# Modeling an Aggregate of Interfaces in a Discrete Space and Time

**Stanislav Simeonov, Neli Simeonova**

***Abstract— Visually impaired people need improvement of their communicability to contact with other people. Different solutions of the improvement of the man-computer interaction could help the visually impaired people use their abilities to full extent. The Human-Computer Interaction (HCI) is a basic term described as the way a user communicates or interacts with computers. To guarantee the ability to acquire information, the computer interface should include software and hardware elements to facilitate this perception. The development of computer technology provides possibilities to combine multiple performing devices into an integrated system aimed to give or facilitate certain services. In somewhat narrower context, the present work is related to the problems in the design of complex interface and performing devices for the people with reduced sight or totally blind ones. This is a prerequisite for the realization not only of interface devices but also of systems capable of partial or full processing of information. In the presented study is shown a model of a single system. The study is the result of research project funded by Bulgarian National Science Fund – NSF Grant No D-ID-02/14 and Grant NHT-333/14.***

***Keywords: Automation, Blind people, Computer Interfaces, IT Architecture, Modeling, User support,***

## I. Introduction

Visually impaired people need improvement of their communicability to contact with other people. Different solutions of the improvement of the man-computer interaction could help the visually impaired people use their abilities to full extent.

The massive deployment of graphical environments helps the users by creating intuitive graphical interfaces. An image of desktop is simulated similar to the usual working environment. After Apple, Microsoft also created a desktop image for the graphical interface of Windows. UNIX operation system also tends to improve its graphical interface.

These graphical interfaces allowing for intuitive interaction with the users, however, are a great challenge for the visually impaired people. The fast expansion of Internet and the use of graphical browsers creates heavy difficulties for the visually impaired. Based on the fact that navigation and individual programs are presented as pictograms, one easily can conclude that blind people need new concept for their interaction with computers.

During the last decade, the natural evolution of user interfaces is oriented to a model based exclusively in graphical presentation. This process limited the use of computers and access to new technologies by visually impaired people. The development of special applications for them requires new technologies, instrumentation and communication media. A blind user cannot manipulate alone the CMOS setup utility, install an OS or use Safe mode in MS Windows. It is hard for him to navigate within the increasing number of interactive network applications.

The Human-Computer Interaction (HCI) is a basic term described as the way a user communicates or interacts with computers. Generally, these are menus, direct manipulation of files and natural languages. The introduction of graphical interfaces started back in 1980-ies with the so called pseudo-graphic representations and auxiliary elements. After 1990, the graphical interfaces were massively introduced not only within certain programs but also as an aid to effective management of OS resources. These interfaces are characterized by a principle based on direct manipulation of objects in the environment. The interfaces involve interpretation of direct user actions (like select, pull, etc.) on the visually representations of the interface objects (e.g. icons) with specific input devices. This definition of HCI style can be generalized as WIMP (Windows, Icons, Menus, Pointers). This term is used in accordance with the Graphical User Interface (GUI). Such interfaces implement graphical presentation of elements like windows, buttons and icons. The individual graphical objects are manipulated by specific input devices allowing intuitive actions of the user. The user executes certain actions on these objects using the input devices. The WIMP style is easy to learn and use. Most applications implement similar ways of visualizing and similar style of work, thus making user life easier. Obviously, many of the WIMP style achievements are useless for the visually impaired people. After 1990, with the migration from text-oriented to graphical interfaces, the jobs of many people with reduced sight were put at risk.

The work of visually impaired with computers has particular characteristics. To guarantee the ability to acquire information, the computer interface should include software and hardware elements to facilitate this perception. Such components are – link to Braille terminals, specialized matrices, voice interfaces in both directions and tactile systems of perception. As a result of the technological developments of individual components, a concentration of multifinctionality is achieved within certain interface.

**Manuscript published on 30 October 2014.**
***** Correspondence Author (s)

**Stanislav Simeonov***, Computer Systems and Technologies, University “Prof. Dr. Asen Zlatarov”, Burgas, Bulgaria.
**Neli Simeonova**, Electrical engineering and electronics, University “Prof. Dr. Asen Zlatarov”, Burgas, Bulgaria.

The development of computer technology provides possibilities to combine multiple performing devices into an integrated system aimed to give or facilitate certain services. In somewhat narrower context, the present work is related to the problems in the design of complex interface and performing devices for the people with reduced sight or totally blind ones. The development of mobile devices and especially the newly elaborated embedded systems present impressive productivity. This is a prerequisite for the realization not only of interface devices but also of systems capable of partial or full processing of information.

The considerations below are not limited to the realization of combined, intelligent and interactive interfaces or complexes of them for visually impaired. The aim is to create conditions for modeling general purpose systems, study their stability alone and within a communication between them independent on the communication standard. Simultaneously, an adequate method for description and studies of computer systems is presented (an intelligent interface, when implemented, is actually a computer system).

## II. Problem Formulation

Every specialized interface combined with embedded system has all the features of an intelligent computer device. Every embedded device is portable which provides the opportunity of certain mobility of the user. It seems logical to look for possibilities to combine the functionalities of each device to obtain an intelligent surface of interconnected devices acting as a unified whole.

Both centralized and distributed approaches are used to realize connectivity. The centralized connectivity problem is solved and it gives limited possibilities to deliver dynamic mobility of every participant.

Real systems are often discrete. However, the processes taking place in them are probabilistic. The probabilistic approach is suitable for their formalization. It is applicable for abstract representation of the structural links, transitions between states and process development. The essence is to define probability parameters reflecting the stochastic nature of processes.

Stochastic process is a mathematical model describing an empiric process which changes depending on certain variable (usually time for real processes) according to a probabilistic determination. Particularly, the stochastic process is a set of random variables $\{X(t), t \in T\}$, defined in a space of random events and indexed by the parameter t $\{t \in T\}$ where t will be referred to as time-parameter. The probability for X(t) to obtain value of i, or P[X(t)=i] is an aggregate in the probabilistic space.

A stochastic process is characterized by three parameters:

- Space of states - The values of the random variable X(t) are called “states” and the aggregate of all possible values form the “space of states” for this process. If the value of $X(t)=P_i$, then the process is in state *i*. If the space of states is finite or countable infinite one, the process occupies discrete states and this is often defined as stochastic chain. In such cases, the space of states is assumed to consist of non-negative integers {0, 1, 2……}. On the other hand, if the space of states is a finite or infinite interval of real numbers, then it is defined as continuous.
- Indexing parameter - As stated above, the index will be regarded as time-parameter in the context of the applied stochastic processes. Similar to the space of states, if the process changes its state in discrete or finite but countable set, then the process is discrete. A discrete process with indexing parameter is called stochastic sequence. In this case, it is written as:

$$\{X_k \mid k \in N = (0, 1, 2 ......)\}$$

instead of the variable denotation {X(t)}.

On the other hand, if the process changes states in accordance with every time-coordinate, then this process is with continuous (time) parameter.

- Statistical dependence - The statistical dependence defines the relationship between certain random variable and another one within the aggregate. This is an essential difference between stochastic and other processes.

As discussed above, the stochastic modeling is based on the stochastic or random process. This is a family of random values as functions of time. Depending on the values of time (discrete or continuous), there are different types of stochastic processes. The most often modeled processes approximate the stationary random process. In this case, its probabilistic characteristics do not change with time. The description and study of a stochastic process is based on:

- Probability for a transition $p_{ij}(t) = P(s_i \rightarrow s_j)$ from state si = S(t) to another state $s_j = S(t+1)$;
- Initial probabilities $p_i(0)$ for the initial state of the process;
- Final probabilities which are constants at certain *t*=1,2,…. and determine the probability with which a process can make a transition to certain state $s_j$ at certain moment t.

Markov process is a stochastic process which shows a simple but quite useful form of dependence between random values of the same aggregate, namely, every random value has distribution depending only on the previous random value.

The implementation of the apparatus of Markov processes for the description and study of computer systems is stipulated by the fact that the functional behavior of the objects in a computer system is characterized by Markov properties. At any moment of time, the probabilistic characteristics of the transition depend on the current status of the process. They do not depend on when or how the system has come to that state. The present work does not aim to characterize or study fully some Markov process. Further, only these properties and features will be discussed which are necessary as the basis for the modeling of concrete systems.

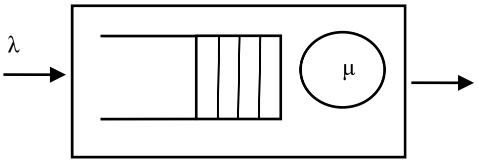


**Fig. 1 First in, first out structure**

The task of modeling is to build a model of an array of intelligent interfaces. They will be discussed as variants of implementation.

The model of each implementation should be as simple as possible, as this tendency corresponds to the tendencies in the manufacturing of highly efficient embedded devices. This is a solid prerequisite for modeling a system in one of its trivial variants (m/m/1). The number of tasks in such a system N(t) can be determined comparatively easy[1]. The number of tasks can be modeled as birth – death – process, and then the probability of an equilibrium state Pk will be determined by an equation of balance. This will allow finding N using the dependence:

$$N = \sum_{k} k \cdot P_k$$

### A. . Description of the elements and dynamics of Markov system (Discrete Markov chain with discrete time distribution)

A Markov system with tails is characterized with incoming tasks distributed according to Poisson distribution and exponential distribution of the time for each process. It can be stated that the system resides in stable state after the fading of the transition processes and efficiency dos not depend on time.

A classic system of type M/M/l is classified into systems with tails where the incoming tasks corresponds to Poisson distribution while service is done by a server system according to exponential distribution of service time. The rate (frequency) of incoming is λ and the rate (frequency) of servicing is μ. They do not depend on the number of serviced in the system, i.e. on the state. The two other parameters according to Kendall notation should be noted here: infinite capacity of a system and service discipline in a queue "First come first served"[1].

First, we will focus our attention on the number of serviced N(t) at moment t. The tasks coming to the system can be regarded as birth while tasks leaving the system – as death, within a "birth – death" – process. In accordance to Poisson distribution, two tasks cannot co-exist within a period of Δt, and the exponential distribution of the servicing time ensures that at most one task can leave the system within the interval Δt. Therefore, N(t) can be regarded as „birth – death" which can pass to the adjacent states within certain interval Δt.

Definition: The stochastic sequence {X$_k$, k є T} is called discrete if the conditional probability for each i and j is determined as follows [1]:

$$P[X_{k+1} = j \mid X_0 = i_0, X_1 = i_1, \dots . X_{k-1} = i_{k-1}, X_k = i] \\ = P[X_{k+1} = j \mid X_k = i]$$

The equation above can be interpreted as: probability distribution (k+1), depends only on the current state k (k – instance) but not on the way it was reached. All the history of the transitions in the system is summarized by the specifications of the current state. In other words, the system does not have memory. The conditional probability on the right hand side of the equation is the probability that the system can pass from state I at moment k to state j at moment (k+1). This probability will be called probability of a single transition. Generally, the probability $P[X_{k+1} = j|X_k = i]$ is a function of time [2] [3]. If it is invariable with tie, the Markov chain is determined as time-homogeneous. For the modeling, we will use time-homogeneous Markov chain. Shortly, the conditional probability can be written as:

$$p_{ij} = P[X_{k+1} = j \mid X_k = i] \qquad (1.0)$$

The set of states of the Markov chain will be denoted here as {0,1,2,3,…..}. The denotation X$_k$=i means that the chain is in state i at moment k. The following expression will be used to denote the probability that the chain is in this state:

$$\pi_i^{(k)} \equiv P[X_k = i]$$

The passing of the Markov chain from one state to another is marked as transition.

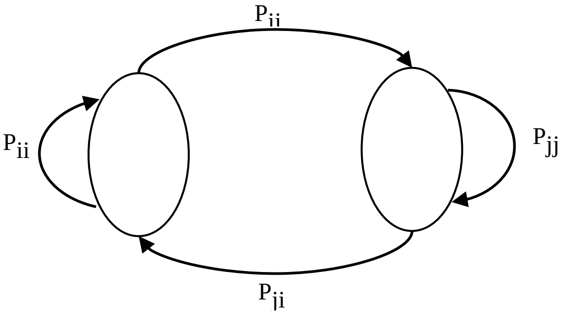


**Fig. 2 Diagram of states and transitions**

The system might pass to another state or remain in the current one. For this reason, the following equations are true:

$$\sum_{j} p_{ij} = 1$$

$$1 - p_{ii} = \sum_{j \neq i} p_{ij}$$

The conditional probability in the expression above reflects only the chain dynamics. For full characteristics of the Markov chain, it is necessary to specify the initial state – initial probability distribution P[X0=i] of the chain. Starting from the initial state, it is possible to calculate the probability of the system to pass to another state at some future moment. This can be done using the theorem for total probability [1]:

$$P[X_{k+1} = j] = \sum_{i=0}^{\infty} P[X_{k+1} = j \mid X_k = i] \cdot P[X_k = i] \qquad (1.1)$$

$$= \sum_{i=0}^{\infty} \pi_i^{(k)} \cdot p_{ij}$$

The discussion so far implies that the calculation by matrices is the more elegant way to formulate the dynamics in systems with Markov chains. To express the probabilities for transitions in a system with n possible states, a matrix P sized *n x n* can be introduced as follows:

*Published By:*
*Blue Eyes Intelligence Engineering*
*and Sciences Publication (BEIESP)*

$$\tilde{P} = (p_{ij}) = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \cdots & & p_{ij} & \\ \cdots & & & p_{nn} \end{pmatrix} \qquad (1.2)$$

The element $p_{ij}$ is the probability for transition defined by eq.(1.0) [4][5]. From the point of view of the micro-modeling of the system studied, it is assumed that it has finite number of states – n. This is the concrete case. The sum of the probabilities for transition to another state and remain in the current state is unity. Thus, the sum of the values in a row is determined as:

$$\sum_j p_{ij} = 1$$

A matrix where the sum of the elements in a row is unity and all of them are positive is called stochastic matrix. Certain Markov chain is fully defined (for single transition) by such a matrix together with the vector of the probability of the initial (current) state, called initialization vector. The probabilities for the system to take certain state from the possible n states at each moment k, can be represented as a vector [6]:

$$\pi^{(k)} = \left(\pi_0^{(k)}, \quad \pi_1^{(k)}, \qquad \pi_n^{(k)},\right)$$

Using the matrix type description, the calculation of the probability to obtain certain state can be written as follows:

$$\pi^{(1)} = \pi^{(0)} \cdot \tilde{P}$$

$$\pi^{(2)} = \pi^{(1)} \cdot \tilde{P}$$

$$\cdots$$

$$\pi^{(n)} = \pi^{(n-1)} \cdot \tilde{P}$$

Substituting $\pi^{(i)}$ in these equations, we obtain:

$$\pi^{(n)} = \pi^{(o)} \cdot \tilde{P}^{(n)}$$

where:

$$\tilde{P}^{(n)} = \tilde{P} \cdot \tilde{P}^{(n-1)} = \tilde{P}^{n}$$

Therefore, it can be written:

$$\tilde{P}^{(k+l)} = \tilde{P}^{(k)} \times \tilde{P}^{(l)}$$

or

$$P_{ij}^{k+l} = \sum_{k=0}^{n} P_{ik}^{k} \cdot P_{kj}^{l} \qquad (1.3)$$

The last expression is the well-known equation of Chapman-Kholmogorov.[1] [2].

### *B. Model with discrete states and discrete time*

Let n servicing devices D1 , D2 , D3 , ..., Dn designed for m types of tasks T1 , T2 , T3 , ..., Tm are given.

Each task Ti charges one or several of the servicing devices $D_j$ , with the service charge is an integer number of requests to the device - $w_{ij}$ .

To describe the device charges, a table of all possible charges [7] [8], will be used. Each row of the matrix corresponds to the distribution of the requests among the servicing devices.

| | D1 | D2 | D3 | D4 |
|---|---|---|---|---|
| $T_i$ | 2 | 1 | 1 | 0 |

Is interpreted as follows[9] [10]: The task $T_i$ sends two requests to the first device, one request for the second and third devices and no requests to device D4. Generaly, the table will look as follows[11]:

| | D1 | D2 | D3 | .... | Dn |
|---|---|---|---|---|---|
| T1 | w11 | w12 | w13 | | w1n |
| T2 | w21 | w22 | w23 | | w2n |
| T3 | w31 | w32 | w33 | | w3n |
| | | | | | |
| Tm | wm1 | wm2 | wm3 | | wmn |
| | | | | | |
| | | D2 | D3 | .... | Dn |

The state of the servicing system of type:

| D1 | D2 | D3 | ... | Dn |
|---|---|---|---|---|
| | | | | |

At certain moment it will be represented by the vector $(a_1, a_2, a_3, \cdots, a_n)$ , $a_i \geq 0, i = 1,2,..j$ , consisting of the number of requests to the servicing system. The coming of each different task $T_i$ is supposed to be described by a Poisson process with parameter $\lambda_i$ within the time interval $\Delta t$.

The time for servicing the requests by device $D_j$ is supposed to be an exponential process of distribution, a random process with parameter $\mu_j$.

The following matrix is assumed for full formal description of the system:

$$\begin{array}{c|cccc} T_{k_p}^{p} & w_{k_p 1} & w_{k_p 2} & \cdots & w_{k_p n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ T_{k_2}^{2} & w_{k_2 1} & w_{k_2 2} & \cdots & w_{k_2 n} \\ T_{k_1}^{1} & w_{k_1 1} & w_{k_2 2} & \cdots & w_{k_n n} \\ \cline{2-5} \overline{A} & a_1 & a_2 & \cdots & a_n \end{array}$$

Where:
A is the state of the servicing system;
$T_{k_i}^{i}, i = 1 \cdots p$ are the requests;
p is the queue length which can be 0;
$k_i$ is the type of the ith request.

The first task from the queue $-T_{k_1}^{1}$ will be accepted by the servicing system D if $a_i$=0 for these values of *i* for which $w_{k_j i} > 0$.

*Published By:*
*Blue Eyes Intelligence Engineering*
*and Sciences Publication (BEIESP)*

This condition is equivalent to: $\sum_{i=1}^{n} w_{k_1 i} a_1 = 0$.

If one or several of the service devices necessary for the first task in the queue are busy, the task must wait for them until they are ready.

The system works by the following sequence:

- A task comes or not;
- Accomplishment of part of the requests of the busy service devices;
- If possible, the servicing system D accepts the first task from the queue.

The process described complies with the conditions for a Markov process, since the next state of the system depends only on the previous state rather than on the whole history of the process.

From the point of view of the object modeled – intelligent interface devices built on the basis of embedded systems, it would be useful to consider concrete cases of simplified systems.

## III. Implementation

Before demonstrating the model, some further explanations should be given. First, a system with a single servicing device is studied. Taking into account the fact that the size of the input/output buffers is a matter of comparatively cheap hardware, no limitations will be set on the queue length. The queue length depends on the efficiency of the input/output devices but this problem has also been solved by the embedded systems. The way each of these devices work stipulates some substantial divergences from the standard considerations of some authors. This requires changes in the approach to the overall modeling which will be clarified below.

Condition for the modeling (micro modeling): the queue holds only one task

Task type is described by a table:

| | |
|---|---|
| T0 | 0 |
| T1 | 1 |

The possible system states are :

0|0, 0|1, 1|0, 1|1

The queued task is on the left, if there is any, while on the right – the request for service to the servicing device which is processed at this moment.

1|1 means that the servicing device is busy at this moment while there is a task in the queue. Since queue length is insignificant, it can be assumed that there is 1 or 0 tasks at the input at the moment.

Accepting a task from the queue (if there is one and if it is possible) by the servicing system; this is described by the transitions matrix:

| | 0\|0 | 0\|1 | 1\|0 | 1\|1 |
|---|---|---|---|---|
| 0\|0 | 1- λΔt | 0 | λΔt | 0 |
| 0\|1 | μΔt | 1- λΔt- μΔt | 0 | λΔt |
| 1\|0 | 0 | ρΔt | 1- ρΔt | 0 |
| 1\|1 | 0 | 0 | μΔt | 1-μΔt |

One of the features of the transitions matrix is that the sum of the members in a row is unity. It does not need a complex proof. The transitions matrix is stochastic and complies with all the conditions discussed in the theoretical background above. From engineering point of view, these facts can be interpreted quite simply: The total probability that the system will pass to one of the defined states or remain in the current one is equal to unity.

The numeric studies of the system with discrete states and discrete time shown above can be summarized as follows:

The test vector at the corresponding values of $\lambda\Delta t, \mu\Delta t$ and $\rho\Delta t$ is (1,0,0,0):

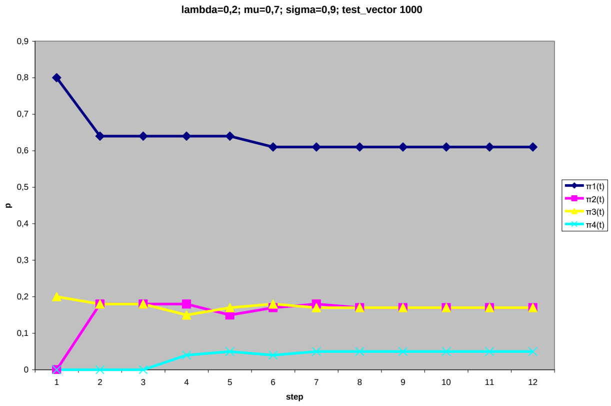
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| π1(t) | 0,8 | 0,64 | 0,64 | 0,64 | 0,64 | 0,61 | 0,61 | 0,61 | 0,61 | 0,61 | 0,61 | 0,61 |
| π2(t) | 0 | 0,18 | 0,18 | 0,18 | 0,15 | 0,17 | 0,18 | 0,17 | 0,17 | 0,17 | 0,17 | 0,17 |
| π3(t) | 0,2 | 0,18 | 0,18 | 0,15 | 0,17 | 0,18 | 0,17 | 0,17 | 0,17 | 0,17 | 0,17 | 0,17 |
| π4(t) | 0 | 0 | 0 | 0,04 | 0,05 | 0,04 | 0,05 | 0,05 | 0,05 | 0,05 | 0,05 | 0,05 |

The test vector at the corresponding values of $\lambda\Delta t, \mu\Delta t$ and $\rho\Delta t$ is (0,1,0,0)

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| π1(t) | 0,7 | 0,63 | 0,51 | 0,59 | 0,62 | 0,6 | 0,6 | 0,61 | 0,61 | 0,61 | 0,61 | 0,61 |
| π2(t) | 0,1 | 0,01 | 0,25 | 0,21 | 0,15 | 0,17 | 0,18 | 0,17 | 0,17 | 0,17 | 0,17 | 0,17 |
| π3(t) | 0 | 0,28 | 0,21 | 0,14 | 0,17 | 0,18 | 0,17 | 0,17 | 0,17 | 0,17 | 0,17 | 0,17 |
| π4(t) | 0,2 | 0,08 | 0,03 | 0,06 | 0,06 | 0,05 | 0,05 | 0,05 | 0,05 | 0,05 | 0,05 | 0,05 |

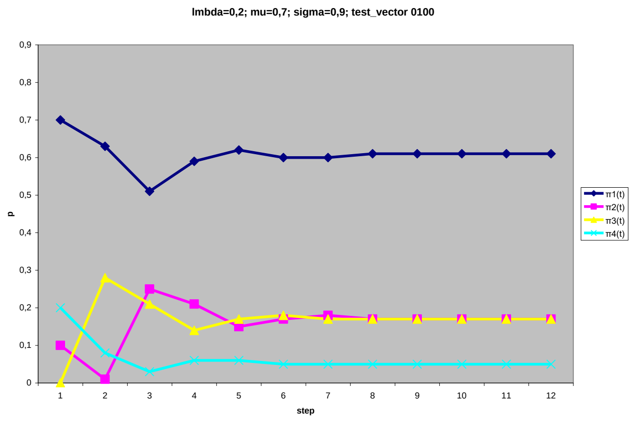
The integers 1,2,.... 12 indicate the sequence number of the iteration. Despite the value of the test vector, it can be seen that after the 7th iteration the system passed to a state where the probabilities for transition remain unchanged. The changes in transition probabilities were plotted:

- With test vector (1,0,0,0)



- With test vector (0,1,0,0)

*Published By:*
*Blue Eyes Intelligence Engineering*
*and Sciences Publication (BEIESP)*
*© Copyright: All rights reserved.*



The plots shown are randomly selected. Simulations were carried out using great variety of values of $\lambda\Delta t, \mu\Delta t$ and $\rho\Delta t$ . The possible test vectors were studied in all the cases.

At the beginning, the plots show big changes in the transition probabilities but they stabilize after the 7th iteration which shows the stabilization of the system. The simulation of the system proved its stability.

In all the cases studied, the stabilized values of the transition probabilities had the same values regardless of the test vector, i.e. the values of $\lambda\Delta t, \mu\Delta t$ and $\rho\Delta t$ were the same with the different test vectors used to investigate the system.

## IV. Conclusion and Research Directions

The studies on the stability and the values of probability for transition to certain state were simulated by a proprietary software written specially for this purpose. It should be remembered that the studies were focused on simplified systems with one servicing device which are special case within the variety of interfaces. Studies will be carried out on systems of complex structure of intelligent interfaces, as well as their interconnectivity and the results will be reported in future publications.

The formulations and studies were carried out on the basis of a project funded by Bulgarian National Science Fund – NSF Grant No D-ID-02/14 and GRAND NHT-333/14.

## References

[1] Chee-Hock N., Boon-Hee S., 2008, Queueing Modelling Fundamentals With Applications in Communication Networks, 2nd Edition, John Wiley & Sons Ltd., ISBN 978-0-470-51957-8 (HB)

[2] Bainov, D. D. and Simeonov, P. S., 1989, Systems with Impulsive Effect: Stability Theory and Applications, Ellis Horwood, Chichester, Systems of Differential Equations, Journal of Mathematical Analysis and Applications, 113 (1986), pp.562-577

[3] Gopalsamy, K. and Zhang, B., 1989, On delay differential equations with impulses, Journal of Mathematical Analysis and Applications, 139, pp.110-122

[4] Halanay, A and Wexler, D., Teoria Calitativa a Sistemelor cu Impulsuri, Editura Academiei Republicii Socialiste Romania, Bukureshti, Romania, 1968

[5] Lakshmikantham, V., Bainov, D. D. and Simeonov, P. S., 1989, Theory of Impulsive Differential Equations, World Scientific, Singapore, New Jersey, London

[6] Liu, X., 2004, Stability of impulsive control systems with time delay, Math. Computer Modeling, 39, pp. 511-519

[7] Randelovic, B., Stefanovic, L. and Dankovic, B., 2000, Numerical solutions of impulsive differential equations, Facta Universitatis, Ser. Math. Inform., 15, pp. 101-111

[8] Stamov, G. T., 2004, Impulsive cellular neural networks and almost periodicity, Proc. of the Japan Academy, 80/10, pp. 198-203

[9] Stamov, G. T and Stamova, I. M., 2007, Almost periodic solutions for impulsive neural networks with delay, Appl. Math. Modeling 31, pp. 1263-1270

[10] Stamova, I. M., 2009, Stability Analysis of Impulsive Functional Differential Equations, Walter de Gruyter, Berlin, New York,

[11] G.T. Stamov, 2012, Almost Periodic Solutions of Impulsive Differential Equations, Springer, Berlin

**Stanislav Simeonov**, Prof. in Computer Science at University "Prof. Dr Asen Zlatarov", Burgas works on computer systems, computer networks and operating systems. Education details: Master of Science and PhD in Electronic (Techical University Chemnitz, Germany).

**Neli Simeonova**, PhD of Mechanical engineering works on systems of automation, control and robotics. Her research interests include kinematic analysis and kinematic computer control of tracked mobile robot and technologies for helping visually impaired people.
Education details: Master of Science and PhD in Engineering (Technical University, Chemnitz, Germany)