V. M. Kichak, Dr. Sc. (Eng.), Prof.; M. V. Vasilskaya

DEVELOPMENT OF THE GENERALIZED SIMULATION MODEL OF A MOBILE COMMUNICATION SYSTEM ON THE BASIS OF AGGREGATE APPROACH

Elaboration of a generalized, open for modifications working mathematical model of the multichannel queue system with arbitrary servicing disciplines is proposed and substantiated. Functional subsystems of the imitation model are built as aggragates. The model is designed for simulation of the processes in mobile communication systems.

Key words: *aggregate, aggregate system, queueing, simulation model, working model, probability distribution, uncertainties, vectorization, multichannel system.*

Problem statement. Today mobile communication systems constitute not only an essential sector of industry and technologies but also an integral component of all production organization systems. Therefore, problems of forecasting and planning of mobile communication systems development are, undoubtfully, of great current importance. High rate of innovations, high uncertainty levels, absence of statistics determine the necessity of mathematical models application for the operation of such systems. One of the important problems in modeling of the mobile communication systems is obtaining the functions of external and internal factors influence on the efficiency of mobile communication systems. The existing methods for calculation of the queues and capacity of mobile communication systems require the knowledge about theoretical frequency distribution for the intervals and volumes of incoming requests (tasks) and about the servicing period duration.

Formulation of the task. We combine the stages of the mathematical model and simulation program development into the working models development stage. Working model is a product of modern software packages for modeling in general and specialized packages for modeling of mobile communication systems. Working model is, in fact, a user interface with package software modules written in C++. Visual representation of the working model corresponds, quite accurately, to the standards of scientific and, particularly, mathematical publications.

Substantiation of the method for influence functions determination

In this work a basic model of the operation of a mobile communication system as of a multichannel queue system (QS) with arbitrary and variale servicing disciplines is chosen. Within the range of the known analytical QS models it is impossible to reflect adequately the specific features and complexity of mobile communication systems. The procedure of such simulation model elaboration was chosen that would meet the requirements of computational efficiency, modularity and adaptability to modifications. The available Matlab, Mathcad, VisSim mathematical software packages were chosen as the model realization means.

At present, an efficient approach to the simulation of complex multichannel systems is division of QS model into aggregates and construction of aggregate models. An aggregate is a generalized element from the theory of dynamic systems. The essence of generalization consists in the folowing: an aggregate can include continuous static and dynamic models, pulse, relay, logic-dynamic models, models of finite automatons as well as models of uncertainty and random disturbances. In particular, those could be classic probability models based on fuzzy logic. The aggregate has inputs and outputs, it represents a universal converter of information.

The model of a complex system, assembled from aggregates, i.e. the aggregate system model, is built using the methods of aggregation and decomposition. In the environment of mathematical software packages an aggregate can be defined as a user function that takes the vectors of the aggregate and of input variables parameters and returns the vector of output variables. Efficient realization of the optimization function, the function of zero roots search and of the balance states is

also possible.

After aggregates and aggregate models are defined and realized, they can be operated as ordinary analytical functions (e.g., sin(), arcsin(), ln()) – they can be integrated, extremums and zero points as well as derivatives with certain constraints are found, i.e. when the simulation model is built, it enables simple detrmination of functional relationships of efficiency indices.

Exept model decomposition into aggregate elements, decomposition of a complex simulation model in time is used: the model creation is not a single-step, but a multistep process. At each step the existing model becomes a little bit more complex, it is refined and undergoes comprehensive testing. Using the model, research is also conducted in accordance with the target. Practically, stepwise model creation makes it possible to exclude errors and, which is the main thing, to receive new knowledge about the modeling object properties, to adjust the imitation model development guidelines.

When a successive model is being developed, the preceding one remains as a means for controlling more complex models and also as a simplified model for simple problems. On the basis of the developed model, simulation models for definite mobile communication systems can be created.

Defining the basic notions

Aggregate models. There are no unified standard model classifications in the literature on modeling. There also does not exist a single definition of aggregate models and aggregate systems. Therefore, these definitions will be refined and concretized in accordance with the objects of research and modeling – mobile communication systems. We shall consider a class of complex systems that are aggregate constructions and have the following properties: there exist such (and usually not a single one) system division into the elements, where each of the obtained elements is an aggregate.

Definition of an aggregate in the theory of systems. The most simple definition of an aggregate is as follows. T set of time moments and X, U, B, Z sets of arbitrary nature are given. The elements of these sets are interpreted in the following way: $t \in T$ – time moment; $x \in X$ – input signal; $u \in U$ – control signal; $y \in Y$ – output signal; $z \in Z$ – an aggregate state. States, inputs, outputs, control are time functions $z(t)$, $x(t)$, $u(t)$ и $y(t)$.

An aggregate is defined as the object $<T, X, U, B, Z, H, G>$, where H, G operators could be random, fuzzy ones. Outputs of transition operators and H, G outputs are time functions $z(t)$, $x(t)$, $u(t)$ and $y(t)$. Aggregates differ from finite automatons and ordinary dynamic systems in the structure of these operators.

Transition and output operators. There exist two approaches in the "aggregate simulation models" trend – the abstract approach, where operator classes with "special states" are considered using artificial examples without substantial interpretations, and the approach aimed at the creation efficient simulation of modeling systems for research and integration into automatic control systems. The sequence of research stages in the latter approach is as follows: creation of an operative models, conducting intensive research on the model and then theoretical generalization for creation of the next-generation simulation models. Neither real objects, nor the models must have "special states" of temporary or ultimate inoperability. This is the approach that we have chosen.

A process of an aggregate operation. An aggregate is a mathematical sheme of a generalized form, the particular cases of which are logic algebra functions, relay-contact networks, final automatons, all classes of queueing systems, logic and logic-dynamic systems, described by ordinary diferential equations, and some other objects. In terms of modeling an aggregate is a universal transformer that recieves input control signals and generates output signals. In this work a set of mathematical aggregates is extended with simulation models of complex random events and fuzzy logic models.

Aggregate systems. Let's consider a class of complex systems that are constructions consisting of aggregates that possess the following property: there exist such system division into elements (in

general case an ambiguous one), where each of the obtained elements is an aggregate. Complex systems of such type will be referred to as aggregate systems or A-systems. It is desirable, but not obligatory for the real system, represented by an aggregate model, to have specialized aggregates for information exchange with external environment. Accordingly, all information circulating in A systems is divided into external and internal information.

For each aggregate conditions of the information transmission are described – delays, requested transmission, the existence of noise and disturbances. As to the aggregates structure, all of them are described by the theory of graphs. The properties of A system are described not only by its properies, but also by its structure. Aggregate connections are interesting for us only in the aspect of certain structures replacement by equivalent aggregates. Naturally, we distinguish sequential, parallel combinations and cycles – inverse connections. This statement can be easily verified at the stage of program realization – by integration of program modules for the aggregates from which the structures are composed. This statement can be substantiated theoretically, but it envolves a large amount of procedures for formalization of aggregate operation scenario – delays, state changes, disturbances etc.

As interpretation of theoretical statements, a basic aggregate QS model will be built.

Choosing a technology for a working model realization

We choose such technology of the basic model development, that is maximally coordinated with the software development technology – for QS description an approach based on the "state vector" is chosen – the state vector structure that is a compromise between convinience, system description accuracy and dimensionality is selected – hierarchical decomposition of the model into small functional modules is performed, each module is thoroughly tested before it is included into the main program – behavior simulation modules are formed as operators of the current state transformation into the next one; – in the transition from one-dimensional systems to multidimensional ones vectorization of computations is used [3].

As generalization the following sequence of models is adopted:

– $x_{k+1} = A \cdot x_k + B \cdot u_k$ – a linear system, the objects are state vectors;

– $Mu_{k+1} = Mu_{k+1} + Nast\left(Mu_k, dMu_k\right)$ – a non-linear system, the objects are state matrices;

– $Mu_{k+1} = Op\left(M_k, U_k, P\right)$ – a non-lenear-system, the objects are arbitrary array structures - control and parameters.

Arbitrary array structure is a matrix with the elements being arbitrary array patterns: scalars with numerical and character variables. Into the simulation models of mobile communication systems user behavior models must be included. As an example of vectorization, a module of fuzzy choice with learning will be considered. It takes a scalar parameter of some element and returns the scalar parameter of choice probability. A scalar module is represented first.

$$dmu(vyb, bolv, nvz) = \left| \begin{array}{l} kys \leftarrow vyb \cdot nvz \\ qq \leftarrow vyb \cdot bolv + rnd(kys) - 0.5 \cdot kys \\ qq \end{array} \right. \quad (1)$$

Module (1) is applied to processing of the matrix object arrays – the arrays of the parameters of these consumers (the statitistics of using mobile communication services for each consumer). At the output we also receive the output data array:

$$dMU = \overrightarrow{dmu(Vyb, Bolv, Nvz)}. \quad (2)$$

Development of QS aggregate modules

Two working goals of the development are set: creation of the open library of modules for different service procedures of the service requests flow, creation of the open library of the

uncertainty simulation modules. In order to be suitable for present-day conditions, service algorithms should be simple – "natural", decentralizing, fault-tolerant, adaptable to the input tasks flow statistics.

"Probability distribution choice" module. The module is based on the functions that generate the arrays of numbers with given probability distribution. Parametric adjustment of these functions for the statistics of real input data is provided. Fig. 1 presents an example of the uncertainty library realization.

$$kk := 3 \qquad Fvx(kk) := \begin{cases} rbinom(1,8,0.9) & if\ kk = 1 \\ rpois(x,\lambda) & if\ kk = 2 \\ rnorm(1,6,1) & if\ kk = 3 \end{cases} \qquad \begin{aligned} Fvx(1) &\rightarrow rbinom(1,8,.9) \\ Fvx(2) &\rightarrow rpois(x,\lambda) \\ Fvx(3) &\rightarrow rnorm(1,6,1) \end{aligned}$$

Fig. 1. Uncertainty library (a part)

Uncertainty simulation module. Fig. 2 shows an example of the service requests flow simulation module with three random parameters: quantity, servicing volume (durability) and priorities.

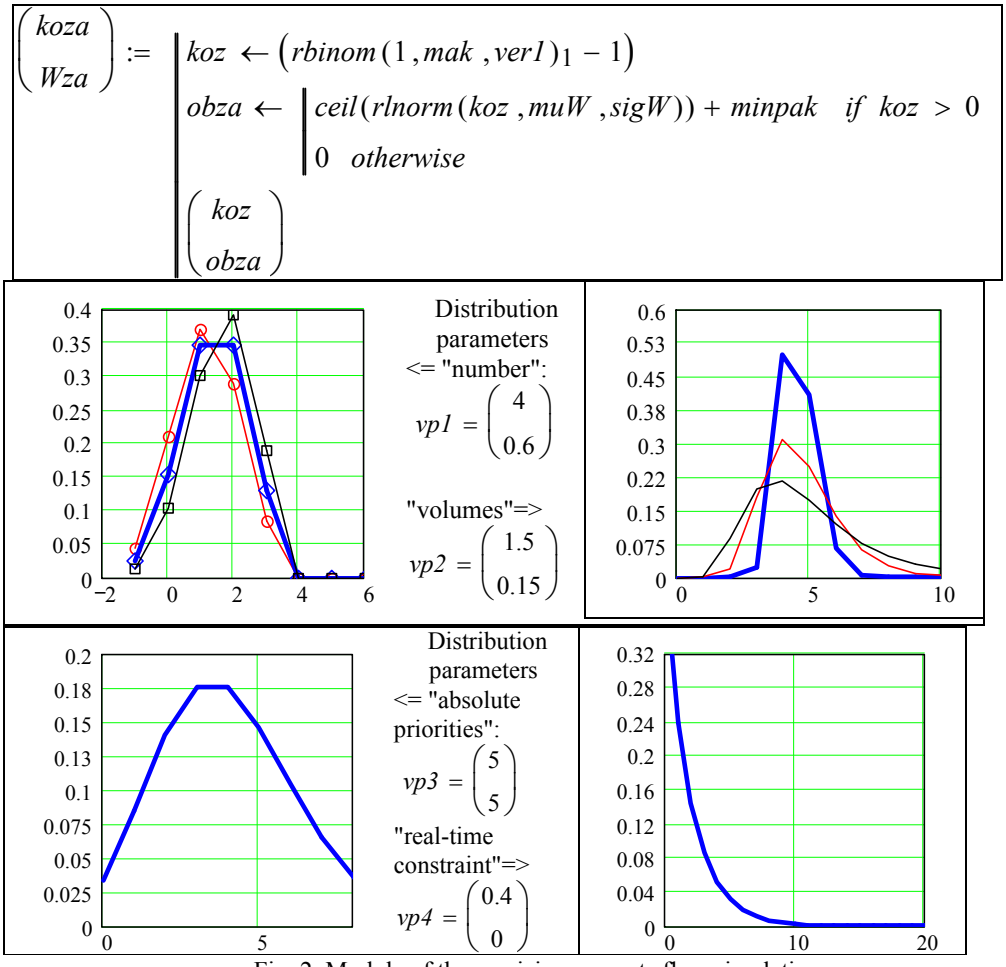

Fig. 2. Module of the servicing requests flow simulation

Classical literature on QS gives detailed classifictions according to the types of input tasks flow probability distribution and processing duration. This is explained by the fact that the search for analytical solutions was performed individually for each type of distribution. The created module of the input tasks flow generation can be adjusted to arbitrary probability distributions.

Queue state and priorities generation module

We shall solve radically the problem of choosing the optimal queue task priority control: a priority is formed as a weighted sum of priorities, determined with different algorithms that are usually used in computation systems [4].

$$Ps_i = \sum_{p=1}^{Np} \alpha_p \cdot prior_{i,p}, \tag{3}$$

where α_p is a weighting coefficient according to p-priority index $prior_{i,p}$ - priority of i task according to p priority index. "Queue with priorities" data structure consists from input data and the data to be calculated – priorities. The queue is represented by a matrix where:

- the first column – "natural queue" indices;
- the second column – task volumes in natural order;
- the third column – waiting in queue (steps, cycles);
- the fourth column – real time constraint (maximal processing delay);
- the fifth column – **absolute priorities** that are interpreted as the importance of the task.

Columns 6, 7, 8 contain **relative priorities** that are calculated and can be changed while waiting in queue:

6 – the task size priority: the smaller is the task size, the higher the priority is;

7 – the priority of waiting in queue: the longer is the process of waiting, the higher the priority is;

8 – the real time priority: the shorter is the period before the request is processed, the higher the task priority is.

The last column 9 contains a total weighting priority that is calculated using (3).

Weighted compression (3) has one more purpose: after zeroing all coefficients exept one, QS with one definite priority can be realized and investigated "in pure form". Availability of the "priorities library" forms the knowledge base for adaptive systems construction.

Dynamics of priorities. Certain data in *Ps0* matrix depend on others or are changing in time in accordance with definite mechanisms. We classify these changes in the following way [5]:

1) disappearance of the queue rows as a result of task processing according to priorities;

2) emergence of new rows when new tasks with corresponding priorities arrive;

3) recalculation of the priorities at each step of the process: increasing the period of waiting in queue, rising the priorities for real-time tasks.

Finally, we must receive the operator of transition from current to the next states of the "queue with priorities" structure. First, we create modules that describe transformation of data structure separate elements – columns and rows.

The module of priorities calculation according to the size of the tasks takes the second column $Ps0^{\langle 2 \rangle}$ (task volume) of the data structure and returns the normalized vector of priorities.

The module of the weighted-mean priority calculation takes 5-8 columns of *Ps0* matrix and returns the weighted-mean vector of priorities that is located in the last column of matrix *Ps0*.

Operation of the modules, considered above, results in 6 – 9 columns generation. As a result, a structure is generated that is convinient for program realization of the wide class of algorithms for controlling the task flow handling process.

The queue processing module. Apart from the function of finding priorities and choosing the task to be processed, the aggregate queue processing model must update the queue state matrix at each step of processing. This could be done in different ways: by zeroing, cutting off or ignoring the processed rows, by their sorting according to different variables with subsequent inverse sorting. As a rule, the worst variant is chosen unless going in " module with less than 7 rows" steps without checking each of the steps. Several alternatives have been tested and the last of them has been justified experimentally. Linguistic model of the queue completion and processing is as follows:

- new tasks are arriving at each step (output of the task flow simulation module);
- at each step the task queue is processed within the carrying capacity;

The following queue processing situations are possible:

1) curent queue = 0; ==> next queue = 0;

2) current queue ≤ of the capacity range ==> next queue = 0;

3) the capacity < first task volume ==> remainder of the first task will still be processed at the next step;

The end goal of the simulation program development is to create such structure and such operator (user function) that the system dynamics would be described by the following difference equation of the system state transformation:

$$ss^{\langle k+1\rangle} = YpurP\left(ss^{\langle k\rangle}\right), \tag{4}$$

$$\text{or } ss^{\langle k+1\rangle} = YpurP1\left(ss^{\langle k\rangle}, u\left(ss^{\langle k\rangle}\right)\right). \tag{5}$$

Queue processing module takes QS capacity and the vector of task queue, sorted according to the chosen priorities, and returns a new queue vector (this is reflected in the definitions ss3, S3 – definition of the corresponding of QS state structure component in the main working model). Verbal description of the queue handling process:

- for the case of a zero queue "a stopper" is made: $nulka \leftarrow (0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0)$;
- a number of requests $Okol$ and the queue volume $Odlin$ are determined;
- queue state matrix is sorted according to the chosen priority (u variable);
- "non-zero queue" condition is generated.

Fig. 3 presents the text of the module that implements this verbal description.

$$\begin{array}{|l|l|}
\hline
smHP(props, ss3, u) := & nulka \leftarrow (0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0) \\
& Okol \leftarrow rows(ss3) \\
& srti \leftarrow csort(ss3, u) \\
& Odlin \leftarrow \sum_{i=1}^{Okol} srti_{i,2} \\
& yslo \leftarrow Odlin > props \\
& S3 \leftarrow \left| \begin{array}{ll} sjelo(props, srti, u) & if\ \ yslo \\ nulka & otherwise \end{array} \right. \\
& S3 \\
\hline
\end{array}$$

Fig. 3. "Queue processing module" aggregate model

In a similar way aggregate models for the following functional modules were developed.

Queue completion module, that recieves new tasks into the queue, calculates the value of relative tasks priority as well as dynamic indices of the queue tasks: waiting in queue, critical time for real-time tasks.

The module of priorities dynamics that traces and calculates the changes, relative priorities.

"Transition operator for priority-servicing QS" module. This module uses the previous aggregate modules. Their connection sheme is presented in fig. 4. Verbal description of the process of transition from the previous to the next state:

- the result of the previous queue $S30 \leftarrow smH\left(props, ss_3\right)$ processing is determined;

- at each step the task queue is processed within the capacity limits;
- the length $S4 \leftarrow length(S30)$ and the volume $S5 \leftarrow mean(S30) \cdot S4$ of the queue after processing are determined;
- using the "queue completion" module: $BY \leftarrow koWza(vp1, vp2)$ the number $S1 \leftarrow BY_1$ and the volume $S2 \leftarrow BY_2$ of new tasks, that have come within a step of the process, are determined;
- a queue for the next step $S31 \leftarrow stack(S30, S2)$ is generated (new tasks are placed at the "tail" of the queue);
- the output is generated – the next state vector of the computing system.

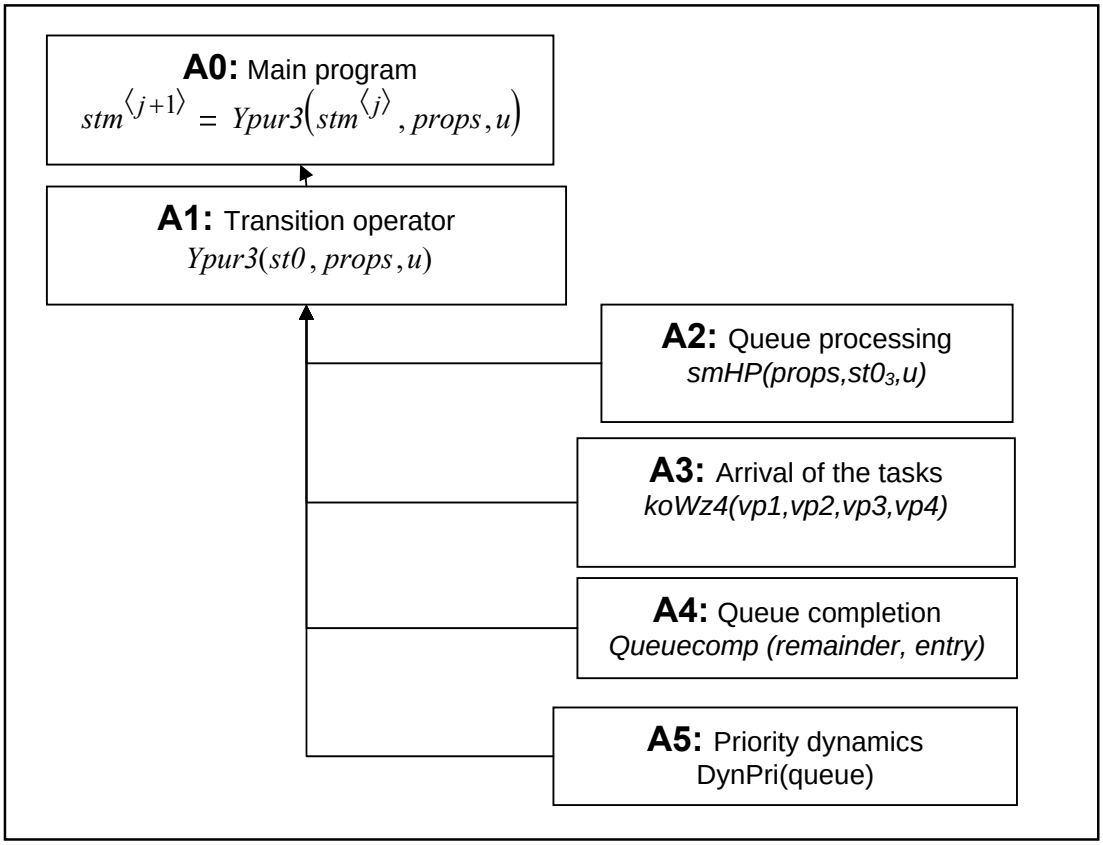

Рис. 4.Structure of the "queue system with arbitrary priorities and arbitrary servicing parameters distribution" aggregate model

There was an alternative for this problem – simulation program created using the traditional technology. It has not been proven due to the comlexity of QS operation in general.

Aggregate simulation model was adjusted to the functional state and an extensive research was conducted with it. Fig. 5 and 6 show two examples of the simulation of servicing processes with different priorities.

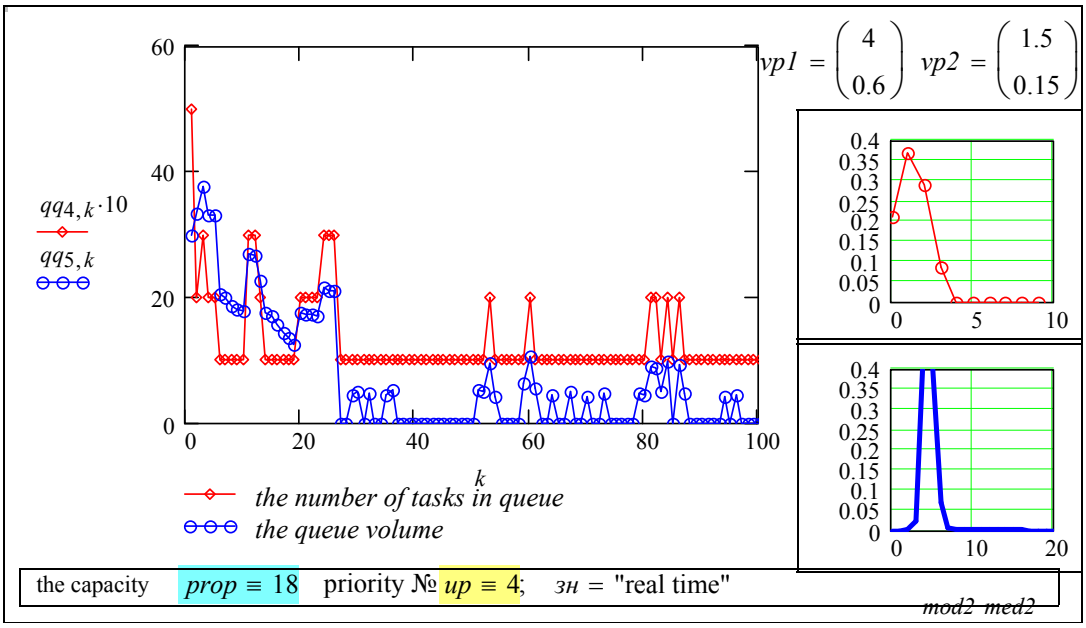

Fig. 5 An example of simulation results. Process for the real-time servicing priority

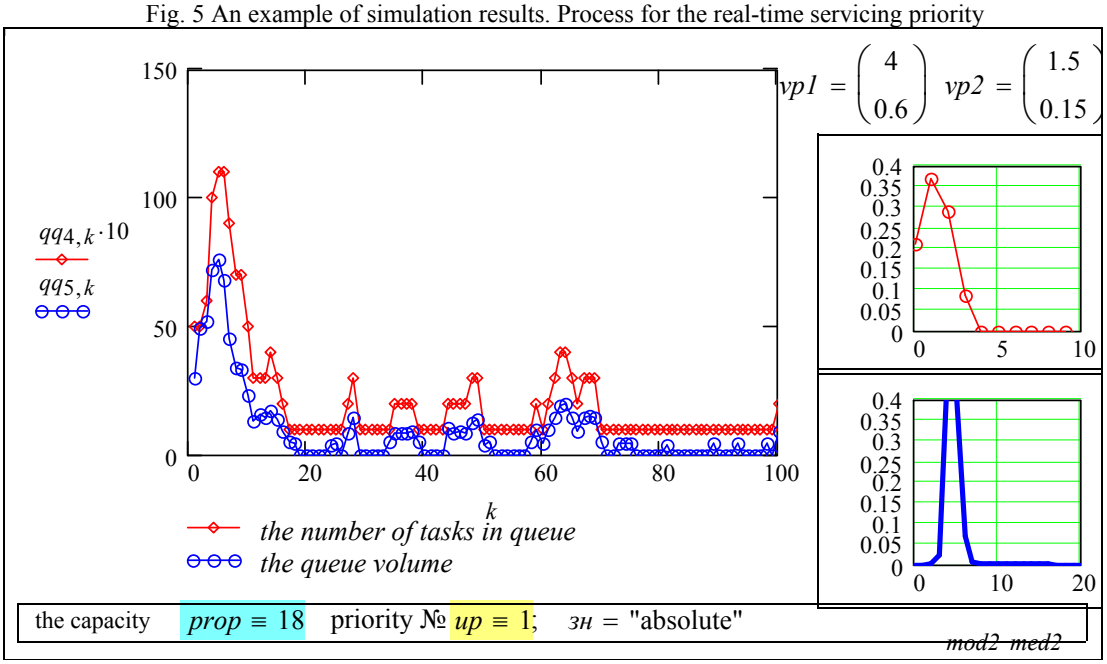

Fig. 6. An example of simulation results. Process for the “absolute” servicing priority

Practical importance of the model development and reseach consists in defining the quidelines of data collection in real mobile communication systems for adjustment and modification of the given simulation model and creation of a predictor model for real systems.

Theoretical importance – defining the guidelines for theoretical substantiation of the revealed queue system properties, generalization of the queue system theoretical models. Determination and analysis of the virtual reality statistics and obtaining the influence functions is considered to be a prospective trend.

REFERENCES

1. Нэгл Томас Т. Стратегия и тактика ценообразования. Руководство для принятия решений, приносящих прибыль / Томас Т. Нэгл. — М.: Питер, 2001. — 375 с.

2. Kelly K. New Rules for the New Economy. 10 radical strategies for a connected world. — Penguin books, 1999. — 180 p.

3. Форрестер Дж. Основы кибернетики предприятия / Дж. Форрестер. – М.: Прогресс, 1971. – 340 с.

4. Боровська Т. М. Основи теорії управління та дослідження операцій. Навчальний посібник / Т. М. Боровська, І. С. Колесник, В. А. Северілов. – Вінниця: УНІВЕРСУМ-Вінниця, 2008. – 242 с. – ISBN 978-966-641-275-4.

5. Моделювання та оптимізація у менеджменті: Навчальний посібник / Т. М. Боровська, В. А. Северілов, С. П. Бадьора, І. С. Колесник. – Вінниця: УНІВЕРСУМ-Вінниця, 2009. – 145 с. – ISBN 978-966-641-287-7.

Vasiliy Kichak – Dr. Sc. (Eng.), Prof. of the Department of telecommunication systems and television, the Institute of radio engineering, communication and instrument engineering.

Maya Vasilskaya – Post-graduate student of the Department of telecommunication systems and television, the Institute of radio engineering, communication and instrument engineering.

Vinnytsia National Technical Universitry.