Synchronized Swarm Operation

Eugene Larkin[1](✉), Tatyana Akimenko[1], and Aleksandr Privalov[2]

[1] Tula State University, Tula 300012, Russia
elarkin@mail.ru
[2] Tula State Lev Tolstoy Pedagogical University, Tula 300026, Russia
privalov.61@mail.ru

Abstract. Physical swarm system, including number of units, operated in physical time according to corporative algorithm, is considered. It is shown, that for proper corporative algorithm interpretation it is necessary to synchronize computational processes in units. Structural-parametric model of synchronized swarm operation, based on Petri-Markov nets apparatus, is worked out. In the Petri-Markov net transitions are abstract analogues of synchronization procedure, while places simulate corporative algorithm parts interpretation by swarm units. Primary Petri-Markov model is transformed into complex semi-Markov process. Formulae for calculation of stochastic and time characteristics of the process are obtained. It is shown, that after transformation all methods of ordinary semi-Markov processes investigation may be used for synchronized systems. With use the concept of distributed forfeit effectiveness of synchronization is evaluated.

Keywords: Swarm · Unit · Corporative algorithm · Semi-Markov process · Petri-Markov net · Time characteristics · Stochastic characteristics · Distributed forfeit effectiveness

1 Introduction

Physical swarms, which solve corporative task, are widely used in different branches of human activity, industrial and mobile robotics, concurrent computation, control systems, etc. [1–5]. Such systems include number of units, each of which operates accordingly to its own algorithm realized on Von Neumann type controllers. Due to consecutive interpretation of algorithm operators and accidental character of data processed, runtime of controller is a random value, and outcome of program operation is stochastic [6]. So for proper operation, when solving a corporative task, swarm should be tuned in such a way, that corporative algorithm should be divided on pieces, which are realized on swarm units, and interpretation of algorithm pieces should be carried out in the proper sequence [7]. Such alignment is called synchronization. For optimal synchronization adequate model of parallel process should be worked out. Below approach to simulation, based on Petri-Markov nets [8], which from one side permits to evaluate random time intervals characteristics, and from other side take into account interaction of parallel processes, is used. Also for optimal synchronization it is necessary to have criterion, which permits to evaluate effectiveness of corporative algorithm division. Below universal criterion,

Y. Tan et al. (Eds.): ICSI 2020, LNCS 12145, pp. 15–24, 2020.
https://doi.org/10.1007/978-3-030-53956-6_2

called distributed forfeit, is proposed, and formulae for effectiveness estimation with use proposed criterion are obtained.

Approaches to simulation of synchronized operation of swarm are currently known insufficiently, that explains necessity and relevance of the investigations in this domain.

2 Petri-Markov Model of Synchronized Operation

Operation of swarm, which includes M units and solves some corporative task, may be described with use Petri-Markov net (PMN) apparatus [8]. Swarm operation model is as follows (Fig. 1).

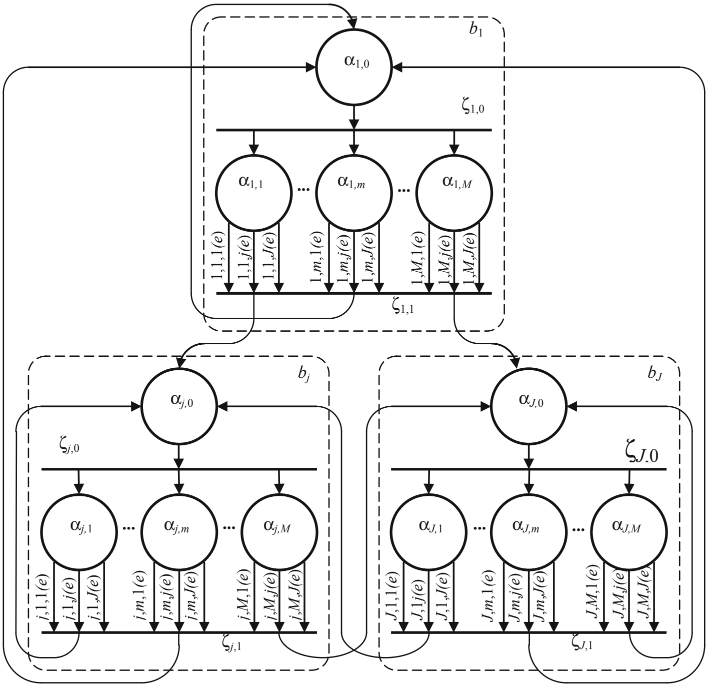

Fig. 1. Petri-Markov network describing the synchronous operation of equipment.

$$\Pi = \{\mathrm{A}, \mathrm{Z}, \iota(\mathrm{Z}), \mathrm{o}(\mathrm{Z})\}, \tag{1}$$

where $\mathrm{A} = \left\{\mathrm{A}_1, \ldots, \mathrm{A}_j, \ldots, \mathrm{A}_J\right\}$ is the set op places, which describes operation of M swarm units; $\mathrm{Z} = \left\{\mathrm{Z}_1, \ldots, \mathrm{Z}_j, \ldots, \mathrm{Z}_J\right\}$ is the set of transitions, which describes synchronization procedures; $\iota(\mathrm{Z})$ $\mathrm{o}(\mathrm{Z})$ are input and output functions of transitions, correspondingly; J number of operators in corporative algorithm of swarm behavior when solving corporative task;

$$\mathrm{A}_j = \left\{\alpha_{j,0}, \alpha_{j,1}, \ldots, \alpha_{j,m}, \ldots, \alpha_{j,M}\right\}, \quad 1 \le j \le J; \tag{2}$$

$$\mathrm{Z}_j = \left\{\zeta_{j,0}, \zeta_{j,1}\right\}, \quad 1 \le j \le J; \tag{3}$$

$$\begin{cases} \iota(\mathrm{Z}) == \left\{\iota(\mathrm{Z}_1), \ldots, \iota\left(\mathrm{Z}_j\right), \ldots, \iota(\mathrm{Z}_J)\right\}; \\ \iota\left(\mathrm{Z}_j\right) = \left\{\iota\left(\zeta_{j,0}\right), \iota\left(\zeta_{j,1}\right)\right\}, 1 \le j \le J; \\ \iota\left(\zeta_{j,0}\right) = \alpha_{j,0}; \\ \iota\left(\zeta_{j,1}\right) = \left\{\alpha_{j,1}, \ldots, \alpha_{j,m}, \ldots, \alpha_{j,M}\right\}; \end{cases} \tag{4}$$

$$\begin{cases} \mathrm{o}(\mathrm{Z}) == \left\{\mathrm{o}(\mathrm{Z}_1), \ldots, \mathrm{o}\left(\mathrm{Z}_j\right), \ldots, \mathrm{o}(\mathrm{Z}_J)\right\}; \\ \mathrm{o}\left(\mathrm{Z}_j\right) = \left\{\mathrm{o}\left(\zeta_{j,0}\right), \mathrm{o}\left(\zeta_{j,1}\right)\right\}, 1 \le j \le J; \\ \mathrm{o}\left(\zeta_{j,0}\right) = \left\{\alpha_{j,1}, \ldots, \alpha_{j,m}, \ldots, \alpha_{j,M}\right\}; \\ \mathrm{o}\left(\zeta_{j,1}\right) = \left\{\alpha_{1,0}, \ldots, \alpha_{j,0}, \ldots, \alpha_{J,0}\right\}. \end{cases} \tag{5}$$

PMN operation may be considered as sequence semi-steps, which may be done either from places to transitions, $\left(\alpha_{j,0}, \zeta_{j,0}\right)$, $\left(\alpha_{j,m,j(e)}, \zeta_{j,1}\right)$, $1 \le j \le J$, $1 \le m \le M$, $1 \le j(e) \le J(e)$, or from transitions to places, $\left(\zeta_{j,0}, \alpha_{j,m}\right)$, $1 \le m \le M$, $1 \le j \le J$; $\left(\zeta_{j,1}, \alpha_{1,0}\right)$, …, $\left(\zeta_{j,1}, \alpha_{l,0}\right)$, …, $\left(\zeta_{j,1}, \alpha_{J,0}\right)$, $1 \le j \le J$. For semi-step execution from places $\alpha_{j,0}, \alpha_{j,1}, \ldots, \alpha_{j,m}, \ldots, \alpha_{j,M}$ into corresponding transitions random time interval t should be spent, which begins from the moment, when semi-step was done into this place. Time intervals are determined with an accuracy to time densities $f_{j,0}(t)$,$f_{j,m,j(e)}(t)$, $1 \le j \le J$, $1 \le m \le M$, $1 \le j(e) \le J(e)$. For semi-step execution from transition $\zeta_{j,0}$ simultaneously to all places $\alpha_{j,1}, \ldots, \alpha_{j,m}, \ldots, \alpha_{j,M}$, constituting its output function $\mathrm{o}\left(\zeta_{j,0}\right)$ only semi-step $\left(\alpha_{j,0}, \zeta_{j,0}\right)$ should be done. For execution of one of semi-step from the transition $\zeta_{j,1}$, the proper $\left(\alpha_{j,m,j(e)}, \zeta_{j,1}\right)$ semi-steps combination to named transition must be done, due to only one direction of the set $\{1(e), \ldots, j(e), \ldots, J(e)\}$ may be choose for doing semi-step (Fig. 1).

In such a way, transitions $\zeta_{j,0}$, and $\zeta_{j,1}$, are the synchronized one: transition $\zeta_{j,0}$ in the sense that all semi-steps, included into its output function $\mathrm{o}\left(\zeta_{j,0}\right)$, are executed simultaneously (synchronous start), but transition $\zeta_{j,1}$ in the sense, that semi-step from it would not be done until all semi-steps from places of $\iota\left(\zeta_{j,1}\right)$ will be done.

PMN timing elements are places. Time of residence PMN at places $\alpha_{1,0}, \ldots, \alpha_{j,0}, \ldots, \alpha_{J,0}$ is as follows

$$f_{j,0}(t) = \delta(t), \tag{6}$$

where $\delta(t)$ is the Dirac δ-function.

Time of residence PMN at places $\alpha_{1,1}, \ldots, \alpha_{1,m}, \ldots, \alpha_{1,M}$, …, $\alpha_{j,1}, \ldots, \alpha_{j,m}, \ldots, \alpha_{j,M}$, …, $\alpha_{J,1}, \ldots, \alpha_{J,m}, \ldots, \alpha_{J,M}$ is defined as the time of wandering through semi-Markov processes [9, 10] $\mu_{j,m}(t)$, $1 \le j \le J$, $1 \le m \le M$, which are abstract analogues of swarm unit onboard computers operation, and are defined as follows (Fig. 2):

$$\mu_{j,m} = \left\{A_{j,m}, \boldsymbol{r}_{j,m}, \boldsymbol{h}_{j,m}(t)\right\}, \tag{7}$$

where $A_{j,m}$, $\left|A_{j,m}\right| = J(a) + 1$, is the set of states, which are abstract analogues of swarm unit onboard computer algorithm operators; $\boldsymbol{r}_{j,m}$ is the $[J(a) + 1] \times [J(a) + 1]$ adjacency

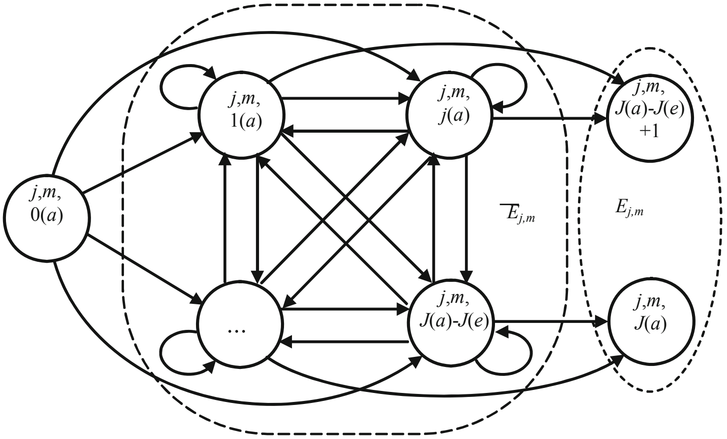

Fig. 2. Semi-Markov process $\mu_{j,m}(t)$

matrix, which describes links between operators; $\boldsymbol{h}_{j,m}(t)$ is the $[J(a) + 1] \times [J(a) + 1]$ semi-Markov matrix, which define time intervals of operators interpretation;

$$A_{j,m} = \left\{a_{j,m,0(a)}, a_{j,m,1(a)}, \ldots, a_{j,m,J(a)-J(e)}, \ldots, a_{j,m,J(a)-J(e)}\right\}; \tag{8}$$

$$\boldsymbol{r}_{j,m} = \left[r_{j,m,j(a),n(a)}\right]; \tag{9}$$

$$\boldsymbol{h}_{j,m}(t) = \left[h_{j,m,j(a),n(a)}(t)\right]. \tag{10}$$

The set $A_{j,m}$ is divided onto two disjoint subsets, subset of non-absorbing states

$$\bar{E}_{j,m} = \left\{a_{j,m,0(a)}, a_{j,m,1(a)}, \ldots, a_{j,m,j(a)}, \ldots, a_{j,m,J(a)-J(e)}\right\}, \tag{11}$$

and subset of absorbing states $E_{j,m}$.

$$\bar{E}_{j,m} = \left\{a_{j,m,J(a)-J(e)+1},\ \ldots,\ a_{j,m,j(e)},\ \ldots,\ a_{j,m,J(a)}\right\}. \tag{12}$$

Wandering through states of semi-Markov process $\boldsymbol{h}_{j,m}(t)$ start at the state $a_{j,m,0(a)}$, which is the abstract analogue of "begin" operator. States $a_{j,m,j(e)} \in \bar{E}_{j,m}$ are the abstract analogues of "end" operators for different outcomes of algorithm operation.

Element $h_{j,m,j(a),n(a)}(t) \in \boldsymbol{h}_{j,m}(t)$ performs weighted time density of m-th swarm unit residence in the state $a_{j,m,j(a)}$ when decision was made about next switch into the state $a_{j,m,n(a)}$.

Weighted time density of the semi-Markov process $\mu_{j,m}$ wandering from the state $a_{j,m,0(a)}$ till the state $a_{j,m,J(a)-J(e)+j(e)} \in E$ is as follows:

$$h_{j,m,j(e)}(t) = {}^{r}\boldsymbol{I}_{0(a)} \cdot L^{-1}\left[\sum_{k=1}^{\infty}\left\{L\left[\boldsymbol{h}_{j,m}(t)\right]\right\}^{k}\right] \cdot {}^{c}\boldsymbol{I}_{J(a)-J(e)+j(e)}, \tag{13}$$

where ${}^{R}\boldsymbol{I}_{0(a)}$ is the row vector of size $[J(a)+1]$, whose $0(a)$-th element is equal to one, and all other elements are equal to zeros; ${}^{c}\boldsymbol{I}_{J(a)-J(e)+j(e)}$ is column vector, whose $\left[J(a)-J(e)+j(e)\right]$-th element is equal to one, and all other elements are equal to zeros; $L[\ldots]$ и $L^{-1}[\ldots]$ are correspondingly direct and inverse Laplace transforms.

When $J(e)=1$ the algorithm simulated has the only outcome, and consequently $h_{j,m,J(e)}(t)=f_{j,m,J(e)}(t)$. When $J(e)>1$, then semi-Markov process $\mu_{j,m}$ gets subset E, veraciously, but the state $a_{j,m,J(a)-J(e)+j(e)}$ it gets with probability [12]

$$p_{j,m,j(e)}=\int\limits_{0}^{\infty} h_{j,m,j(e)}(t)dt. \tag{14}$$

Pure (non-weighted) time density, expectation and dispersion are equal, correspondingly [13]

$$f_{j,m,j(e)}(t)=\frac{h_{j,m,j(e)}(t)}{p_{j,m,j(e)}}. \tag{15}$$

$$T_{j,m,j(e)}=\int\limits_{0}^{\infty} t\cdot f_{j,m,j(e)}(t)dt; \tag{16}$$

$$D_{j,m,j(e)}=\int\limits_{0}^{\infty}\left(t-T_{m(\Sigma),n(\Sigma)}\right)^{2}\cdot f_{j,m,j(e)}(t)dt. \tag{17}$$

Due to $\mu_{j,m}$ gets subset E, veraciously,

$$\sum_{j(e)=1(e)}^{J(e)} p_{j,m,j(e)}=1. \tag{18}$$

3 Transformation PMN to Complex Semi-Markov Process

With use formulae (13), (14), (15) obtained, Petri-Markov sublet, circled on the Fig. 2 with dashed line, may be replaced with the single state, and so the PMN Π may be replaced with complex semi-Markov process, which describes behavior of the swarm as a whole. Circled with dashed line subset is as follows

$$\Pi_j=\left\{\mathrm{A}_j,\mathrm{Z}_j,\iota\left(\mathrm{Z}_j\right),\mathrm{o}\left(\mathrm{Z}_j\right)\right\}, \tag{19}$$

$$\mathrm{A}_j=\left\{\alpha_{j,0},\alpha_{j,1},\ldots,\alpha_{j,m},\ldots,\alpha_{j,M}\right\}, \tag{20}$$

$$\begin{cases}\mathrm{Z}_j=\left\{\zeta_{j,0},\zeta_{j,1}\right\};\\ \iota\left(\zeta_{j,0}\right)=\alpha_{j,0};\\ \iota\left(\zeta_{j,1}\right)=\left\{\alpha_{j,1},\ldots,\alpha_{j,m},\ldots,\alpha_{j,M}\right\};\end{cases}, \tag{21}$$

$$\begin{cases} \mathrm{o}\left(\mathrm{Z}_j\right) = \left\{\mathrm{o}\left(\zeta_{j,0}\right), \mathrm{o}\left(\zeta_{j,1}\right)\right\}; \\ \mathrm{o}\left(\zeta_{j,0}\right) = \left\{\alpha_{j,1}, \ldots, \alpha_{j,m}, \ldots, \alpha_{j,M}\right\}; \\ \mathrm{o}\left(\zeta_{j,1}\right) = \left\{\alpha_{1,0}, \ldots, \alpha_{j,0}, \ldots, \alpha_{J,0}\right\} \end{cases}, \tag{22}$$

Semi-Markov process is as follows

$$\mu = \{B, \boldsymbol{r}, \boldsymbol{h}(t)\}, \tag{23}$$

where $B = \left\{b_1, \ldots, b_j, \ldots, b_J\right\}$ is the set of states, which are abstract analogues of execution by swarm the complex operation due to algorithm of swarm behavior;

$\boldsymbol{r} = \left[r_{j,n}\right]$ is the $J \times J$ adjacency matrix; $\boldsymbol{h}_{j,m}(t) = \left[h_{j,n}(t)\right]$ is the $J \times J$ semi-Markov matrix.

Let us define probabilities and time densities of switch from the complex state $b_j \in B$ to the complex state $b_n \in B$. Semi-steps $\left(\alpha_{j,0}, \zeta_{j,0}\right)$ and $\left(\zeta_{j,0}, \alpha_{j,m}\right)$, $1 \le m \le M$ are executed during the time which is defined with Dirac δ-function. Semi-steps $\left(\zeta_{j,1}, \alpha_{0,n}\right)$, $1 \le n \le J$ are executed after logical conditions fulfillment also during defied with Dirac δ-function time. So time of residence the complex semi-Markov process in the state b_j till switch to the complex state b_n may be defined as result of competition between ordinary semi-Markov processes $\mu_{j,m}$, with taking into account outcomes of getting subsets $E_{j,m}$ states.

For definition of all possible outcomes, from indexes trios $\left[j, m, j(e)\right]$, $1 \le m \le M$, following set may be constructed

$$\tilde{J}_{j,m} = \left\{\left[j, m, 1(e)\right], \ldots, \left[j, m, j(e)\right], \ldots, \left[j, m, J(e)\right]\right\}, \quad 1 \le m \le M. \tag{24}$$

Cartesian product of sets $\tilde{J}_{j,m}$ gives all possible combinations of outcomes of swarm units operation:

$$\begin{gathered} \tilde{J}_j = \coprod_{m=1}^{M} \tilde{J}_{j,m} = \{([j, 1, 1(e)], \ldots, [j, m, 1(e)], \ldots, [j, M, 1(e)]), \ldots, \\ ([j, 1, j(e)], \ldots, [j, m, j(e)], \ldots, [j, M, j(e)]), \ldots, ([j, 1, J(e)], \ldots, [j, m, J(e)], \ldots, [j, M, J(e)])\}; \end{gathered} \tag{25}$$

From (25) may be selected those vectors, combination of trios of which permit to do emi-step $\left(\zeta_{j,1}, \alpha_{n,0}\right)$:

$$\begin{aligned} \tilde{J}_j \supseteq \tilde{J}_{j(n)} = \{&\left(\left[j, 1, i(e, n)\right], \ldots, \left[j, m, j(e, n)\right], \ldots, \left[j, M, k(e, n)\right]\right), \\ &\ldots, \left(\left[j, 1, l(e, n)\right], \ldots, \left[j, m, q(e, n)\right], \ldots, \left[j, M, s(e, n)\right]\right)\}; \end{aligned} \tag{26}$$

$$\left|\tilde{J}_{j(n)}\right| = \mathrm{K}(n) \tag{27}$$

Probability of $\kappa(n)$-th combination emergence is as follows:

$$p_{j,\kappa(n)} = \prod_{m=1}^{M} p_{j,m,j[e,\kappa(n)]}, \quad 1 \le \kappa(n) \le \mathrm{K}(n). \tag{28}$$

So, probability $p_{j,n}$ of switch the complex semi-Markov process μ from b_j to b_n is equal to the sum

$$p_{j,n} = \sum_{\kappa(n)=1}^{\mathrm{K}(n)} p_{j(n),\kappa(n)}. \tag{29}$$

For calculation of time density $f_{j,n}(t)$ one should consider the competition [9, 14] on the transition $\zeta_{j,1}$ between ordinary semi-Markov processes $\mu_{j,m}$. Residence at the state b_j is considered as completed, when last ordinary semi-Markov process reaches its $E_{j,m}$ state according the combination (25). This is why semi-Markov processes compete for not being the last in the competition. Time of reaching subset $E_{j,m}$ by all M may be described with the following formula:

$$\begin{aligned} f_{j,j[e,\kappa(n)]}(t) &= \frac{d \prod_{m=1}^{M} F_{j,m,j[e,\kappa(n)]}(t)}{dt} \\ &= \sum_{m=1}^{M} f_{j,m,j[e,\kappa(n)]}(t) \prod_{\substack{i=1, \\ i \neq m}}^{M} F_{j,i,j[e,\kappa(n)]}(t). \end{aligned} \tag{30}$$

where $f_{j,m,j[e,\kappa(n)]}(t)$ is the time density of reaching the transition $\zeta_{j,1}$ by m-th semi-Markov process due to $\kappa(n)$-th combination; $F_{\ldots}(t) = \int\limits_0^t f_{\ldots}(\tau)d\tau$ if the function of distribution of probabilities.

With taking into account combination of outcomes, weighted and pure time densities of switch from the state b_j to the state b_n are as follows:

$$h_{j,n}(t) = \sum_{\kappa(n)=1}^{\mathrm{K}(n)} p_{j,\kappa(n)} f_{j,j[e,\kappa(n)]}(t). \tag{31}$$

$$f_{j,n}(t) = \frac{h_{j,n}(t)}{p_{j,n}}. \tag{32}$$

After transformation for investigation of swarm behavior investigation and calculation of wandering time intervals all possible methods of semi-Markov process analysis may be used [9–12].

4 Effectiveness of Synchronization

One of the important aspects of swarm operation organization is elimination of unproductive units downtime when corporative task is solved. Parameter, which defines effectiveness, may be any, but when investigation of relay-races, distributed forfeit is of widely used. Let us considered competition of m-th и l-th swarm units, first of which gets the transition $\zeta_{j,1}$ during time $f_{j,m,j[e,\kappa(n)]}(t)$, but second - during the time $f_{j,l,j[e,\kappa(n)]}(t)$. In the case of winning in competition the first swarm unit he waits until l-th swarm unit gets $\zeta_{j,1}$. Waiting time is calculated as follows [15]:

$$f_{j,m\to l,j[e,\kappa(n)]}(t)=\frac{\eta(t)\int\limits_0^\infty f_{j,m,j[e,\kappa(n)]}(\tau)f_{j,l,j[e,\kappa(n)]}(t+\tau)d\tau}{\int\limits_0^\infty F_{j,m,j[e,\kappa(n)]}(t)dF_{j,l,j[e,\kappa(n)]}(t)},\tag{33}$$

where τ is an auxiliary argument; $\eta(t)$ is the Heaviside function.

Probability of such event, expectation and dispersion of waiting time are as follows

$$p_{j,m\to l,j[e,\kappa(n)]}=\int\limits_0^\infty F_{j,m,j[e,\kappa(n)]}(t)dF_{j,l,j[e,\kappa(n)]}(t);\tag{34}$$

$$T_{j,m\to l,j[e,\kappa(n)]}=\int\limits_0^\infty t\cdot f_{j,m\to l,j[e,\kappa(n)]}(t)dt;\tag{35}$$

$$D_{j,m\to l,j[e,\kappa(n)]}=\int\limits_0^\infty\left[t-T_{j,m\to l,j[e,\kappa(n)]}\right]^2 f_{j,m\to l,j[e,\kappa(n)]}(t)dt.\tag{36}$$

Every of value (34), (35), (36) may characterized those or that effectiveness aspect, but more universal is the criterion, which is defined as distributed forfeit [15] $c_{j,m\to l,j[e,\kappa(n)]}(t)$. Forfeit, receives m-th swarm unit from the l-th swarm unit if it gets the transition $\zeta_{j,1}$ earlier. Latecomer pays forfeit during whole the time until he m-th swarm unit waits him. In this case weighted forfeit sum is equal to

$$C_{j,m\to l,j[e,\kappa(n)]}=p_{j,m\to l,j[e,\kappa(n)]}\int\limits_o^\infty c_{j,m\to l,j[e,\kappa(n)]}(t)f_{j,m\to l,j[e,\kappa(n)]}(t)dt;\tag{37}$$

Common forfeit sum, which m-th swarm unit receives from all other units by $\kappa(n)$-th combination variant is as follows:

$$C_{j,m,j[e,\kappa(n)]}=\sum_{\substack{l=1,\\ l\neq m}}^{M}C_{j,m\to l,j[e,\kappa(n)]}.\tag{38}$$

Common forfeit sum, which m-th swarm unit receives from all other units in the case of further switch into state b_n is equal to

$$C_{j,m,n}=\sum_{\kappa(n)=1}^{\mathrm{K}(n)}p_{j,m,j[e,\kappa(n)]}\cdot C_{j,m,j[e,\kappa(n)]}.\tag{39}$$

Sum $C_{j(b),m}$ depends on parameters of ordinary semi-Markov processes (7), and forfeit discipline. Such sum may be used as optimization criterion in the task of producing optimal swarm behavior.

5 Conclusion

Working out the model of swarm synchronized operation opens new page in parallel systems theory because it permits to link real physical parameters of hardware with structure and logics of operation oh corporative algorithm, distributed among swarm units. Algorithm splitting may be done with those or that way, but with use approach proposed, swarm program designer for every mode of splitting may evaluate main characteristics both corporative algorithm as a whole, and parts of it, realized on swarm units.

Further investigation in this area should be directed to working out an algorithm splitting optimization method, based on proposed approach to parallelization modeling and evaluation of effectiveness.

The research was supported by the Foundation for Basic Research under the project 19-47-710004 r_a.

References

1. Tzafestas, S.G.: Introduction to Mobile Robot Control, p. 750. Elsevier (2014)
2. Siciliano, B.: Springer Handbook of Robotics, p. 1611. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-30301-5
3. Squillante, M.S.: Stochastic analysis and optimization of multiserver systems. In: Ardagna, D., Zhang, L. (eds.) Run-time Models for Self-managing Systems and Applications, pp. 1–24. Springer, Basel (2010). https://doi.org/10.1007/978-3-0346-0433-8_1
4. Landau, I.D., Zito, G.: Digital Control Systems, Design, Identification and Implementation, p. 484. Springer, Heidelberg (2006). https://doi.org/10.1007/978-1-84628-056-6
5. Aström, J., Wittenmark, B.: Computer Controlled Systems: Theory and Design, p. 557. Tsinghua University Press. Prentice Hall (2002)
6. Larkin, E.V., Ivutin, A.N.: Estimation of latency in embedded real-time systems. In: 3rd Meditteranean Conference on Embedded Computing (MECO 2014), Budva, Montenegro, 15–19 June 2014, pp. 236–239 (2014). https://doi.org/10.1109/MECO.2014.6862704
7. Pacheco, P.S.: An Introduction to Parallel Programming, p. 361. Elsevier (2011)
8. Larkin, E.V., Malikov, A.A., Ivutin, A.N.: Petri-Markov model of fault-tolerant computer systems. In: 4th International Conference on Control, Decision and Information Technologies (CoDIT), Barcelona, Spain, pp. 416–420. IEEE (2017)
9. Du, R., Ai, S., Hu, O.: Competition and cooperation between brands in a segment: an analysis based on a semi-Markov model. Int. J. Serv. Sci. **2**(1), 70–82 (2009)
10. Janssen, J., Manca, R.: Applied Semi-Markov Processes, p. 310. Springer, Boston (2006). https://doi.org/10.1007/0-387-29548-8
11. Jiang, Q., Xi, H.-S., Yin, B.-Q.: Event-driven semi-Markov switching state-space control processes. IET Control Theory Appl. **6**(12), 1861–1869 (2012)
12. Larkin, E., Ivutin, A., Kotov, V., Privalov, A.: Semi-Markov modelling of commands execution by mobile robot. In: Ronzhin, A., Rigoll, G., Meshcheryakov, R. (eds.) ICR 2016. LNCS (LNAI), vol. 9812, pp. 189–198. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-43955-6_23
13. Kobayashi, H., Marl, B.L., Turin, W.: Probability, Random Processes and Statistical Analysis, p. 812. Cambridge University Press, Cambridge (2012)

14. Eisentraut, C., Hermanns, H., Zhang, L.: Concurrency and composition in a stochastic world. In: Gastin, P., Laroussinie, F. (eds.) CONCUR 2010. LNCS, vol. 6269, pp. 21–39. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15375-4_3
15. Larkin, E.V., Ivutin, A.N., Kotov, V.V., Privalov, A.N.: Simulation of Relay-races. Bull. South Ural State Univ. Math. Model. Program. Comput. Softw. **9**(4), 117–128 (2016)