \Gamma), \Gamma \in \mathbb{R}^+$, the complexity of the obtained solution does not depend on the value of Γ, since this value is taken *arbitrarily large* and, then, the objective function as well as the constraints do not depend on Γ. In fact, it is just a trick commonly used in operational analysis of queueing models to perform the proof of the results using "simple" arithmetics. The results obtained using this kind of operational analysis are the deterministic counterpart of the classical average steady-state solutions computed for stochastic models.

The bounding techniques for TPNs can be considered as a generalization of existing performance bound computation techniques for Timed/SPNs [15]. First, we can observe that if we apply the "new" bounding techniques to Timed/SPNs we obtain the "old" LP problems. Indeed, considering a Timed/SPN, the solution domain of the LP problem (*LPold*), stated with the constraints marked as ✓ in the *Timed/SPN* column of Table II, and the solution domain of the corresponding LP problem (*LPnew*), stated using all the constraints of Table II, are equal. This equality can be proved by observing that:

- The *structural* constraints (*9,10,11*) are the same in the two LP problems.
- The *enabling operational law* constraints (*12,13,14,15*) of *LPnew* are weaker than the corresponding ones (*12',13',14',15'*), and, then, they do not contribute to the definition of the solution domain. Indeed, the mean service time falls in the interval $[a, b]$, that is equal to $[0, \infty)$ for exponential distributions, associated to transitions of SPNs, and it is equal to $a = b = \bar{S}$ for deterministic ones, associated to transition of Timed PNs.
- The *routing* constraints (*16*) and (*17*) become constraints *17'* when applied on immediate conflicting transitions (since weights are fixed in Timed/SPNs) and on stochastically timed transitions, respectively.

Then, since the LP problems have the same objective functions they are actually the same problem for Timed/SPNs. Secondly when TPN/TPNF are considered, the LP problem *LPold* cannot be applied. We can use the LP problem *LPnew* to obtain bounds, then extending the application of the bounding techniques to TPN/TPNF. Finally, concerning XTPNs, we can state both the LP problems *LPold* and *LPnew*, where *LPnew* includes *enabling operational law* constraints (*12-15*) that depend only on the extremes of the firing intervals. The solution domain of *LPold* is a subset of the solution domain *LPnew*, while the two problems have the same objective function $f(\bar{\mathbf{M}}, \mathbf{x})$ to be either minimized or maximized. Since the optimal solutions fall on the borders of the solution domains, we obtain that: $LB_{new} <= LB_{old} <= UB_{old} <= UB_{new}$, where LB_i and UB_i, $i \in \{old, new\}$, are the lower and upper bounds - minimal and maximal optimal solutions of LP_i.

V. Case study: a robot-control application

We have selected from the literature a mobile-robot navigation problem [39] in order to test our analysis technique and to compare the results with those previously obtained by other authors.

maximize [or minimize] $f(\bar{\mathbf{M}}, \mathbf{x})$ (with f a linear function of $\bar{\mathbf{M}}, \mathbf{x}$)

subject to the following constraints	TPN	Timed/SPN	Restrictions
Structural			
reachability:			
(9) $\bar{\mathbf{M}} = \mathbf{M_0} + (\mathbf{F} - \mathbf{B})^{\mathbf{T}}\sigma$	✓	✓	
token flow:			
(10) $\sum_{t\in {}^\bullet p} x[t]F(t,p) = \sum_{t\in p^\bullet} x[t]B(t,p),\ \forall p \in P$	✓	✓	average operational behavior for p
non negativity:			
(11) $\bar{\mathbf{M}}, \sigma \geq 0,\ \forall t \in T:\ x[t] \geq 0$	✓	✓	
Enabling operational law			
throughput upper bound inequalities:			
(12) $\forall t \in T$ and $\forall p \in {}^\bullet t:\ x[t] \leq \frac{\bar{M}[p]}{a[t]B(t,p)}$	✓		
(12') $\forall t \in T$ and $\forall p \in {}^\bullet t:\ x[t] \leq \frac{\bar{M}[p]}{\bar{S}[t]B(t,p)}$		✓	
throughput lower bound inequalities:			t persistent
$\forall t \in T, {}^\bullet t = \{p\}:$			
(13) $x[t]b[t] \geq \frac{\bar{M}[p]-B(t,p)+1}{B(t,p)}$	✓		cond. **1** or **2** in Prop.3.6
(13') $x[t]\bar{S}[t] \geq \frac{\bar{M}[p]-B(t,p)+1}{B(t,p)}$		✓	
(14) $x[t]b[t] \geq k\frac{\bar{M}[p]-kB(t,p)+1}{N[p]-kB(t,p)+1}$	✓		cond. **1** or **2**, $N[p], k$ in Prop.3.6
(14') $x[t]\bar{S}[t] \geq k\frac{\bar{M}[p]-kB(t,p)+1}{N[p]-kB(t,p)+1}$		✓	$N[p], k$ in Prop.3.6
$\forall t \in T, {}^\bullet t = \{p_1, p_2\}:$			
(15) $x[t]b[t]B(t,p_1) \geq$	✓		cond. **1** or **2** in Prop.3.6,
$\bar{M}[p_1] - B(t,p_1) + 1 - N[p_1]\left(1 - \frac{\bar{M}[p_2]-B(t,p_2)+1}{N[p_2]-B(t,p_2)+1}\right)$			$N[p_1], N[p_2]$ in Prop.3.6
(15') $x[t]\bar{S}[t]B(t,p_1) \geq$	~~✓~~	✓	$N[p_1], N[p_2]$ in Prop.3.6
$\bar{M}[p_1] - B(t,p_1) + 1 - N[p_1]\left(1 - \frac{\bar{M}[p_2]-B(t,p_2)+1}{N[p_2]-B(t,p_2)+1}\right)$			
Routing			$\forall t_j, t_k \in ECS$
(16) $r^i[t_j]x[t_k] \leq r^s[t_k]x[t_j],\quad r^i[t_k]x[t_j] \leq r^s[t_j]x[t_k]$	(TPNF)		well-behaved conflict
(17) $\frac{x[t_j]}{P[t_j]} = \frac{x[t_k]}{P[t_k]}$	(XTPN)		well-behaved conflict, **C1** and **C2** in Prop. 3.8
(17') $\frac{x[t_j]}{r_j} = \frac{x[t_k]}{r_k}$		✓	

TABLE II

LP PROBLEMS FOR THE BOUND COMPUTATION OF TPN AND TIMED/SPNS

The robot is composed of a mobile platform and several sensors that get information from the environment. More precisely, a 3-D laser range-finder with two degrees of freedom is used in navigation tasks with obstacle avoidance. The laser information is used to modify the nominal trajectory while the robot moves near an obstacle. Three main processes are involved in the application:

- the robot-control process, that controls the robot motion and is periodic;
- the laser process, that provides proximity information used by the robot-control process to avoid the obstacles and it is also a periodic process; and
- the supervisory process, that supervises the whole set of robotic tasks to detect if a goal or subgoal in the trajectory is reached, updates the current goal point and manages the system alarms. It is a non periodic process.

In the laser process, the location of sensed points is corrected considering the motion of the robot while the rotating sensor gathers the points.

The system is characterized by the following real-time constraints:

- the control loop has a sample period that is established at the analysis phase. However, due to the time constraints of the robot's internal control system, this sample period must be greater than 180 ms.
- The 3-D laser sends a new scan every 100 ms and the controller must accept and process the sensor data at this rate.

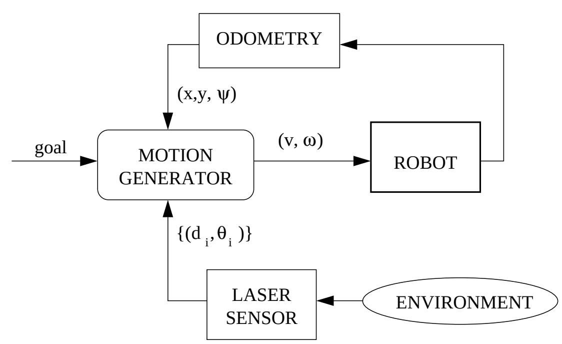

Fig. 3. General control scheme for the navigation

Figure 3 depicts the general control scheme. In each period, the motion generator computes the velocity commands (linear and angular, (v, ω)), taking into account the information provided by odometry (vehicle location consisting of position and orientation, (x, y, ψ)) and by the laser sensor (distance and angle of each ray, (d_i, θ_i)). The laser information is processed to compute the points belonging to the environment and to obstacles, and thus modifying the motion computed in the previous period.

We omit here more detailed robotic aspects of the problem, like motion generation, real-time obstacle avoidance or correction and integration techniques for the sensed points, that can be found in [39].

Montano and Villarroel [39] proposed the use of the TPN formalism for the whole life cycle, from specification to automatic code generation. They built a TPN model of the system that is showed in Figure 4, where the immediate transitions are drawn as thin bars and the timed ones as white bars. To improve readability, broken arcs are used: the name associated to a broken arc is the name of the transition/place from which the arc is originated or to which the arc is addressed.

The system consists of three main processes: 1) the control process, modelled by the subnet in the middle; 2) the laser process, represented by the subnet on the right; and 3) the supervisory process, represented by the subnet on the left.

The three processes run on a multiprocessor without contention on a common CPU. Both the control process and the laser process are characterized by a periodic activation mechanism. In particular, the firing time of transition *Control_Periodic_Activation*, modelling the sample period of the control process, is parameterized (parameter T), while the firing time of transition *LaserPeriod*, modelling the sample period of the laser process, is set to 0.1 $sec.$

The objective of the analysis is to find the lower bound of the robot-control period, i.e., the minimum sample period of the process controlling the robot motion that ensures the deadlines are met. To achieve this objective, [39] adopted the following analysis approach:

1) they transformed the original TPN model into a set of timed Petri Net models (with constant deterministic timing), since they did not have any tool to analyze TPN models.
2) They set the value of the parameter T within a certain interval ($T \geq 0.18$ $sec.$).
3) For each timed Petri Net model, they checked the fulfillment of the timing requirements through the inspection of the state space. In particular, they verified on-the-fly whether: **(a)** the model was one bounded and **(b)** specified pair of places were not simultaneously marked. Indeed, the presence of non binary places in the net of Figure 4 means that there exists in the system an event that cannot be processed at the required rate; thus, the representative tokens are accumulated in a place (as in place $CT2$).

   On the other hand, the simultaneous marking of an activity place (as CA) and the corresponding activation place (as $CT2$) means that the periodical activity has not finished before the next activation starts, that is the deadline has not been met.

The last two steps were iterated by increasing the sample period of the control process of 10 $ms.$ and the minimum value satisfying the timing requirements resulted to be 0.2 $sec.$

Now we come back to the TPN model suggested by [39] for the analysis. To apply the bounding technique proposed in Section III, we use an *ergodic* version of the TPN model of Figure 4, which is obtained by eliminating the possibility of reaching the final goal from the supervisory process and of terminating the periodic processes. This modification guarantees the liveness of the net and it does not affect the metric of interest, that is the completion time of the control process.

We have evaluated the control process completion time in the worst case (T^w_{exec}) by computing the throughput lower bound of transition *ControlEndPeriod* and by taking its inverse (i.e., we applied the Little's operational law [13]). The experiment has been repeated by incrementing the sample period T in the interval $(0.18, \infty)$ $sec.$. until the condition $T^w_{exec} \leq T$ is not satisfied. The min-LP problem has been generated automatically with our bound solver for TPNs (its generation took 7.94 $sec.$). It has as objective function $f(x) = x[ControlEndPeriod]$ and it is characterized by 1247 variables and 1039 constraints, where 931 are equality constraints.

In particular, the constraints generated are reachability, token flow and enabling operational law constraints, i.e., constraints *(9, 10, 11, 12, 13, 15)* summarized in Table II.

Using first the incremental step of 10 $ms.$, as the authors in [39], we have obtained the same results, that is $T = 0.2 sec.$. Then, the experiments have been carried out by using a smaller incremental step (1 ms), then obtaining a better value for $T = 0.193 sec.$. The time required to solve the LP problems is about 0.88 $sec.$. Both the generation and the solution of the LP problems have been performed on a Pentium 4 PC with 1.60GHz CPU.

Observe that the state space of the TPN model of Figure 4 is not bounded when the sample period is lower or equal to $0.192 sec.$ Then, the assessment of the timing requirements by using enumerative techniques has to be carried out, necessarily, by adopting on-the-fly verification methods, as made by [39], that stop the state space generation when the unacceptable conditions **(a)** and **(b)** are detected and avoid the complete construction of the state space.

The time required to solve the LP problems does not depend instead on the value assigned to the sample period and, in general, the bounding technique can be applied also in case of unbounded TPN models.

Nevertheless, the bounds obtained using the techniques presented in this article guarantee the fulfillment of timing requirements under the average operational behavior assumption, while the ones computed by [39] ensure the fulfillment of the timing requirements for all finite observation periods.

VI. CONCLUSION

TPNs are a suitable modelling formalism for quantitative analysis of real time systems. They capture timing constraints, through the specification of minimum and maximum time delays of system activities, still maintaining the non determinism that is intrinsic in the system specification at the early stages.

In this paper, we have presented a structural performance analysis technique for TPNs and their stochastic extensions (i.e., TPNF and XTPN). We have shown that it is possible to compute bounds, under the average operational behavior assumption, by solving proper LP problems. The set of constraints, characterizing such LP problems, are derived from the net structure, the initial marking, and the parameters that

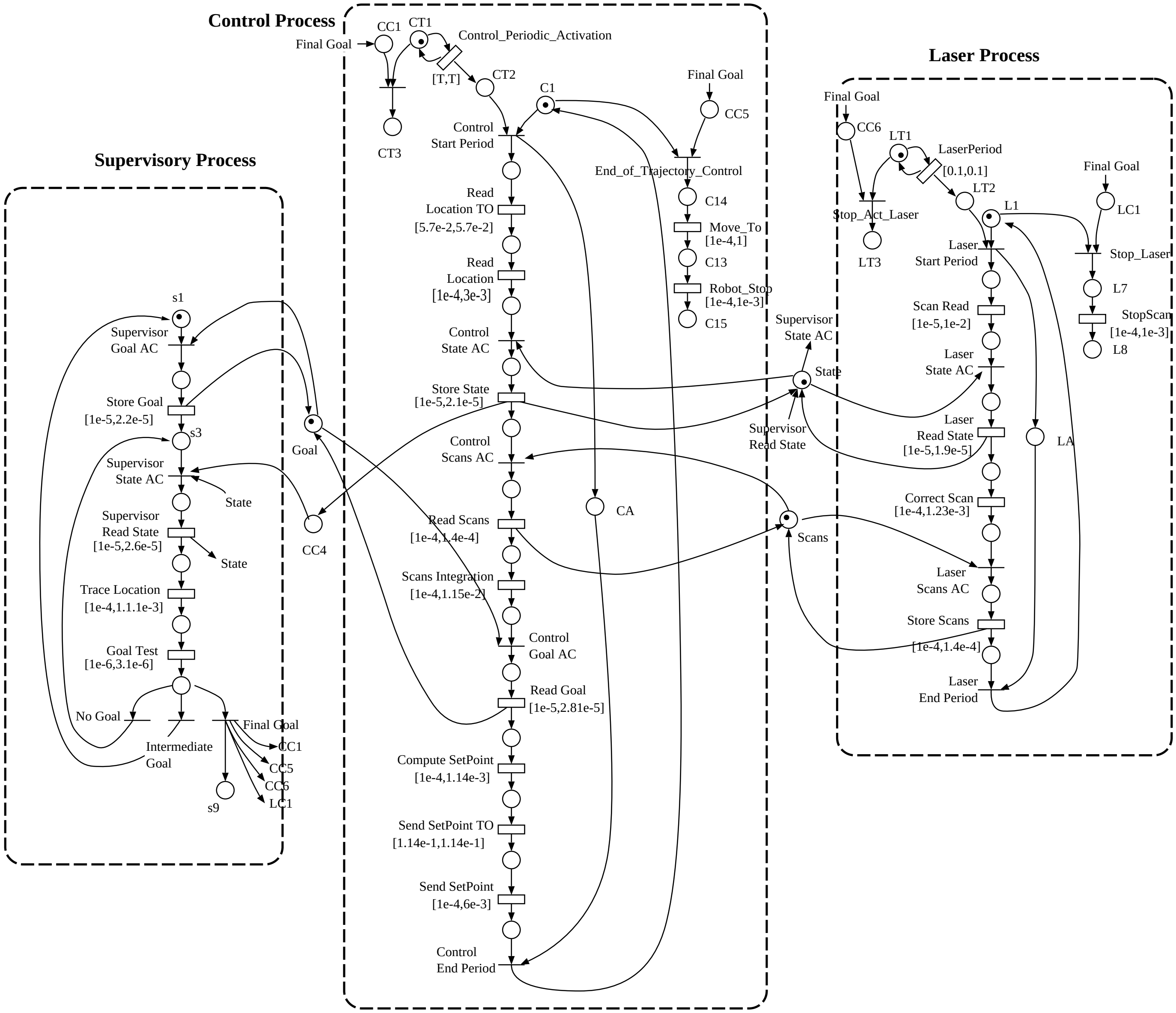

Fig. 4. The TPN model of the robot-control application.

define the time interpretation. Theoretically, the LP problems have a complexity linear in the complexity of the net, that is the number of LP variables and constraints increases linearly with the size (no. of places and transitions) of the net.

The technique presented here is an extension of a previous LP-based bound computation technique developed for Timed PNs/SPNs. The time interpretation considered in this paper is that the firing of transitions is restricted within an interval that defines, per each transition, the earliest and the latest firing time relative to the instant at which it was enabled. A similar interval based definition is possible for the conflict resolution policy at free choice conflicts, that leads to the introduction of the TPNFs. TPNFs have a practical interest, for example, in the modelling of real-time systems like flexible manufacturing systems (FMS). Usually, to model a production plan with Petri Nets it is necessary to establish the proportion of parts that must be produced for each class of parts during a period of time and, in many cases, this is carried out by fixing firing frequencies of transitions that represent the starting of the production of each class of parts. The possibility of modelling the production plan with an interval frequency increases the expression power of the model, making the FMS even more flexible (i.e., it is possible to define the production plan with a “fairness constraint” rather than with fixed ratios). When XTPN are considered, the non determinism of TPN is reduced in a probabilistic manner by adding a stochastic measure for the duration of activities. We can then compute bounds for all the possible model executions, under the average operational behavior assumption, using the TPN approach, and bounds for the average stochastic behavior of the model, using the Timed PN/SPN approach.

We have implemented a performance bound solver for TPNs [17], that has been used for the computation of bounds in the FMS example and in the case study. The solver has been integrated in the *DrawNET* [40] modeling and analysis framework.

With respect to tightness of the computed bounds, a general theoretical result has not been obtained. This was also the case for the computation of bounds for Timed PNs or SPNs already present in the literature. As in the case of (qualitative) structural theory of net systems, the derived performance-oriented results are specially powerful for some well-known subclasses of nets, like strongly connected marked graphs, live and bounded free choice nets, or nets with freely related T-semiflows (like the running example of FMS). For more general classes, the quality of the bounds could be poor, specially for throughput lower bounds. Nevertheless, comments on the usefulness of the bounding techniques given in [27] still remain valid in our context: that is the accumulated experience in applying them to a wide range of systems can provide some guidelines to establish when the techniques give useful bounds.

On the other hand, efficient splitting of the intervals of input parameters, proposed in [30] for interval-based performance analysis, is a promising technique that could be exploited in our context to overcome bound looseness and deserves further investigation. Thus, we think that a first step in the performance oriented structural analysis of TPN models has been achieved in this paper.

Acknowledgment

Authors would like to thank José Luis Villarroel, that helped us in the modelling of the case study. We also thank the anonymous referees and the associate editor, who reviewed the paper and helped to improve it. Finally, our work has been supported by the project DPI2006-15390 of the Spanish Ministry of Science and Innovation, and by the *World Wide Style* researcher mobility program by the University of Torino.

References

[1] L. Thiele and R. Wilhelm, "Design for timing predictability," *Real-time Systems*, vol. 28, no. 2-3, pp. 157–177, November 2004.

[2] T. Murata, "Petri Nets: Properties, Analysis and Applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, April 1989.

[3] G. Chiola, "Timed Petri Nets," in *Performance Models for Discrete Event Systems with Synchronizations: Formalisms and Analysis Techniques*, G. Balbo and M. Silva, Eds. Zaragoza, Spain: Editorial KRONOS, September 1998, vol. 1.

[4] P. Merlin and D. Faber, "Recoverability of communication protocols," *IEEE Trans Commun.*, vol. COM-24, no. 9, 1976.

[5] G. Chiola, M. A. Marsan, G. Balbo, and G. Conte, "Generalized stochastic petri nets: A definition at the net level and its implications," *IEEE Trans. Softw. Eng.*, vol. 19, no. 2, pp. 89–107, 1993.

[6] G. Juanole and Y. Atamna, "Dealing with Arbitrary Time Distributions with the Stochastic Timed Petri Net Model - Application to Queueing Systems," in *Proc. of PNPM*, 1991, pp. 32–41.

[7] J. Roux and G. Juanole, "Functional and Performance Analysis Using Extended Time Petri Nets," in *Proceedings of PNPM*, 1987, pp. 14–23.

[8] B. Berthomieu and M. Diaz, "Modeling and Verification of Time Dependent Systems Using Time Petri Nets," *IEEE Trans. on Soft. Eng.*, vol. 12, no. 3, pp. 259–273, March 1991.

[9] B. Berthomieu and M. Menasche, "An enumerative approach for analysing Time Petri Nets," in *IFIP congress*, Paris, France, 1983.

[10] L. Popova, "On Time Petri Nets." *Journal Inform. Process. Cybern., EIK (formerly: Elektron. Inform. verarb. Kybern.)*, vol. 27, no. 4, pp. 227–244, 1991.

[11] R. H. Sloan and U. A. Buy, "Reduction Rules for Time Petri Nets." *Acta Informatica*, vol. 33, no. 7, pp. 687–706, 1996.

[12] L. Popova-Zeugmann and D. Schlatter, "Analyzing paths in time Petri Nets," *Fundam. Inf.*, vol. 37, no. 3, pp. 311–327, 1999.

[13] P. Denning and J. Buzen, "The Operational Analysis of Queueing Network Models," *ACM Comput. Surv.*, vol. 10, no. 3, pp. 225–261, 1978.

[14] C. Ramchandani, "Analysis of asynchronous concurrent systems by Timed Petri Nets," Ph.D. dissertation, MIT Cambridge, MA, United Kingdom, 1974.

[15] G. Chiola, C. Anglano, J. Campos, J. M. Colom, and M. Silva, "Operational Analysis of Timed Petri Nets and Application to the Computation of Performance Bounds," in *Proc. of the 5^{th} Int. Workshop on Petri Nets and Performance Models*, IEEE-CS, Ed., Toulouse, France, October 1993, pp. 128–137.

[16] S. Bernardi and J. Campos, "On Performance Bounds for Interval Time Petri Nets," in *Proc. of the 1^{st} Int. Conference on Quantitative Evaluation of Systems (QEST'04)*. Enschede, The Netherlands: IEEE Computer Society, September 2004, pp. 50–59.

[17] E. Pacini, S. Bernardi, and M. Gribaudo, "ITPN-PerfBound: A performance bound tool for Interval Time Petri Nets," in *Proc. of TACAS*, ser. LNCS, S. Kowalewski and A. Philippou, Eds., vol. 5505. Springer-Verlag, 2009, pp. 50–53.

[18] E. Vicario, "Static Analysis and Dynamic Steering of Time-Dependent Systems," *IEEE Trans. on Software Engineering*, vol. 27, no. 8, pp. 728–748, August 2001.

[19] J. Wang, Y. Deng, and G. Xu, "Reachability Analysis of Real-Time Systems Using Time Petri Nets," *IEEE Trans. on Systems, Man and Cybernetics – Part B: Cybernetics*, vol. 30, no. 5, pp. 725–736, October 2000.

[20] D. Xu, X. He, and Y. Deng, "Compositional Schedulability Analysis of Real-Time Systems Using Time Petri Nets," *IEEE Trans. on Software Engineering*, vol. 28, no. 10, pp. 984–996, October 2002.

[21] Z. Gu and K. Shin, "Analysis of Event-Driven Real-Time Systems with Time Petri Nets: a Translation-Based Approach," in *IFIP 17^{th} World Computer Congress, Stream on Distributed and Parallel Embedded Systems (DIPES)*, B. Kleinjohann, K. Kim, L. Kleinjohann, and A. Rettberg, Eds. Montreal (Canada): Kluwer, August 2002, pp. 31–40.

[22] R. Mascarenhas, D. Karumuri, U. Buy, and R. Kenyon, "Modeling and Analysis of a Virtual Reality System with Time Petri Nets." in *Proc. of the International Conference of Software Engineering (ICSE)*. Kyoto (Japan): IEEE Computer Society, April 1998, pp. 33–42.

[23] G. Bucci, A. Fedeli, L. Sassoli, and E. Vicario, "Timed State Space Analysis of Real-Time Preemptive Systems," *IEEE Trans. on Software Engineering*, vol. 30, no. 2, pp. 97–111, February 2004.

[24] D. Lime and O. Roux, "A Translation Based Method for the Timed Analysis of Scheduling Extended Time Petri Nets," in *In Proc. of the 25^{th} IEEE International Real-Time Systems Symposium (RTSS04)*. IEEE Computer Society, 2004, pp. 187–196.

[25] G. Bucci, R. Piovosi, S. Sassoli, and E. Vicario, "Introducing probability within state class analysis of dense-time-dependent systems," in *Proc. of the 2^{nd} Int. Conf. on the Quantitative Evaluation of Systems, Sept. 2005*. IEEE Computer Society, 2005, pp. 13–22.

[26] V. Sassoli and E. Vicario, "Close form derivation of state-density functions over DBM domains in the analysis of non-Markovian models," in *Proc. of the 4^{th} Int. Conf. on the Quantitative Evaluation of Systems, Sept. 2007*. IEEE Computer Society, 2007, pp. 59–68.

[27] G. Avrunin, J. Corbett, L. Dillon, and J. Wileden, "Automated derivation of time bounds in uniprocessor concurrent systems," *IEEE Transaction on Software Engineering*, vol. 20, no. 9, pp. 708–719, 1994.

[28] S. Majumdar, "Application of Relational Interval Arithmetic to Computer Performance Analysis: a Survey," *Constraints*, vol. 2, no. 2, pp. 215–235, 1997.

[29] S. Majumdar and R. Ramadoss, "Interval-based performance analysis of computing systems," in *MASCOTS'95, Proc. of the 3^{rd} Int. Workshop on Modeling, Analysis, and Simulation On Computer and Telecommunication Systems, January, Durham, North Carolina, USA*, P. W. Dowd and E. Gelenbe, Eds. IEEE Computer Society, 1995, pp. 345–351.

[30] J. Lüthi and C. Lladó, "Splitting techniques for interval parameters and their application to performance models," *Perform. Eval.*, vol. 51, no. 1, pp. 47–74, 2003.

[31] M. A. Marsan and et al., "The Effect of Execution Policies on the Semantics and Analysis of Stochastic Petri Nets." *IEEE Trans. on Soft. Eng.*, vol. 15, no. 7, pp. 832–846, 1989.

[32] B. Berthomieu, "La méthode des Classes d'États pour l'Analyse des Réseaux Temporels - Mise en Oeuvre, Extension à la multi-sensibilisation, Modélisation des Systèmes Réactifs," in *Proc. of MSR'2001*. Toulouse, France: Hermes Sciences, 2001, pp. 254–263.

[33] H. Ohl, "Foncionnements repetitifs de systemes flexibles de production manufacturiere: analyse et optimisation des performances a l'aide de reseaux de Petri," Ph.D. dissertation, L'Université des sciences

et technologies de Lille, Laboratoire d'Automatique et Informatique industrielle de Lille - Ecole Centrale de Lille, Septembre 1995.

[34] L. Recalde, E. Teruel, and M. Silva, "On Linear Algebraic Techniques for Liveness Analysis of PT Systems," *Journal of Circuits Systems and Computers*, vol. 8, no. 1, pp. 223–265, 1998.

[35] N. Bengston, "Measuring Errors in Operational Analysis Assumptions," *IEEE Trans. on Software Engineering*, vol. 13, no. 7, pp. 767–776, 1987.

[36] S. Bernardi and G. Balbo, "Concurrent Generalized Petri Nets: Regenerative Conditions," in *Proc. of the* 9^{th} *International Conference on Petri Nets and Performance Models (PNPM'01).* Aachen, Germany: IEEE Computer Society, September 2001, pp. 125–134.

[37] Z. Liu, "Performance Analysis of Stochastic Timed Petri Nets Using Linear Programming Approach," *IEEE Trans. on Soft. Eng.*, vol. 24, no. 11, November 1998.

[38] N. Karmarkar, "A new polynomial time algorithm for linear programming," *Combinatorica*, vol. 4, pp. 373–395, 1984.

[39] L. Montano, F. García, and J. Villarroel, "Using the Time Petri Net formalism for specification, validation, and code generation in robot-control applications," *International Journal of Robotics Research*, vol. 19, no. 1, pp. 59–76, January 2000.

[40] G. Gribaudo, D. Codetta Raiteri, and F. Franceschinis, "Draw-Net, a customizable multi-formalism, multi-solution tool for the quantitative evaluation of systems," in *Proc. of the* $2nd$ *Int. Conference on the Quantitative Evaluaiton of Systems (QEST 2005), September 2005*. IEEE Computer Society, 2005, pp. 257–258.

Simona Bernardi received the MS degree in mathematics and the PhD degree in computer science, in 1997 and 2003, respectively, both from the University of Torino (Italy). Since 2005, she has held a researcher position at the University of Torino. She has been vising researcher at the Department of Computer Science and System Engineering of the University of Zaragoza, Spain, and at the Department of System and Computer Engineering of the Carleton University, Canada. Her research interests include software performance and dependability engineering, UML and object-oriented software development methodologies, formal methods for the modelling and analysis of software systems. She has served as a referee for international journals and as a program committee member for several international conferences and workshops.

Javier Campos was born in Jaca (Spain) in 1963. He received the MS degree in applied mathematics and the Ph.D. degree in systems engineering and computer science from the University of Zaragoza, Spain, in 1986 and 1990, respectively. In 1986 he joined the Department of Electrical Engineering and Computer Science of the University of Zaragoza. In 2005, Dr. Campos was named Full Professor in Languages and Computing Systems at the Department of Computer Science and Systems Engineering of the University of Zaragoza. He was the Director of the Department from 2001 to 2003. His research interests include modelling and performance evaluation of distributed and concurrent systems, Petri nets, software performance engineering and discrete event systems in automation. He has supervised the completion of ninety Master Thesis students and two PhD students. Since 1989 he co-authored about eighty papers published in refereed journals and conferences. He has been plenary session invited speaker and tutorial invited speaker in several important meetings. He has served in the Program Committee, sometimes as a Chair, of several international conferences. Dr. Campos is a Founding Member of the Aragón Institute for Engineering Research. He is member of the IEEE IES Technical Committee on Factory Automation and Associate Editor of IEEE Transactions on Industrial Informatics.