20

ISPN: Modeling Stochastic with Input Uncertainties Using an Interval-Based Approach

Sérgio Galdino and Paulo Maciel
Polytechnic School of Pernambuco University
& Federal University of Pernambuco
Brazil

1. Introduction

Analytic techniques are frequently used for performance analysis of discrete event systems. Conventional models have a set of single value input parameters (such as mean resource demands) and give single value results for each performance index of interest (such as mean system throughput). However, this single point characterization of parameters is insufficient when uncertainties and variabilities are related with system parameters. As an application domain, we may highlight software performance engineering, which accomplishes performance modelling in several phases design-cycle and throughout implementation (Girault & Valk, 2003; Smith, 1990). Even if uncertainties and variabilities may be associated with one or more parameters of the system in early stages of system design, the expert designer might have a suitable guess related to the interval of values associated with these parameters due to previous experience.
The current availability of software tools for performance evaluation allows one to hide the technicalities from the end-user. Users specify their performance model using some high-level modeling language supported by tools such as PEPSY-QNS, TimeNET 4.0, SPNP 6.0, GreatSPN 2.0, or PEPA, in which the underlying mathematical model is automatically generated and analyzed. We propose the adoption of intervals to represent the uncertainties in the parameters of ISPN (Interval Stochastic Petri Net) models (Galdino & Maciel, 2006; Galdino et al., 2007a;b). Therefore, the set of methods considered for Markov chain steady-state analysis have to be adapted for taking into account interval arithmetic. In ISPN the exponential transition rates and immediate transition weights are represented by intervals. This chapter focuses on ISPN using MATLAB with INTLAB toolbox. We briefly introduce the interval arithmetic. Afterwards, we describe the ISPN and outline the approach adopted to the respective interval steady state analysis. We present two ISPN system models and the respective results of analysis. Further possibilities of the method are also suggested.

2. Background

Prior to present the ISPN, this section introduces some basic concepts needed to understand how interval arithmetics may be used for evaluating system's metrics. Hence, we initially introduce some concepts on interval arithmetics.

2.1 Notation

Throughout this paper, all scalar variables are denoted by ordinary lowercase letters (a). Interval variables are enclosed in square brackets ([a]). Underscores and overscores denote lower and upper bounds, respectively.
A real interval $[x]$ is a nonempty set of real numbers

$$[x] = [\underline{x}, \overline{x}] = \{\tilde{x} \in \mathbb{R} : \underline{x} \leq \tilde{x} \leq \overline{x}\} \qquad (1)$$

where $\underline{x}$ and $\overline{x}$ are called the *infimum (inf)* and *supremum (sup)*, respectively, and $\tilde{x}$ is a point value belonging to an interval variable $[x]$.
The set of all intervals $\mathbb{R}$ is denoted by $I(\mathbb{R})$ where

$$I(\mathbb{R}) = \{[\underline{x}, \overline{x}] : \underline{x}, \overline{x} \in \mathbb{R} : \underline{x} \leq \overline{x}\} \qquad (2)$$

2.2 Interval arithmetic

Let $+$, $-$, $\times$, and $\div$ denote the arithmetic operations. If $\bullet$ denotes any of these arithmetic operation for real numbers x and y, then the corresponding operation for arithmetic on interval numbers $[x]$ and $[y]$ is

$$[x] \bullet [y] = \{x \bullet y | x \in [x], y \in [y]\}.$$

Thus the interval $[x] \bullet [y]$ resulting from the operations contain every possible number that can be found as $x \bullet y$ for each $x \in [x]$, and each $y \in [y]$.
A form of interval arithmetic perhaps first appeared in 1924 and 1931 in (Chen & van Emden, 1995; Young, 1931), then later in (Ullrich, 1990). Modern development of interval arithmetic began with R. E. Moore's dissertation (Moore, 1962) as a method for determining absolute errors of an algorithm, considering all data errors and rounding, after R.E. Moore introduced interval analysis (Moore, 1966). Interval arithmetic is an arithmetic defined on sets of intervals, rather than sets of real numbers. The power of interval arithmetic lies in its implementation on computers. In particular, outwardly rounded computations allows rigorous enclosures.
An important result is the inclusion property theorem . Rall aptly calls this the fundamental theorem of interval analysis (Hansen & Walster, 2004; Rall, 1969).

Theorem 2.1 (Fundamental Theorem). *If the function $f([x]_1, [x]_2, [x]_3, \cdots [x]_n)$ is an expression with a finite number of intervals $[x]_1, [x]_2, [x]_3, \cdots [x]_n \in I(\mathbb{R})$ and interval operations $(+, -, \times, \div)$, and if $[w]_1 \subseteq [x]_1, [w]_2 \subseteq [x]_2, [w]_3 \subseteq [x]_3, \cdots, [w]_n \subseteq [x]_n$ then $f([w]_1, [w]_2, [w]_3, \cdots, [w]_n) \subseteq f([x]_1, [x]_2, [x]_3, \cdots, [x]_n)$.*

2.3 Enclosures for the range of function

Consider a function f from $\mathbb{R}^n$ to $\mathbb{R}$. The interval function [f] from $I(\mathbb{R}^n)$ to $I(\mathbb{R})$ is an inclusion function [f] if

$$\forall [\mathbf{x}] \in I(\mathbb{R}^n),\ f([\mathbf{x}]) \subset [f]([\mathbf{x}])$$

Interval analyses provides, for a large class of functions f, inclusion functions such that $[f]([\mathbf{x}]$ is not too large.
An inclusion function for $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is obtained with two optimizations to compute the $infimum$ anad $supremum$ of f. However, these optimization problems are far from trivial in general. An more tractable approach uses the following theorem (Jaulin et al., 2001), which is a direct consequence of Theorem 2.1.

Theorem 2.2 (Natural Inclusion Function Theorem)**.** *Consider a function*

$$f : \mathbb{R}^n \rightarrow \mathbb{R}, (x_1, x_2, x_3, \cdots x_n) \mapsto f(x_1, x_2, x_3, \cdots x_n),$$

expressed as a finite composition of the operators $+$, $-$, $\times$, *and* $\div$ *and elementary functions (sin, cos, exp, sqr, ...). An inclusion monotonic and thin (degenerated) inclusion function* $[f] : I(\mathbb{R})^n \rightarrow I(\mathbb{R})$ *for* f *is obtained by replacing each real variable* x_i *by an interval variable* $[x_i]$ *and each operator or function by its interval counterpart. This function is called the natural inclusion function of* f*. If* f *involves only continuous operators and continuous elementary functions, the [f] is convergent.If, moreover, each of the variables* $(x_1, x_2, x_3, \cdots x_n)$ *occurs at most once in the formal expression of f then* $[f]$ *is minimal.*

Natural inclusion functions are not minimal in general, because of the dependence problem. The accuracy of the resulting interval strongly depends on the expression of f. The use of natural inclusion functions is not always the best choice. An important field of interval analysis is the use other types of inclusion functions that would provide best enclosures.

3. ISPN formalism

Petri Nets (PNs) represent a family of forms of graphical representation for description of systems whose dynamics are characterized by concurrency, synchronization, mutual exclusion, and conflict, which are typical features of distributed environments (Murata, 1989). PNs incorporate a notion of local state and a rule for state change (transition firing) that allow them to capture both the static and the dynamic characteristics of a real system being explicit about time considerations. The introduction of timing concepts into PN models were proposed later by Ramchandani (Ramchandani, 1974), Merlin (Merlin & Farber, 1976), and Sifakis (Sifakis, 1978) from distinct viewpoints. Molloy (Molloy, 1981) as well as Florin (Florin & Natkin, 1989) proposed PN models in which stochastic timing was considered. These works opened the possibility of connecting PN theory and stochastic modeling. Nowadays, these models as well as their extensions are generically named Stochastic Petri nets (SPN).
ISPN is an extension of GSPN (*Generalized Stochastic Petri Nets*) model in order to introduce the interval analysis (Galdino & Maciel, 2006) and has a lot of potential to practical applications. As part of the contribution in the development of this work, the authors also brought out its related practical applications in performance and dependability evaluation (Galdino & Maciel, 2006; Galdino et al., 2007a;b). GSPN is a particular timed PN (**Petri Net**) that incorporates both stochastic timed transitions (represented as white boxes) and immediate transitions (represented as thin black bars). Timed transitions have an exponentially distributed firing time and immediate transitions fire in zero time. GSPNs were originally defined (Marsan, Balbo & Conte, 1984) and later modified as described in (Brinksma, 2001). A formal definition of ISPN is provided below. This definition keeps to the SPN definition presented in (German, 2000), but considers real intervals assigned to transition delays and weights instead of adopting real single values.
Let $ISPN = (P, T, I, O, \Pi, G, M_0, Atts)$ be an interval stochastic Petri net, where

- $P = \{p_1, p_2, \cdots p_n\}$ is the set of places,
- $T = \{t_1, t_2, \cdots t_m\}$ is the set of transitions,
- $I \in (\mathbb{N}^n \rightarrow \mathbb{N})^{n \times m}$ is a matrix of marking-dependent multiplicities of input arcs, where the i_{jk} entry of I gives the possibly marking-dependent arc multiplicity of input arcs from place p_j to transition t_k $[A \subset (P \times T) \cup (T \times P) - set\ of\ arcs]$,

- $O \in (\mathbb{N}^n \to \mathbb{N})^{n \times m}$ is a matrix of marking-dependent multiplicities of output arcs, where o_{jk} entry of O gives the possibly marking-dependent arc multiplicity of output arcs from transition t_j to place p_k,
- $H \in (\mathbb{N}^n \to \mathbb{N})^{n \times m}$ is a matrix of marking-dependent multiplicities of inhibitor arcs, where h_{jk} entry of H gives the possibly marking-dependent arc multiplicity of inhibitor arcs from place p_j to transition t_k,
- $G \in (N^n \to \{true, false\})^m$ is a vector that assigns a guard condition related to place markings to each transition,
- $M_0 \in \mathbb{N}^n$ is a vector that assigns the initial marking of each place (initial state),
- $Atts = (Dist, W, Markdep, Police, Concurrency)^m$ comprises the set of attributes for transitions, where
  - $[Dist] \in \mathbb{N}^m \to [\mathcal{F}]$ is a possible marking-dependent firing interval distribution function (the domain of $[\mathcal{F}]$ *is* $[0, \infty)$),
  - $[W] \in \mathbb{N}^m \to I(\mathbb{R}^+)$ is a possible marking-dependent interval weight,
  - $Markdep \in \{constant, enabdep\}$ where the transition interval firing timing distribution could be marking independent ($constant$) or enabling dependent ($enabdep$ - the distribution depends on the actual enabling condition),
  - $Police \in \{prd, prs\}$ is the preemption policy (*prd- preemptive repeat different* means that when a preempted transition becomes enabled again the previous elapsed firing time is lost; *prs- preemptive resume*, in which the firing time related to a preempted transition is resumed when the transition becomes enabled again),
  - $Concurrency \in \{ss, is\}$ is the degree of concurrency of transitions, where *ss* represents single server semantics and *is* depicts infinite-server semantics.

It is worth noting that if only point intervals are assigned to delays and immediate transition weights the model is reduced to a GSPN. This case the ISPN analysis provides GSPN verified results.

3.1 ISPN steady state analysis

ISPN is considered to be a high-level formalism for ICTMC (Interval Continuous Time Markov Chain) generation Galdino et al. (2007b). The classical algorithms found in literature (Bolch et al., 2006) are adapted to take into account the interval coefficients of the ISPN model. This is a contribution to analysis of models ISPN that will be presented in sequel. The analysis of models using ISPN is accomplished in four subtasks:

- generation of the IERG (Interval Extended Reachability Graph),
- elimination of vanishing markings and the corresponding state transitions,
- interval steady-state analysis,
- computation of measures. Standard measures such as the average number of tokens in places and the throughput of timed transitions are computed using interval arithmetic.

3.1.1 Generation of the IERG

The first of four steps of ISPN analysis is the IERG generation (*interval extended reachability graph*). From the IERG the set of markings $\mathcal{M} = \mathcal{T} \cup \mathcal{V}$ is divided into set of tangible markings $\mathcal{T}$ and vanishing $\mathcal{V}$. Through the elimination of vanishing markings discussed below, using methods of interval analysis, we obtain the infinitesimal generator matrix $[Q]$ of ICTMC underlying an ISPN model.

From a given ISPN, an interval extended reachability graph (IERG) is generated containing markings as nodes and interval stochastic information attached to arcs so as to relate markings to each other. The ISPN reachability graph is a directed graph $RG(ISPN) = (V, E)$, where $V = RS(ISPN)$ and $E = \left\{\langle \mathbf{m}, t, \mathbf{m}' \rangle \mid \mathbf{m}, \mathbf{m}' \in RS(ISPN) \;\; and\; \mathbf{m} \overset{t}{\rightarrow} \mathbf{m}'\right\}$ are the set of nodes and edges, respectively. If an ISPN model is bounded, the $RG(ISPN)$ is finite and it can be constructed, for example, based on Algorithm 5.1: *Computation of the Reachability Graph* p. 61 from (Girault & Valk, 2003).

The $RG(ISPN)$ is constructed, in this work, using the **Algorithm 1** below. The activity defined in Step 2.1 ensures that no marking is visited more than once. Each visited marking is labeled (Step 2.1), and Step 2.2.3 ensures that only unique added markings to V are those that were not previously added. When the marking is visited, only those edges that represents the firing of an enabled transition are added to the set E (Step 2.2.4).

```
===================================================================
Algorithm 1

(** IERG generation **)

Input - A ISPN model.
Output - A directed graph $RG(ISPN) = (V,E)$ of a limited network system.

1.  Initialize $RG(ISPN) = (\{\mathbf{m}_0\}, \emptyset)$;$\mathbf{m}_0$ is unlabelled.
2.  while there are an unlabeled node m in $V$ do
  2.1 Select an unlabeled node $\mathbf{m} \in V$ label it
  2.2 for each enabled transition t in m do
    2.2.1 Calculate $\mathbf{m}'$ such that $\mathbf{m} \overset{t}{\rightarrow} \mathbf{m}'$;
    2.2.2 if there are $\mathbf{m}'' \in V$ such that $\mathbf{m}'' \overset{\sigma}{\rightarrow} \mathbf{m}'$ and $\mathbf{m}'' \leq \mathbf{m}'$
          then the algorithm fails and ends;
               (no limitation condition was detected).
    2.2.3 if there is no $\mathbf{m}'' \in V$ such that $\mathbf{m}'' = \mathbf{m}'$
          then $V := V \cup \{\mathbf{m}'\}$; ($\mathbf{m}'$ é um nó não etiquetado).
     2.2.4 $E := E \cup \{\langle \mathbf{m}; t; \mathbf{m}' \rangle\}$
3.  The algorithm is successful and RG(ISPN) is the interval extended
reachability graph.

============================================================================
```

3.1.2 Elimination of vanishing markings

The second of four steps of ISPN analysis is the elimination of vanishing markings, which is the step for generating the ICTMC from a given ISPN. Once the IERG has been generated, it is transformed into an ICTMC by the use of matrix algorithms Bolch et al. (2006).

The markings set $\mathcal{M} = \mathcal{V} \cup \mathcal{T}$ in the reachability set of an ISPN is partitioned into two sets, the vanishing markings $\mathcal{V}$ and the tangible markings $\mathcal{T}$. Let:

$$[\mathbf{P}]^{\mathcal{V}} = [\mathbf{P}]^{\mathcal{V}\mathcal{V}} \mid [\mathbf{P}]^{\mathcal{V}\mathcal{T}} \tag{3}$$

denote an interval matrix, where

- $[\mathbf{P}]^{\mathcal{VV}}$ - denotes the interval transition probabilities between vanishing markings,
- $[\mathbf{P}]^{\mathcal{VT}}$ - denotes the interval transition probabilities from vanishing markings to the tangible markings.

Furthermore, let

$$[\mathbf{U}]^{\mathcal{T}} = [\mathbf{U}]^{\mathcal{TV}} \mid [\mathbf{U}]^{\mathcal{TT}} \tag{4}$$

denote an interval matrix, where

- $[\mathbf{U}]^{\mathcal{TV}}$- represents interval transition rates from tangible to vanishing markings;
- $[\mathbf{U}]^{\mathcal{TT}}$- represents interval transition rates between tangible markings.

Now, we obtain the interval rate matrix $[\mathbf{U}]$. This matrix has dimensions $|\mathcal{T}| \times |\mathcal{T}|$, where $\mathcal{T}$ denotes the set of tangible markings.

$$[\mathbf{U}] = [\mathbf{U}]^{\mathcal{TT}} + [\mathbf{U}]^{\mathcal{TV}}(1 - [\mathbf{P}]^{\mathcal{VV}})^{-1}[\mathbf{P}]^{\mathcal{VT}} \tag{5}$$

The interval matrix of the infinitesimal generator is $[\mathbf{Q}] = [q]_{ij}$, where its entries are given by:

$$[q]_{ij} = \begin{cases} [u]_{ij} \;\; if \; i \neq j \\ -\sum_{\substack{k \in \mathcal{T} \\ k \neq i}} [u]_{ik} \;\; if \; i = j \end{cases} \tag{6}$$

where $\mathcal{T}$ denotes the set of tangible markings.

3.1.3 Steady-state probability vector evaluation

Now we describe the third of four steps of ISPN analysis. The steady-state solution of the ICTMC model underlying the ISPN is obtained by solving the interval linear system of equations with as many equations as the number of tangible markings.

$$\begin{cases} [\pi] \cdot [\mathbf{Q}] = \mathbf{0} \\ \sum_{M \in \mathcal{T}} [\pi]\,(M) = 1 \end{cases} \tag{7}$$

$[\pi]$ is the interval vector for the equilibrium pmf (probability mass function) over the reachable tangible markings, and we write $[\pi]\,(M)$ for the interval steady-state probability of a given tangible marking M.

Once the interval generator matrix $[\mathbf{Q}]$ of the ICTMC associated with a ISPN model has been derived, the steady state probability is calculated so that other respective metrics might be subsequently computed.

ISPN models deal with system uncertainties by considering intervals for representing time as well as weights assigned to transition models. The proposed model and the respective methods, adapted to take interval arithmetic into account, allow the influence of simultaneous parameters and variabilities on the computation of metrics to be considered, thereby providing rigorously bounded metric ranges. It is also important to stress that even when only taking into account thin intervals, one may make use of the proposed model, since rounding and truncation errors are naturally dealt with in interval arithmetic, so that the metrics results obtained are certain to belong to the intervals computed.

3.1.4 Interval performance indices

The computation of performance indices (metrics) of interest is the fourth and final step in the analysis ISPN. In the case of ISPN steady state analysis, where interval p.m.f. has already been obtained, indices are calculated by interval function evaluation. Interval performance indices are interval functions extended on classical indices (Marsan, Bobbio, Conte & Cumani, 1984).

4. Examples of ISPN models

The purpose of this section is to present clearly all steps of ISPN analysis. Two examples are used. One is very simple and can be followed up and have calculations performed without using a computer. The second case, however, you must use a software with an interval arithmetic library as a tool to carry out by all his calculations. Example 1 has only two tangible markings and two vanishing markings. Example 2 has sixteen tangible markings and twelve vanishing markings. The performance evaluations are carried out in MATLAB with the INTLAB toolbox (MATLAB *toolbox* INTLAB framework). The ISPN model analysis considering only degenerated intervals (points) leads to the classic model GSPN, with verified computations (self-validating).

4.1 Example 1: ISPN analysis of a single machine

The model depicted in Figure 1 represents a failure prone machine and finite capacity buffer (Desrochers & Al-Jaar, 1994). Table 1 presents (degenerated) interval rates of timed transition firing per unit time, where $[\nu]$ represents the production rate interval, $[\lambda]$ represents the failure rate interval, and $[\mu]$ represents the repair rate interval. Here we have a model equivalent to the GSPN model, because there are only degenerate interval parameters.

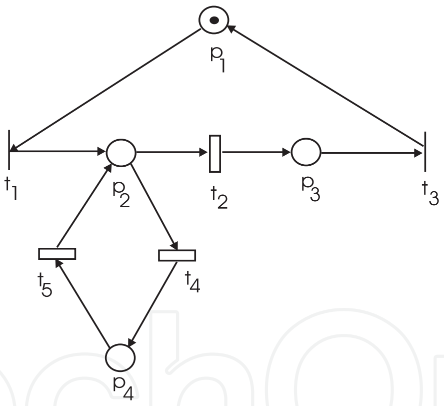

Fig. 1. The Single Machine module.

Transition	Value $([t]^{-1})$	Symbol
$[t_2]$	[10, 10]	$[\nu]$
$[t_4]$	[3, 3]	$[\mu]$
$[t_5]$	[5, 5]	$[\lambda]$

Table 1. Transition Firing Rates (degenerated intervals) for the Single Machine One-Buffer Transfer Line.

As a result of the first step of ISPN analysis we obtain the reachability set (Table 2), and the reachability graph (Figure 2).

State	Marking (m_1, m_2, m_3, m_4)
1	$M_0 = (1,\ 0,\ 0,\ 0\ )$
2	$M_1 = (0,\ 1,\ 0,\ 0\ )$
3	$M_2 = (0,\ 0,\ 1,\ 0\ )$
4	$M_3 = (0,\ 0,\ 0,\ 1\ )$

Table 2. Reachability set and distribution markings from ISPN of Figure 1.

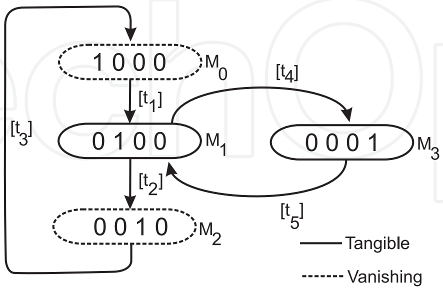

Fig. 2. Reachability graph and interval embedded Markov chain

Finally, we obtain the matrices $[\mathbf{P}]^{\mathcal{VV}}$, $[\mathbf{P}]^{\mathcal{VT}}$, $[\mathbf{U}]^{\mathcal{TV}}$ and $[\mathbf{U}]^{\mathcal{TT}}$:

$$[\mathbf{P}]^{\mathcal{VV}} = \begin{pmatrix} [\,0,\ 0] & [\,0,\ 0] \\ [\,1,\ 1] & [\,0,\ 0] \end{pmatrix} \quad [\mathbf{P}]^{\mathcal{VT}} = \begin{pmatrix} [\,1,\ 1] & [\,0,\ 0] \\ [\,0,\ 0] & [\,0,\ 0] \end{pmatrix}$$

$$[\mathbf{U}]^{\mathcal{TT}} = \begin{pmatrix} [\,0,\ 0] & [\,5,\ 5] \\ [\,3,\ 3] & [\,0,\ 0] \end{pmatrix} \quad [\mathbf{U}]^{\mathcal{TV}} = \begin{pmatrix} [\,0,\ 0] & [\,10,\ 10] \\ [\,0,\ 0] & [\,0,\ 0] \end{pmatrix}.$$

Afterwards, carry out the elimination of vanishing markings (Equation 5) to obtain the matrix of rate intervals $[U]$. The matrix of rate intervals represents an IREMC (*Interval Reduced Embedded Markov Chain* on Figure 3):

$$[\mathbf{U}] = \begin{pmatrix} [\,10,\ 10] & [\,5,\ 5] \\ [\,3,\ 3] & [\,0,\ 0] \end{pmatrix}.$$

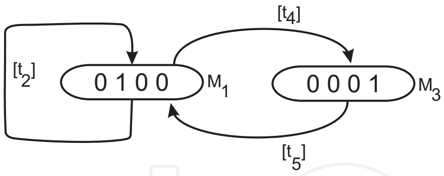

Fig. 3. Interval Reduced Embedded Markov Chain

Finally, using Equation 6, we find the infinitesimal generator interval matrix:

$$[\mathbf{Q}] = \begin{pmatrix} [\,\text{-}5,\ \text{-}5] & [\,5,\ 5] \\ [\,3,\ 3] & [\,\text{-}3,\ \text{-}3] \end{pmatrix}.$$

The third step of ISPN analysis solves the system of interval linear equations described by Equation (7). The interval linear equations solution is carried out by the verifylss function of the MATLAB *toolbox* INTLAB. Substituting the last equation of system $([\pi]_1, [\pi]_2) \cdot [\mathbf{Q}] = 0$ by the normalization condition $[\pi]_1 + [\pi]_2 = 1$, the linear system $([\pi]_1, [\pi]_2) \cdot [\mathbf{A}] = [\mathbf{b}]$ is obtained. The solution of this system directly provides the steady state probabilities of tangible states. Considering

$$[\mathbf{A}] = \begin{pmatrix} -3 & 5 \\ 1 & 1 \end{pmatrix} \; and \; [\mathbf{b}] = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

the M-file MATLAB toolbox INTLAB **case1v.m**, used for calculating verified probabilities and machine production rate, is given bellow:

```
%       INPUT:     A coeffifiente matrix
%                  b right hand side vector
%       OUTPUT:    x interval probabilities vector solution
%                  P machine production rate
format long
intvalinit('displayinfsup')
A=intval([-3,5;1,1])
b=[0;1]
x=verifylss(A,b)
P=10*x(1)
```

Executing **case1v.m** yields:

```
>> case1v
===> Default display of intervals by infimum/supremum (e.g. [ 3.14 , 3.15 ])
intval A =
[ -3.00000000000000, -3.00000000000000] [ 5.00000000000000, 5.00000000000000]
[  1.00000000000000,  1.00000000000000] [ 1.00000000000000, 1.00000000000000]
b =
     0
     1
intval x =
[   0.62499999999998,   0.62500000000001]
[   0.37499999999999,   0.37500000000001]
intval P =
[   6.24999999999998,   6.25000000000001]
>>
```

The verified interval bounds of each state probabilities on tangible states are:

$$[\pi]_{(1)} = [\, 0.62499999999998,\ 6.25000000000001]$$

and

$$[\pi]_{(2)} = [\, 0.37499999999999,\ 0.37500000000001].$$

Finally we can make the fourth and final step of analysis ISPN, computation of metrics. The machine production rate is

$$[P] = [\, 6.24999999999998,\ 6.25000000000001]$$

(calculated with the formula $[P] = [\pi]_{(1)} \cdot [t_2]$). This results exhibit the enclosure of exact value obtained by GSPN analysis. The ISPN analysis results give us verified results, ensuring that the exact value is certain to belong to the intervals computed. One can, for example, to compare this result with interval $P = 6.25$ exact value in this simple case.

Introducing parameters with input uncertainties

Now we calculate a solution in which the parameters are not known exactly, but it is known that they are within certain intervals. Lets consider that rates are $[\mu] = 3 \pm 0.01 = [2.99,\ 3.01]$ and $[\lambda] = 5 \pm 0.01 = [4.99,\ 5.01]$ intervals.

As a result from the first step of analysis (by-product of the reachability set), we obtain the matrices $[\mathbf{P}]^{\mathcal{VV}}$, $[\mathbf{P}]^{\mathcal{VT}}$, $[\mathbf{U}]^{\mathcal{TV}}$ e $[\mathbf{U}]^{\mathcal{TT}}$:

$$[\mathbf{P}]^{\mathcal{VV}} = \begin{pmatrix} [\,0,\ 0] & [\,0,\ 0] \\ [\,1,\ 1] & [\,0,\ 0] \end{pmatrix} \qquad [\mathbf{P}]^{\mathcal{VT}} = \begin{pmatrix} [\,1,\ 1] & [\,0,\ 0] \\ [\,0,\ 0] & [\,0,\ 0] \end{pmatrix}$$

$$[\mathbf{U}]^{\mathcal{TT}} = \begin{pmatrix} [\,0,\ 0] & [\,4.99,\ 5.01] \\ [\,2.99,\ 3.01] & [\,0,\ 0] \end{pmatrix} \; [\mathbf{U}]^{\mathcal{TV}} = \begin{pmatrix} [\,0,\ 0] & [\,10,\ 10] \\ [\,0,\ 0] & [\,0,\ 0] \end{pmatrix}.$$

Afterwards, carry out the elimination of vanishing markings (Equation 5), to obtain the matrix of rate intervals $[U]$:

$$[\mathbf{U}] = \begin{pmatrix} [\,10,\ 10] & [\,4.99,\ 5.01] \\ [\,2.99,\ 3.01] & [\,0,\ 0] \end{pmatrix}.$$

Finally, using Equation 6, we find the infinitesimal generator interval matrix:

$$[\mathbf{Q}] = \begin{pmatrix} [\,\text{-}5.01,\ \text{-}4.99] & [\,4.99,\ 5.01] \\ [\,2.99,\ 3.01] & [\,\text{-}3.01,\ \text{-}2.99] \end{pmatrix}.$$

Considering

$$[\mathbf{A}] = \begin{pmatrix} -3 & 5 \\ 1 & 1 \end{pmatrix} \; and \; [\mathbf{b}] = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

the M-file MATLAB toolbox INTLAB **case1i.m**, used for calculating verified probabilities and machine production rate, is given bellow:

```
%       INPUT:     A coeffifiente matrix
%                  b right hand side vector
%       OUTPUT:    x interval probabilities vector solution
%                  P machine production rate
format long
intvalinit('displayinfsup')
A=infsup([-3.01,4.99;1,1],[-2.99,5.01;1,1])
b=[0;1]
x=verifylss(A,b)
P=10*x(1)
```

Executing **case1i.m** yields:

```
>> case1i
===> Default display of intervals by infimum/supremum (e.g. [ 3.14 , 3.15 ])
intval A =
[ -3.01000000000000, -2.99000000000000] [ 4.99000000000000, 5.01000000000000]
[  1.00000000000000,  1.00000000000000] [ 1.00000000000000, 1.00000000000000]
b =
     0
     1
intval x =
[   0.62374656249999,   0.62625343750001]
[   0.37374656249998,   0.37625343750001]
intval P =
[   6.23746562499999,   6.26253437500001]
>>
```

The interval bounds of each state probabilities on tangible states are:

$$[\pi]_{(1)} = [\,0.62374656249999,\ 0.62625343750001]$$

and

$$[\pi]_{(2)} = [\, 0.37374656249998,\ 0.37625343750001].$$

Finally we can make the computation of machine production rate:

$$[P] = [\, 6.23746562499999,\ 6.26253437500001]$$

(calculated with the formula $[P] = [\pi]_{(1)} \cdot [t_2]$). This result represents the variabilities when the rates are in $[\mu] = [2.99,\ 3.01]$ and $[\lambda] = [4.99,\ 5.01]$ intervals.

4.2 Example 2: ISPN analysis of Two-Machine One-Buffer Transfer Line Model

Consider the Two-Machine One-Buffer Transfer Line Model in Figure 4 (Desrochers & Al-Jaar, 1994). Table 3 presents (degenerated) interval rates of timed transition firing per unit time, where $[\nu_i]$ represents the production rate intervals, $[\lambda_i]$ represents the failure rate intervals, and $[\mu_i]$ represents the repair rate intervals. Here we have a model equivalent to the GSPN model, because there are only degenerate interval parameters.

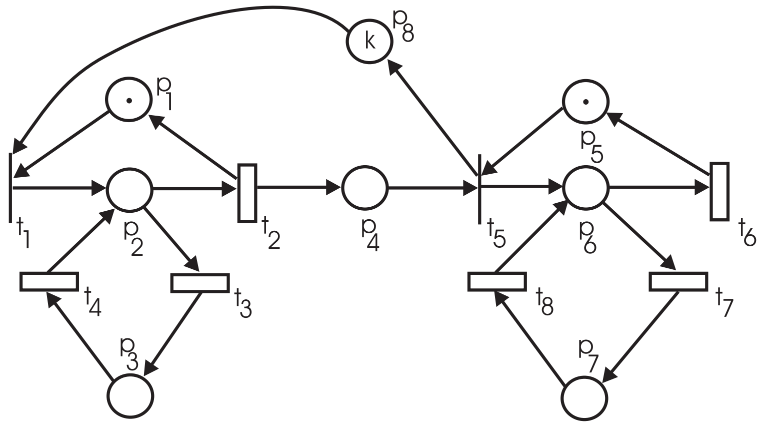

Fig. 4. Two-Machine One-Buffer Transfer Line Model ($k = 3$)

Transition	Value ($[t]^{-1}$)	Symbol
$[t_2]$	[1, 1]	$[\nu_1]$
$[t_3]$	[3, 3]	$[\lambda_1]$
$[t_4]$	[5, 5]	$[\mu_1]$
$[t_6]$	[2, 2]	$[\nu_2]$
$[t_7]$	[4, 4]	$[\lambda_2]$
$[t_8]$	[6, 6]	$[\mu_2]$

Table 3. Interval transition firing rates for the Two-Machine One-Buffer Transfer Line model.

As a result of the first step of ISPN analysis we obtain the reachability set (Table 4) and the reachability graph (Table 5).

Markings enabling the transitions t_1 and t_5 are vanishing, because enabled transitions are immediate (state changes that take negligible amounts of time to occur). Can be identified twelve vanishing markings M_0, M_2, M_4, M_5, M_7, M_{12}, M_{13}, M_{17}, M_{19}, M_{22}, M_{24}, M_{26} (firing of immediate transitions t_1 and t_5) and other markings are tangibles.

State	Marking[1]	State	Marking[1]
1	$M_0 = [1,0,0,0,1,0,0,3]$	15	$M_{14} = [0,0,1,1,0,1,0,1]$
2	$M_1 = [0,1,0,0,1,0,0,2]$	16	$M_{15} = [0,1,0,1,0,0,1,1]$
3	$M_2 = [1,0,0,1,1,0,0,2]$	17	$M_{16} = [0,1,0,2,0,1,0,0]$
4	$M_3 = [0,0,1,0,1,0,0,2]$	18	$M_{17} = [0,0,1,1,1,0,0,1]$
5	$M_4 = [0,1,0,1,1,0,0,1]$	19	$M_{18} = [0,0,1,1,0,0,1,1]$
6	$M_5 = [1,0,0,0,0,1,0,3]$	20	$M_{19} = [1,0,0,2,0,0,1,1]$
7	$M_6 = [0,1,0,0,0,1,0,2]$	21	$M_{20} = [1,0,0,3,0,1,0,0]$
8	$M_7 = [1,0,0,1,0,1,0,2]$	22	$M_{21} = [0,0,1,2,0,1,0,0]$
9	$M_8 = [0,0,1,0,0,1,0,2]$	23	$M_{22} = [0,1,0,2,1,0,0,0]$
10	$M_9 = [0,1,0,0,0,0,1,2]$	24	$M_{23} = [0,1,0,2,0,0,1,0]$
11	$M_{10} = [0,1,0,1,0,1,0,1]$	25	$M_{24} = [1,0,0,3,1,0,0,0]$
12	$M_{11} = [0,0,1,0,0,0,1,2]$	26	$M_{25} = [1,0,0,3,0,0,1,0]$
13	$M_{12} = [1,0,0,1,0,0,1,2]$	27	$M_{26} = [0,0,1,2,1,0,0,0]$
14	$M_{13} = [1,0,0,2,0,1,0,1]$	28	$M_{27} = [0,0,1,2,0,0,1,0]$

[1]- $Marking = [\ m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8\ ]$

Table 4. Reachability set and distribution markings from ISPN of Figure 4.

Marking $\vert$ ***Firing of transition*** $\rangle$ ***New marking***				
$M_0\ \vert t_1\rangle\ M_1$	$M_1\ \vert T_2\rangle\ M_2$	$M_1\ \vert T_3\rangle\ M_3$	$M_2\ \vert t_1\rangle\ M_4$	$M_2\ \vert t_5\rangle\ M_5$
$M_3\ \vert T_4\rangle\ M_1$	$M_4\ \vert t_5\rangle\ M_6$	$M_5\ \vert t_1\rangle\ M_6$	$M_6\ \vert T_2\rangle\ M_7$	$M_6\ \vert T_3\rangle\ M_8$
$M_6\ \vert T_6\rangle\ M_1$	$M_6\ \vert T_7\rangle\ M_9$	$M_7\ \vert t_1\rangle\ M_{10}$	$M_8\ \vert T_4\rangle\ M_6$	$M_8\ \vert T_6\rangle\ M_3$
$M_8\ \vert T_7\rangle\ M_{11}$	$M_9\ \vert T_2\rangle\ M_{12}$	$M_9\ \vert T_3\rangle\ M_{11}$	$M_9\ \vert T_8\rangle\ M_6$	$M_{10}\ \vert T_2\rangle\ M_{13}$
$M_{10}\ \vert T_3\rangle\ M_{14}$	$M_{10}\ \vert T_6\rangle\ M_4$	$M_{10}\ \vert T_7\rangle\ M_{15}$	$M_{11}\ \vert T_4\rangle\ M_9$	$M_{11}\ \vert T_8\rangle\ M_8$
$M_{12}\ \vert t_1\rangle\ M_{15}$	$M_{13}\ \vert t_1\rangle\ M_{16}$	$M_{14}\ \vert T_4\rangle\ M_{10}$	$M_{14}\ \vert T_6\rangle\ M_{17}$	$M_{14}\ \vert T_7\rangle\ M_{18}$
$M_{15}\ \vert T_2\rangle\ M_{19}$	$M_{15}\ \vert T_3\rangle\ M_{18}$	$M_{15}\ \vert T_8\rangle\ M_{10}$	$M_{16}\ \vert T_2\rangle\ M_{20}$	$M_{16}\ \vert T_3\rangle\ M_{21}$
$M_{16}\ \vert T_6\rangle\ M_{22}$	$M_{16}\ \vert T_7\rangle\ M_{23}$	$M_{17}\ \vert t_5\rangle\ M_8$	$M_{18}\ \vert T_4\rangle\ M_{15}$	$M_{18}\ \vert T_8\rangle\ M_{14}$
$M_{19}\ \vert t_1\rangle\ M_{23}$	$M_{20}\ \vert T_6\rangle\ M_{24}$	$M_{20}\ \vert T_7\rangle\ M_{25}$	$M_{21}\ \vert T_4\rangle\ M_{16}$	$M_{21}\ \vert T_6\rangle\ M_{26}$
$M_{21}\ \vert T_7\rangle\ M_{27}$	$M_{22}\ \vert t_5\rangle\ M_{10}$	$M_{23}\ \vert T_2\rangle\ M_{25}$	$M_{23}\ \vert T_3\rangle\ M_{27}$	$M_{23}\ \vert T_8\rangle\ M_{16}$
$M_{24}\ \vert t_5\rangle\ M_{13}$	$M_{25}\ \vert T_8\rangle\ M_{20}$	$M_{26}\ \vert t_5\rangle\ M_{14}$	$M_{27}\ \vert T_4\rangle\ M_{23}$	$M_{27}\ \vert T_8\rangle\ M_{21}$

Table 5. Literal description of reachability graph from ISPN of Figure 4.

Finally, we obtain the matrices $[\mathbf{P}]^{\mathcal{VV}}$, $[\mathbf{P}]^{\mathcal{VT}}$, $[\mathbf{U}]^{\mathcal{TV}}$ and $[\mathbf{U}]^{\mathcal{TT}}$:

$$[\mathbf{P}]^{\mathcal{VV}} = \begin{pmatrix}
[0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] \\
[0,\ 0] & [0,\ 0] & [0{,}5, 0{,}5] & [0{,}5,\ 0{,}5] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] \\
[0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] \\
[0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] \\
[0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] \\
[0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] \\
[0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] \\
[0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] \\
[0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] \\
[0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] \\
[0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [1,\ 1] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] \\
[0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0] & [0,\ 0]
\end{pmatrix}$$

$$[\mathbf{P}]^{\mathcal{V}\mathcal{T}} = \left(\begin{array}{cccccccccccc}
[1,1] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [1,1] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [1,1] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [1,1] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [1,1] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [1,1] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [1,1] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [1,1] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [1,1] & [0,0] & [0,0] & [0,0] & [0,0]
\end{array}\right.$$

$$\left.\begin{array}{cccc}
[0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [1,1] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0]
\end{array}\right)$$

$$[\mathbf{U}]^{\mathcal{T}\mathcal{T}} = \left(\begin{array}{cccccccccccc}
[0,0] & [3,3] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] \\
[5,5] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] \\
[2,2] & [0,0] & [0,0] & [3,3] & [4,4] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [2,2] & [5,5] & [0,0] & [0,0] & [0,0] & [4,4] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [6,6] & [0,0] & [0,0] & [0,0] & [3,3] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [3,3] & [4,4] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [6,6] & [5,5] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [5,5] & [0,0] & [0,0] & [0,0] & [0,0] & [4,4] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [6,6] & [0,0] & [0,0] & [0,0] & [0,0] & [3,3] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [1,1] \\
[0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [6,6] & [5,5] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [5,5] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [6,6] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [6,6] \\
[0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0] & [0,0]
\end{array}\right.$$

$$\left.\begin{array}{cccc}
[0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] \\
[3,3] & [4,4] & [0,0] & [0,0] \\
[0,0] & [0,0] & [0,0] & [0,0] \\
[0,0] & [0,0] & [4,4] & [0,0] \\
[0,0] & [0,0] & [0,0] & [4,4] \\
[0,0] & [0,0] & [1,1] & [3,3] \\
[0,0] & [0,0] & [0,0] & [0,0] \\
[6,6] & [5,5] & [0,0] & [0,0]
\end{array}\right)$$

$$
[\mathbf{U}]^{\mathcal{TV}} = \begin{pmatrix}
[0, 0] & [1, 1] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0,0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [1, 1] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [1, 1] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [2, 2] & [0, 0] & [0, 0] & [0, 0] & [1, 1] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [2, 2] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [1, 1] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [2, 2] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [2, 2] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [2, 2] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0]
\end{pmatrix}
$$

Afterwards, carry out the elimination of vanishing markings (Equation 5), to obtain the matrix of rate intervals $[U]$ representing the IREMC:

$$
[\mathbf{U}] = \left(\begin{matrix}
[0, 0] & [3, 3] & [1, 1] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[5, 5] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[2, 2] & [0, 0] & [0, 0] & [3, 3] & [4, 4] & [1, 1] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [2, 2] & [5, 5] & [0, 0] & [0, 0] & [0, 0] & [4, 4] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [6, 6] & [0, 0] & [0, 0] & [0, 0] & [3, 3] & [0, 0] & [1, 1] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [2, 2] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [3, 3] & [4, 4] & [1, 1] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [6, 6] & [5, 5] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [2, 2] & [0, 0] & [5, 5] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [4, 4] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [6, 6] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [3, 3] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [2, 2] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [1, 1] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [6, 6] & [5, 5] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [2, 2] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [2, 2] & [0, 0] & [5, 5] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [6, 6] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [6, 6] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0]
\end{matrix}\right.
$$

$$
\left.\begin{matrix}
[0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [1, 1] & [0, 0] & [0, 0] \\
[3, 3] & [4, 4] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [4, 4] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [4, 4] \\
[0, 0] & [0, 0] & [1, 1] & [3, 3] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[6, 6] & [5, 5] & [0, 0] & [0, 0]
\end{matrix}\right) .
$$

Finally, using Equation 6, we find the infinitesimal generator interval matrix:

$$[\mathbf{Q}] = \left(\begin{array}{ccccccccc}
[-4, -4] & [3, 3] & [1, 1] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[5, 5] & [-5, -5] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[2, 2] & [0, 0] & [-10, -10] & [3, 3] & [4, 4] & [1, 1] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [2, 2] & [5, 5] & [-11, -11] & [0, 0] & [0, 0] & [4, 4] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [6, 6] & [0, 0] & [-10, -10] & [0, 0] & [3, 3] & [0, 0] & [1, 1] \\
[0, 0] & [0, 0] & [2, 2] & [0, 0] & [0, 0] & [-10, -10] & [0, 0] & [3, 3] & [4, 4] \\
[0, 0] & [0, 0] & [0, 0] & [6, 6] & [5, 5] & [0, 0] & [-11, -11] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [2, 2] & [0, 0] & [5, 5] & [0, 0] & [-11, -11] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [6, 6] & [0, 0] & [0, 0] & [-10, -10] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [2, 2] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [6, 6] & [5, 5] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [2, 2] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0]
\end{array} \right.$$

$$\left. \begin{array}{ccccccc}
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[1, 1] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [4, 4] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[0, 0] & [3, 3] & [0, 0] & [0, 0] & [1, 1] & [0, 0] & [0, 0] \\
[-10, -10] & [0, 0] & [1, 1] & [3, 3] & [4, 4] & [0, 0] & [0, 0] \\
[0, 0] & [-11, -11] & [0, 0] & [0, 0] & [0, 0] & [0, 0] & [0, 0] \\
[2, 2] & [0, 0] & [-6, -6] & [0, 0] & [0, 0] & [4, 4] & [0, 0] \\
[5, 5] & [0, 0] & [0, 0] & [-11, -11] & [0, 0] & [0, 0] & [4, 4] \\
[6, 6] & [0, 0] & [0, 0] & [0, 0] & [-10, -10] & [1, 1] & [3, 3] \\
[0, 0] & [0, 0] & [6, 6] & [0, 0] & [0, 0] & [-6, -6] & [0, 0] \\
[0, 0] & [6, 6] & [5, 5] & [0, 0] & [-11, -11] & [0, 0] & [0, 0]
\end{array} \right)$$

out by the verifylss function of the MATLAB
The third step of ISPN analysis solves the system of interval linear equations described by Equation (7). The interval linear equations solution is carried out by the verifylss function of the MATLAB *toolbox* INTLAB. Substituting the last equation of system $[\vec{\pi}] \cdot [\mathbf{Q}] = 0$ by the normalization condition $\sum_{i=1}^{16} [\pi]_i = 1$, the linear system $([\pi]_1, [\pi]_2) \cdot [\mathbf{A}] = [\mathbf{b}]$ is obtained. The solution of this system directly provides the steady state probabilities of tangible states:

$$[\vec{\pi}]^t = \begin{pmatrix} [\ 0.30162341059172.\ \ 0.30162341059173] \\ [\ 0.20129241213850.\ \ 0.20129241213851] \\ [\ 0.10001579083719.\ \ 0.10001579083720] \\ [\ 0.05079591445866.\ \ 0.05079591445867] \\ [\ 0.05701697985273.\ \ 0.05701697985274] \\ [\ 0.05122652318522.\ \ 0.05122652318523] \\ [\ 0.03402132703571.\ \ 0.03402132703572] \\ [\ 0.02728986215974.\ \ 0.02728986215975] \\ [\ 0.03607316886026.\ \ 0.03607316886027] \\ [\ 0.02968055852738.\ \ 0.02968055852739] \\ [\ 0.01976172320179.\ \ 0.01976172320180] \\ [\ 0.02527256969829.\ \ 0.02527256969830] \\ [\ 0.01396928749535.\ \ 0.01396928749536] \\ [\ 0.02086458086920.\ \ 0.02086458086922] \\ [\ 0.02032580994373.\ \ 0.02032580994374] \\ [\ 0.01077008114445.\ \ 0.01077008114446] \end{pmatrix}.$$

Finally we can make the fourth (final) step of analysis ISPN, i.e. computation of metrics. The average utilization of machines, i.e., the probability that a machine is processing a part are:

$$[UM_1] = [prob](m(p_2) = 1) \ \ and \ \ [UM_2] = [prob](m(p_6) = 1).$$

The evaluation result provides the following values:

$$[UM_1] = [0.59650101272372, 0.59650101272374] \ \ and$$

$$[UM_2] = [0.29825050636186, 0.29825050636187].$$

These results gives interval bounds to exact value and can be used to verify conventional analysis of GSPN results.

Experiment for Two-Machine One-Buffer Transfer Line Model

Table 6 shows the average machine utilization, UM_1 and UM_2, for three μ_1 rate intervals (degenerated intervals). ISPN analysis results, provided by ISPN MATLAB toolbox INTLAB, are GSPN ordinary results with verified interval bounds.

Interval rate $[\mu_1]$	**Machine utilization**	
	$[UM_1]$	$[UM_2]$
[0.1E2, 0.1E2]	[0.11946700722573, 0.11946700722574]	[0.59733503612865, 0.59733503612866]
[0.1E1, 0.1E1]	[0.59650101272372, 0.59650101272374]	[0.29825050636186, 0.29825050636187]
[0.2E0, 0.2E0]	[0.62490104707753, 0.62490104707755]	[0.06249010470775, 0.06249010470776]

Table 6. Experiment for Two-Machine One-Buffer Transfer Line Model for three MR (Machining Rate) = μ_1 (degenerated interval). Results obtained with ISPN MATLAB toolbox INTLAB Prototype Tool.

Interval rate $[\mu_1]$	**Machine utilization**	
	$[UM_1]$	$[UM_2]$
$[0.099E2, 0.101E2]$ $=$ $0,100E2 \pm 0,001E2$	$[0.11399679921745, 0.12493721523401]$ $=$ $0.11946700722573 \pm 0.00547020800828$	$[0.57960670982656, 0.61506336243075]$ $=$ $0.59733503612865 \pm 0.01772832630210$
$[0.099E1, 0.101E1]$ $=$ $0,100E1 \pm 0,001E1$	$[0.49611631459760, 0.69688571084986]$ $=$ $0.59650101272373 \pm 0.10038469812613$	$[0.22827877740233, 0.36822223532140]$ $=$ $0.29825050636187 \pm 0.06997172895954$
$[0,199E0, 0,201E0]$ $=$ $0,200E0 \pm 0,001E0$	$[0.54449037658809, 0.70531171756699]$ $=$ $0.62490104707754 \pm 0.08041067048945$	$[0.02346019982939, 0.10152000958611]$ $=$ $0.06249010470775 \pm 0.03902990487836$

Table 7. The average machine utilization results obtained with ISPN MATLAB toolbox INTLAB Prototype Tool to Two-Machine One-Buffer Transfer Line Model for three μ_1 rate intervals.

Introducing parameters with input uncertainties:

In sequel, the variations in the rates of exponential transitions are considered. To avoid redundancy, will not be displayed detailing of ISPN analysis as in previous examples. Table 7 shows the average machine utilization, UM_1 and UM_2 for three $[\mu_1]$ rate intervals. All exponential rate variabilities have ±1 as errors in the 3^{rd} significant digits:

ISPN.m Line 59 modification for each experiment:

- AT(3,1)= [infsup(0.099E2,0.101E2),infsup(2.99,3.01),infsup(4.99,5.01),infsup(1.99,2.01), infsup(3.99,4.01),infsup(5.99,6.01)];
- AT(3,1)= [infsup(0.099E1,0.101E1),infsup(2.99,3.01),infsup(4.99,5.01),infsup(1.99,2.01), infsup(3.99,4.01),infsup(5.99,6.01)];
- AT(3,1)= [infsup(0.199E0, 0.201E0),infsup(2.99,3.01),infsup(4.99,5.01),infsup(1.99,2.01), infsup(3.99,4.01),infsup(5.99,6.01)];

5. ISPN MATLAB toolbox INTLAB prototype tool

ISPN M-file MATLAB toolbox INTLAB is a prototype for the modeling and evaluation of ISPNs in which exponential transition rates and immediate transition weights may be represented by intervals. Models are specified by matrix input/output arc multiplicity of transitions as a direct mapping of usual graphical Petri Nets representation description of systems. The stationary analysis is based on Markov theory. An interval embedded Markov chain (IEMC), constructed and solved by interval methods, allow us computation metrics.
The current prototype is still being used but **ISPN.n** will allow you to write your own features and to tailor ISPNs to your own needs.

The **ISPN.m** used for calculating verified probabilities and the machine utilization rate from ISPN model of Figure 4, is given bellow:
Uncomment specified lines to display:

- **Line 191:** Reachability set and distribution markings from ISPN model (Table 4)
- **Line 192:** Literal description of reachability graph from ISPN model (Table 4)
- **Line 213:** $[P]^{\mathcal{VV}}$
- **Line 218:** $[P]^{\mathcal{VT}}$
- **Line 223:** $[U]^{\mathcal{TV}}$
- **Line 229:** $[U]^{\mathcal{TT}}$
- **Line 237:** $[U]$
- **Line 237:** $[Q]$

```
  1. %
  2. %  ISPN prototype tool
  3. %
  4. %  INPUT MODEL : Two-Machine One-Buffer Transfer Line Model ( Fig. 4 )
  5. %
  6. %  Obs: Using matrix notation of ISPN model
  7. datestr(now,0)
  8. format long
  9. clear At % Clear variable At
 10. % input arc multiplicity of immediate transitions, (-) minus means input
 11. At(1,1)= {[-1, 0;   % P1
 12.             0, 0;   % P2
 13.             0, 0;   % P3
 14.             0,-1;   % P4
 15.             0,-1;   % P5
 16.             0, 0;   % P6
 17.             0, 0;   % P7
 18.            -1, 0]}; % P8
 19. % labels of immediate transition
 20. At(2,1)={ ['t1';'t5']};
 21. % weight of immediate transition
 22. At(3,1)= {[1,1]};
 23. % celldisp(At) % uncomment display cell array contents
 24. clear AtO %  Clear variable AtO
 25. % output arc multiplicity of immediate transitions
 26. AtO(1,1)= {[ 0, 0;   % P1
 27.              1, 0;   % P2
 28.              0, 0;   % P3
 29.              0, 0;   % P4
 30.              0, 0;   % P5
 31.              0, 1;   % P6
 32.              0, 0;   % P7
 33.              0, 1]}; % P8
 34. % celldisp(AtO) % uncomment display cell array contents
 35. clear Ai % Clear variable Ai
 36. % arc multiplicity of inhibitor arcs (associeted to immediate transitions)
 37. Ai(1,1)= {[0, 0;   % P1
 38.            0, 0;   % P2
 39.            0, 0;   % P3
 40.            0, 0;   % P4
 41.            0, 0;   % P5
 42.            0, 0;   % P6
 43.            0, 0;   % P7
 44.            0, 0]}; % P8
 45. % celldisp(Ai) % uncomment display cell array contents
 46. clear AT % Clear variable AT
 47. % input arc multiplicity of timed transitions, ( - )  minus means input
```

```
AT(1,1)= {[ 0, 0, 0, 0, 0, 0;
           -1,-1, 0, 0, 0, 0;
            0, 0,-1, 0, 0, 0;
            0, 0, 0, 0, 0, 0;
            0, 0, 0, 0, 0, 0;
            0, 0, 0,-1,-1, 0;
            0, 0, 0, 0, 0,-1;
            0, 0, 0, 0, 0, 0]};
% labels of timed transitions
AT(2,1)={ [’T2’;’T3’;’T4’;’T6’;’T7’;’T8’]};
% interval rate  of timed transitions (degenereted)
AT(3,1)= {[1,3,5,2,4,6]};
% server semantics of timed transitions
AT(4,1)= {[’SS’;’SS’;’SS’;’SS’;’SS’;’SS’]};
% celldisp(AT) % uncomment display cell array contents
clear ATO  % Clear variable ATO
ATO(1,1)= {[1, 0, 0, 0, 0, 0;
            0, 0, 1, 0, 0, 0;
            0, 1, 0, 0, 0, 0;
            1, 0, 0, 0, 0, 0;
            0, 0, 0, 1, 0, 0;
            0, 0, 0, 0, 0, 1;
            0, 0, 0, 0, 1, 0;
            0, 0, 0, 0, 0, 0]};
% celldisp(ATO)  % uncomment display cell array contents
clear M % Clear variable M
clear d % Clear variable d
% Initial marking of each place (initial state)
M(1,1)={ [1;0;0;0;1;0;0;3] };
M(2,1)={ ’M0’ };
im=0;
ivm=1;
itm=1;
n=size(At{1},2); % number of columns of At
nT=size(AT{1},2); % number of columns of  AT
m=size(AT{1},1);  % number of rows of AT
id=1;
ic=1;
Q=infsup(0.0,0.0);
while im < size(M,2)
    j=0;
    t=0;
    tvm=1;
    mx=M{1,im+1};
    for i=1:n
        mt=At{1,1} (1:end,i);
        ai=0;
        mi=Ai{1,1} (1:end,i);
        mi=mx-mi;
        if mi==mx | min(mi)<0
            ai=1;
        end
        md=mx+mt;
        if min(md)>=0 & ai==1
            d{id,1}=strcat(M{2,1+im} (1:end),’|’, At{2} (i,1:end),’>’);
            md=md+AtO{1,1} (1:end,i);
            t=1;
            tvm=0;
            id=id+1;
            c=size(M,2);
            xt=’new’;
            for ix = 1:c
                if md == M{1,ix}
                    x=M{2,ix};
                    d{id-1,2}=x;
                     xt=’old’;
                     Q(im+1,ix)=At{3,1} (i);
                    break
                end
            end
            if xt==’new’
                strx=strcat(’M’,num2str(c));
                d{id-1,2}=strx;
                M{1,c+1} =md;
                M{2,c+1}= strx;
                Q(im+1,c+1)=At{3,1} (i);
            end
        end
    end
    % If there is no firing of immediate transitions so we try
    % to firing timed transitions
    if t==0
        for i= 1:nT
           mt=AT{1,1} (1:end,i);
           md=mx+mt;
           min(md);
           if min(md) >= 0
             d{id,1}=strcat(M{2,1+im} (1:end),’|’, AT{2} (i,1:end),’>’);
             md=md+ATO{1,1} (1:end,i);
             ma=md;
             ga=0;
             while min(ma)>=0
                 ma=ma+AT{1,1} (1:end,i);
                 ga=ga+1;
             end
             t=1;
             id=id+1;
             c=size(M,2);
             xt=’new’;
             for ix = 1:c
                    if md == M{1,ix}
                        x=M{2,ix};
                        d{id-1,2}=x;
                        xt=’old’;
                        if AT{4} (i,1:end) == ’SS’
                              Q(im+1,ix)=AT{3,1} (i);
                        end
                        if AT{4} (i,1:end) == ’IS’
                            Q(im+1,ix)=ga*AT{3,1} (i);
                        end
                        break
                    end
               end
                if xt==’new’
                    strx=strcat(’M’,num2str(c));
                    d{id-1,2}=strx;
                    M{1,c+1} =md;
                    M{2,c+1}= strx;
                    if AT{4} (i,1:end) == ’SS’
                              Q(im+1,c+1)=AT{3,1} (i);
                          end
                    if AT{4} (i,1:end) == ’IS’
                              Q(im+1,c+1)=ga*AT{3,1} (i);
                    end
                end
            end
        end
    end
    im=im+1; % provoca termino do loop while
    if tvm==0
        vm(ivm)=im-1;
        ivm=ivm+1;
    else
        tm(itm)=im-1;
        itm=itm+1;
    end
```

```
end
clear At % Clear variable At
clear AtO % Clear variable AtO
clear AT % Clear variable AT
clear ATO % Clear variable ATO
clear Ai % Clear variable Ai
%===============================================
% celldisp(M);  % uncomment display reachability set and distribution markings from ISPN = ( Table 4 )
% d % uncomment display literal description of reachability graph from ISPN = ( Table 5 )
% vm % uncomment display vanishing markings index vector
% tm % uncomment display tangible markings index vector
%==============================================
% ’number of vanishing markings’
ivm=ivm-1;
% ’number of tangible markings’
itm=itm-1;
n=ivm+itm;
clear PVV % Clear variable PVV
PVV=intval(zeros(ivm,ivm));
i = (1:ivm);
j=(1:ivm);
PVV(i,j)=Q(vm(i)+1,vm(j)+1);
% weigths uniformization of immediate transitions
for i= 1:ivm
    s=sum(PVV(i,1:ivm));
    if s>1
        PVV(i,1:ivm)=PVV(i,1:ivm)/s;
    end
end
% PVV % uncomment display PVV
clear PVT % Clear variable PVV
i = (1:ivm);
j=(1:itm);
        PVT(i,j)=Q(vm(i)+1,tm(j)+1);
% PVT % uncomment display PVT
clear UTV % Clear variable UTV
i = (1:itm);
j=(1:ivm);
      UTV(i,j)=Q(tm(i)+1,vm(j)+1);
% UTV % uncomment display UTV
clear UTT  % Clear variable UTT
i = (1:itm);
j=(1:itm);
    UTT(i,j)=Q(tm(i)+1,tm(j)+1);
clear Q % Clear variable Q
% UTT % uncomment display UTT
% ’Calculating X=eye(size(PVV,1))-PVV’
X=eye(size(PVV,1))-PVV;
clear PVV % Clear variable PVV
% ’Calculating X=inv(X)’
X=inv(X);
% ’Calculating  U=UTT+UTV*X*PVT’
U=UTT+UTV*X*PVT;
% U % uncomment display U
clear PVT % Clear variable PVT
clear UTT % Clear variable UTT
clear UTV % Clear variable UTV
clear X % Clear variable X
n=size(U,1);
t=sum(U.’);
    for i=1:n
        U(i,i)=-t(i);
    end
Q=U;
% Q  % uncomment display Q
clear U % Clear variable U
```

```
QZ=Q.';
clear Q % Clear variable Q
m=size(QZ,1);
j= 1:m;
QZ(m,j)=1;
QZ;
Z(m)=1;
Z=Z.';
% 'Calculating x=verifylss(QZ,Z)'
x=verifylss(QZ,Z)
% [UM1] = [prob](m(p_{2}) = 1)
'UM1'
 s1=0;
 for i= 1:n
    if M{1,tm(i)+1} (2) >0
             s1=s1+x(i);
    end
end
intvalinit('displaymidrad')
s1
intvalinit('displayinfsup')
s1
% [UM2] = [prob](m(p_{6}) = 1)
'UM2'
s1=0;
for i= 1:n
   if M{1,tm(i)+1} (6) >0
             s1=s1+x(i);
   end
end
intvalinit('displaymidrad')
s1
intvalinit('displayinfsup')
s1
datestr(now,0)
```

Executing **ISPN.m** yields:

```
>> ISPN
ans =
01-Apr-2011 23:59:14
intval x =
[   0.30162341059172,   0.30162341059173]
[   0.20129241213850,   0.20129241213851]
[   0.10001579083719,   0.10001579083720]
[   0.05079591445866,   0.05079591445867]
[   0.05701697985273,   0.05701697985274]
[   0.05122652318522,   0.05122652318523]
[   0.03402132703571,   0.03402132703572]
[   0.02728986215974,   0.02728986215975]
[   0.03607316886026,   0.03607316886027]
[   0.02968055852738,   0.02968055852739]
[   0.01976172320179,   0.01976172320180]
[   0.02527256969829,   0.02527256969830]
[   0.01396928749535,   0.01396928749536]
[   0.02086458086920,   0.02086458086922]
[   0.02032580994373,   0.02032580994374]
[   0.01077008114445,   0.01077008114446]
ans =
UM1
===> Default display of intervals by midpoint/radius (e.g. < 3.14 , 0.01 >)
intval s1 =
<   0.59650101272373,  0.00000000000001>
===> Default display of intervals by infimum/supremum (e.g. [ 3.14 , 3.15 ])
intval s1 =
[   0.59650101272372,   0.59650101272374]
ans =
UM2
===> Default display of intervals by midpoint/radius (e.g. < 3.14 , 0.01 >)
intval s1 =
<   0.29825050636187,  0.00000000000001>
===> Default display of intervals by infimum/supremum (e.g. [ 3.14 , 3.15 ])
intval s1 =
[   0.29825050636186,   0.29825050636187]
ans =
01-Apr-2011 23:59:14
>>
```

6. Concluding remarks

In this chapter, ISPN is used as an approach to ISPN performance analysis in which the exponential rates fall within pre-assumed intervals. ISPN is mainly applied in modeling, where input data are known within definite interval of accuracy. Such uncertainties include the errors involved with experimental data obtained from measurements. This framework provides a way to formalize and study problems related to the presence of uncertainties. Such uncertainties include data errors occurring during data measurements and rounding errors generated during calculations. The model proposed and the related method of analysis, involves the case of simultaneous variability in values of parameters. As an immediate consequence, the ISPN analysis, designed for evaluation of results obtained from measurements, may appear to be useful for engineers and technicians as a tool for decision making. As future works, methods for interval transient analysis and simulation should considered. Furthermore, other case studies should also be take into account. ISPN MATLAB toolbox INTLAB Prototype Tool will allow you to specify your own ISPNs. We plan to post future developments of ISPN MATLAB toolbox INTLAB Prototype Tool.

7. References

Bolch, G., Greiner, S. & Trived, K. (2006). *Queueing networks and Markov chains: modeling and performance evaluation with computer science applications*, A Wiley-Inerscience publication, John Wiley and Sons, Inc, Second Edition.

Brinksma, E. (2001). *Lectures on formal methods and performance analysis: revised lectures/ First EEF Summer School on Trends in Computer Science. Berg en Dal, The Netherlands, July 3 - 7, 2000. 9ed.*, Springer, Ed Brinksma (editor).

Chen, H. M. & van Emden, M. H. (1995). Adding interval constraints to the moore skelboe global optimization algorithm, *in* V. Kreinovich (ed.), *Extended Abstracts of APIC'95, International Workshop on Applications of Interval Computation*, Reliable Computing (Supplement), pp. 54–57.

Desrochers, A. & Al-Jaar, R. (1994). *Applications of Petri nets in manufacturing systems: modeling, control, and performance analysis*, IEEE Press, Piscataway, NJ, USA.

Florin, G. & Natkin, S. (1989). Matrix product form solution for closed synchronized queueing networks, In In Proc. 3rd Intern. Workshop on Petri Nets and Performance Models, pp. 29–39.

Galdino, S. & Maciel, P. (2006). Interval generalized stochastic petri net models in performance evaluation, *IEEE International Conference on Systems, Man, and Cybernetics 2006*, Taiwan, pp. 2723– 2728.

Galdino, S., Maciel, P. & Rosa, N. S. (2007a). Interval generalized stochastic petri net models in dependability evaluation, *IEEE International Conference on Systems, Man, and Cybernetics 2007*, Montreal, pp. 2502–2507.

Galdino, S., Maciel, P. & Rosa, N. S. (2007b). Interval markovian models in dependability evaluation, *International Journal of Pure and Applied Mathematics* Vol. 41(No. 2): 151–176.

German, R. (2000). *Performance Analysis of Communicating Systems - Modeling with Non-Markovian Stochastic Petri Nets*, Wiley.

Girault, C. & Valk, R. (2003). *Petri Nets for Systems Engineering: a guide to modeling, verification, and applications*, Springer.

Hansen, E. & Walster, G. (2004). *Global Optimization Using Internal Analysis*, Second Edition, Revised and Expanded, Marcel Dekker, Inc., New York.

Jaulin, L., Kieffer, M., Didrit, O. & Walter, E. (2001). *Applied Interval Analysis: with Examples in Parameter and State Estimation, Robust Control and Robotics*, Springer-Verlag.

Marsan, M., Balbo, G. & Conte, G. (1984). A class of generalized sthochastic petri nets for the performance evaluation of multiprocessor systems, *ACM Transactions on Computer Systems* 2: 93–122.

Marsan, M., Bobbio, A., Conte, G. & Cumani, A. (1984). Performance analysis of degradable multiprocessor systems using generalized stochastic petri nets, *IEEE Computer Society Newsletters, 6 SI-* 1: 47–54.

Merlin, P. & Farber, D. (1976). Recoverability of communication protocols: Implications of a theoretical study, Vol. 24, No. 9, IEEE Trans. Comm., pp. 1036–1043.

Molloy, M. (1981). *On the Integration of Delay and Throughput Measures in Distributed Processing Models*, PhD thesis, UCLA, Los Angeles, CA.

Moore, R. E. (1962). *Interval Arithmetic and Automatic Error Analysis in Digital Computing*, PhD thesis, Stanford University.

Moore, R. E. (1966). *Interval Analysis*, Prentice Hall, Englewood Clifs, NJ, USA.

Murata, T. (1989). Petri nets: Properties, analysis and application (an invited survey paper), *Proceedings of the IEEE* 77: 541–580.

Rall, L. (1969). *Computacional Solution of Nonlinear Operator Equations*, Wiley, New York.

Ramchandani, C. (1974). *Analysis of Asynchronous Concurrent Systems by Timed Petri Nets*, PhD thesis, MIT, Cambridge, MA.

Sifakis, J. (1978). Petri nets for performance evaluation, In 3rd Intern. Symp. IFIP, pp. 75–93.

Smith, C. U. (1990). *Petri Nets for Systems Engineering: a guide to modeling, verification, and applications*, Addison-Wesley, Reading, MA.

Ullrich, C. (1990). *in* C. Ullrich (ed.), *Contributions to Computer Arithmetic and Self-Validating Numerical Methods*, Vol. 7, J. C. Baltzer AG, Scientific Publishing Co.

Young, R. C. (1931). The algebra of many-valued quantities, *Math. Ann.* 104: 260–290.

APPLICATIONS OF MATLAB IN SCIENCE AND ENGINEERING
Edited by Tadeusz Michalowski
INTECH