|  |  |  |  |
| --- | --- | --- | --- |
|  | ***Modeling and Automation of Industrial Processes***  *MEEC/MEGE - 2021 / 2022* |  | *Group: \_\_\_\_*  *\_\_-\_\_\_\_\_*  *\_\_-\_\_\_\_\_*  *\_\_-\_\_\_\_\_*  *\_\_-\_\_\_\_\_* |

***2nd Lab. - Alarm System as a Discrete Event System [[1]](#footnote-1)***

***Part B – Stochastic Simulation and Markov Chains***

This laboratory assignment aims at studying Discrete Event Systems (DESs) in the aspects of modeling and analysis of properties. This assignment develops the previous assignment by describing its high-level system as a DES and studying its operation.

The main objective of this last phase, part B, is the application of stochastic simulation to assess system operation, namely permanence (percentage of time) the system is in each state. Alternatively, the Petri net system with stochastic transitions can also be described as a Markov Chain (MC), from which the permanence may also be estimated.

In this part of the work, the Petri net describing the intrusion alarm system is simplified to have a small incidence matrix. The number of places (conditions) is reduced to three:

– OFF mode

– Presence detection mode

– Alarm mode

The set of transitions is and the incidence matrix is:

where , and an entry denotes two arcs, namely and . the initial marking is .

The tools mostly used in this work, part B, run in MATLAB / JAVA. Using the login information indicated in the course webpage, see in the course SVN:

- The graphical freeware editor "PIPE2" which allows creating Petri Net models to import with the MATLAB toolbox "TPN5", function rdp (use the version available in the SVN)

- MATLAB functions simulating Petri nets, in particular the *"5 Philosophers"*.

**B1.** *[Graphical representation]*Draw the Petri net proposed in this guide, part B, using the PIPE2 editor. Simulate by hand a small sequence of events, show the reached states, therefore demonstrating the functionality of the net.

**B2.** *[Reachable Set]* Indicate the reachable set, , of the Petri net. While in general a Petri net cannot be represented as a Finite State Machine (FSM), can the Petri net proposed in this part of the work be represented as an FSM?

**B3.** *[Stochastic firing vector]* Let denote the probability of firing transition . Let the following characterize all transitions:

which imply , and . This probabilistic definition allows running a Petri net in a stochastic way, not having to manually define the firing of transitions, by setting the firing vector with the following code:

alpha1=0.06; alpha2=0.45; beta=0.92; gamma=0.84;

a= rand(1); b= rand(1); c= rand(1);

q1= (a<alpha1);

q2= (alpha1<=a & a<alpha1+alpha2);

q3= (alpha1+alpha2 <= a);

qk= [q1 q2 q3 (b<beta) (beta<=b) (c<gamma) (gamma<=c)]';

Use this firing vector in the simulation of the Petri net defined in this part, part B, of the assignment (see also suggestions in Annex A2). After the simulation, compute the percentage of the total time the Petri net stays in each state .

**B4.** *[Markov chain, right stochastic matrix]* Alternatively to simulating the Petri net with stochastic instances of the firing vector, one may compute the percentage of time the net stays in a state using a Markov chain formalism [[2]](#footnote-2). Considering the states of a Petri net, , one forms a Markov chain state vector

where denotes the probability of the Petri net being at state . The dynamic equation of the Markov chain is . In the Petri net here considered, we have

Matrix is known as a *right stochastic matrix*, meaning that each line sums to one. In case the Petri net starts at state , one has . As grows to infinity, the state converges to a steady value simply computed as the eigenvector of associated to the eigenvalue . What is the value of ?

Note, in MATLAB one computes eigenvalues and eigenvectors with:

[vecs, vals]= eig( A' ); eigenvalues= diag(vals), eigenvectors= vecs

**B5.** *[Tuning the stochastic parameters]* Suggest a way of tuning the values and so that   
 where .

1. Revised by Prof. José Gaspar (2022). [↑](#footnote-ref-1)
2. See references

   Chapter 7 of "Introduction to dynamic systems; theory, models, and applications", David G. Luenberger, Wiley 1979

   Chapter 7 of "Introduction to discrete event systems", C. Cassandras and S. Lafortune, Springer 2008 [↑](#footnote-ref-2)