Preprints, 1st IFAC Conference on Modelling, Identification and
Control of Nonlinear Systems
June 24-26, 2015. Saint Petersburg, Russia

Simultaneous Perturbation Stochastic Approximation in Decentralized Load Balancing Problem ⋆

Natalia Amelina, [*] **Victoria Erofeeva,** [*] **Oleg Granichin,** [**] **Nikolai Malkovskii** [*]

[*] *Saint Petersburg State University (Faculty of Mathematics and Mechanics), St. Petersburg, Russia*
e-mail: ngranichina@gmail.com, vicki.ultramarine@gmail.com, malkovskynv@gmail.com
[**] *Saint Petersburg State University (Faculty of Mathematics and Mechanics, and Research Laboratory for Analysis and Modeling of Social Processes), Institute of Problems in Mechanical Engineering, Russian Academy of Sciences, and ITMO University, St. Petersburg, Russia*
e-mail: oleg_granichin@mail.ru

Abstract:
In this work the load balancing problem is studied for decentralized stochastic network with *unknown but bounded* noise in measurements and varying productivities of agents. The load balancing problem is formulated as a consensus problem in a stochastic network. Consideration of Laplasian potential function corresponded to the network graph allows to introduce a new randomized local voting protocol with constant step-size which is based on simultaneous perturbation stochastic approximation algorithm. The conditions are formulated for the approximate consensus achievement which corresponds to achieving of a suboptimal level of agents' load. The new algorithm is illustrated by simulations.

Keywords: Simultaneous perturbation stochastic approximation, randomized algorithms, multiagent systems, consensus problem.

1. INTRODUCTION

In recent years the consensus approach has been widely used for solving different practical problems Olfati-Saber and Murray (2004); Olfati-Saber et al. (2007); Ren et al. (2007); Ren and Beard (2008); Chebotarev and Agaev (2009); Kar and Moura (2009); Granichin et al. (2012); Amelin et al. (2013); Lewis et al. (2014), including the load balancing problem Amelina et al. (2015). For the problem of achieving consensus a lot of theoretical results were obtained. In Tsitsiklis et al. (1986); Huang and Manton (2009); Li and Zhang (2009) the stochastic approximation type algorithms were used for achieving the consensus, and their applicability under some statistical uncertainties was analyzed in Amelina and Fradkov (2012); Amelina et al. (2015), where it was assumed that measurement noise and delays have a statistical nature with standard properties of zero-mean and bounded covariance.

Emphasize, when the undirected topology graph has a spanning tree, the load balancing problem can be reformulated as a minimization problem of a Laplacian potential associated with a graph (see Olfati-Saber and Murray (2004)). In this paper we suggest to use *a simultaneous perturbation stochastic approximation (SPSA)* for solving this problem. SPSA algorithm recursively generates estimates along a random directions and uses only two observations of minimized function at each iteration. SPSA and similar procedures with one (or two) measurements per iteration were introduced in Granichin (1989, 1992) Polyak and Tsybakov (1990). and Spall (1992). They are similar to random search methods Rastrigin (1963). The general overview of SPSA type algorithms and their applications in different fields are done in Granichin et al. (2015). Generally, a centralized algorithm for load balancing which is based on SPSA was considered in Granichin and Amelina (2015); Granichin (2015).

The paper is organized as follows. In Section II, the problem statement is described, and basic concepts of a graph theory that are used hereinafter are introduced. In Section III, the load balancing control strategy is considered. Section IV presents a new result about a mean-risk optimization problem under linear constrains. In Section V we introduce the new randomized local voting protocol and Section VI gives conditions of an asymptotic mean square ε-consensus. Simulation results are given in Section VII. Section VIII contains conclusions.

2. PROBLEM FORMULATION

Let the network system be composed by m agents (processors, machines, *etc.*) which are numbered by naturals $i,\ i = 1, \dots, m$, and $N = \{1, \dots, m\}$ be a set of agents in the system. This system executes a set of tasks of the same type. Tasks feed to the system in different discrete time instants $t = 0, 1, \dots$ through different agents. Agents perform incoming tasks in

⋆ The authors acknowledge the Russian Ministry of Education and Science (agreement 14.604.21.0035, unique no. RFMEFI60414X0035), RFBR (projects 13-07-00250, 14-08-01015, and 15-08-02640), and SPbSU (project 6.37.181.2014).

parallel. Tasks can be redistributed among agents based on feedbacks. We assume, that the task can not be interrupted after it has been assigned to the agent.

In this paper we use the following notation and terms from the matrix and graph theories. A communication graph (N, E) is defined by a set of nodes N and a set of edges E. A dynamic network of d agents is determined by a set of dynamic systems (agents) that interact according to the communication graph. We associate a weight $a^{i,j} > 0$ with each edge $(j, i) \in E$. A graph can be represented by *an adjacency matrix* $A = [a^{i,j}]$ with weights $a^{i,j} > 0$ if $(j, i) \in E$, and $a^{i,j} = 0$ otherwise. Assume, that $a^{i,i} = 0$. We use the notation $\mathcal{G}_A$ for a graph which is represented by an adjacency matrix A. Define *a weighted in-degree* of node i as a sum of i-th row of matrix A: $d^i(A) = \sum_{j=1}^{n} a^{i,j}$, and $D(A) = \mathrm{diag}\{d^i(A)\}$ as a corresponding diagonal matrix. Let $\mathcal{L}(A) = D(A) - A$ denotes the *Laplacian* of the graph $\mathcal{G}_A$. Note, that the sum of rows of the Laplacian equals to zero. The symbol $d_{\max}(A)$ stands for a maximum in-degree of the graph $\mathcal{G}_A$, $Re(\lambda_2(A))$ is the real part of the second eigenvalue of matrix A ordered by absolute magnitude, A^{T} is the transpose matrix. Let $N^i = \{j : a^{i,j} > 0\}$ be a "neighbors" set of agent $i \in N$, $|N^i|$ is a corresponding number of "neighbors". The graph $\mathcal{G}_A$ is called undirected if $a^{i,j} = a^{j,i}$ for all $i, j \in N$.

At each time instant t the behavior of each agent $i \in N$ is described by two characteristics:

- q_t^i is the queue length of atomic elementary tasks of agent i at time instant t;
- θ_t^i is the productivity of agent i at time instant t.

Here and below, an upper index of agent i is used as a corresponding number of an agent (not as an exponent). The execution time of a task varies from one agent to another and depends on a productivity of an agent.

Consider the case when the dynamic model of the system is described by the following equations

$$q_{t+1}^i = q_t^i - \theta_t^i + z_t^i + u_t^i, \quad i \in N, \ t = 0, 1, \ldots, \tag{1}$$

where z_t^i are amounts of new system tasks received through agent i at time instant t; $u_t^i \in \mathbb{R}$ are control actions (redistributed tasks to agent i at time instant t — parts of system tasks previously received through other agents), which could (and should) be chosen.

We assume, that to form the control strategy u_t^i each agent $i \in N$ has knowledge about its own productivity, productivities of its neighbors and noisy data about its own queue length:

$$y_t^{i,i} = q_t^i + \xi_t^{i,i}, \tag{2}$$

and, if the neighbors set N^i is not empty, the knowledge about productivities of its neighbors and noisy observations about its neighbors' queue lengths:

$$y_t^{i,j} = q_t^j + \xi_t^{i,j}, \ j \in N_t^i, \tag{3}$$

where $\{w_t^{i,j}\}$ is an observation noise.

Denote T_t^i as a time moment when agent i completes currently assigned tasks (at time moment t). T_t^i can be formally described as:

$$T_t^i = \min_{\tau} \sum_{k=t}^{\tau} \theta_k^i \geq q_t^i.$$

Consider the problem of minimization of implementation time of all tasks:

$$\max_{i \in \{1,\ldots,m\}} T_t^i(q_0^i, u_1^i, z_1^i, u_2^i, z_2^i, \ldots) \to \min_{u_1^1,\ldots,u_1^m,u_2^1,\ldots}. \tag{4}$$

For the stationary case when $z_t^i = 0$ (i.e. there are no new receiving tasks for $t > 0$), such value does not vary over time and so the problem becomes a worst-case optimization problem (moreover, it is easy to show that the problem can be further reduced to minimization of some "good" convex functional). For the nonstationary case the problem is more difficult as we should trace "drifting" minimum point.

3. LOAD BALANCING

An ideal scheduling algorithm is the one which keeps all the nodes busy executing essential tasks, and minimizes the internode communication required to determine the schedule and pass data between tasks. The scheduling problem is particularly challenging when the tasks are generated dynamically and unpredictably in the course of executing the algorithm. This is the case when many recursive divide-and-conquer algorithms have to be used, including backtrack search, game tree search and branch-and-bound computation.

When all queue lengths and productivities (performance) of nodes are known, then the best control strategy is a proportional distribution of tasks such that

$$q^1/\theta^1 = q^2/\theta^2 = \cdots = q^m/\theta^m.$$

The proof of this result is not difficult and could be found, for example, in Amelina et al. (2015). This control strategy is called *load balancing*.

The reasons mentioned above allow us to reformulate the considering problem: *the goal is to maintain the balanced (equal) load across the network.*

Assume, that the following conditions are satisfied

A1: Graph $\mathcal{G}_A$ is undirected, and it has a spanning tree.
A2: $\theta_t^i \geq \theta_{\min} > 0, \ \forall i \in N, t = 0, 1, \ldots.$

(Note, if Assumption **A1** is satisfied then $0 < Re(\lambda_2(A))$ (see Lewis et al. (2014))).

If we take $x_t^i = q_t^i/\theta_t^i$ as a state of agent i of considered dynamic network at time instants $t = 0, 1 \ldots$, then the control goal of achieving consensus in network will correspond to the optimal redistribution of tasks among agents (see Amelina et al. (2015)). Under this notation, the dynamics of each agent can be rewritten as

$$x_{t+1}^i = x_t^i + \tilde{f}_t^i + \tilde{u}_t^i, \tag{5}$$

where $\tilde{f}_t^i = z_t^i/\theta_t^i - 1$, and $\tilde{u}_t^i = \bar{u}_t^i/\theta_t^i, \ i \in N$ are "normalized" control actions.

We can rewrite Equation (5) in the vector form

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{f}_t + \mathbf{u}_t, \tag{6}$$

where m-vectors $\mathbf{x}_t$, $\mathbf{f}_t$, and $\mathbf{u}_t$ consist of corresponding elements $x_t^1, \ldots, x_t^m$, $\tilde{f}_t^1, \ldots, \tilde{f}_t^m$, and $\tilde{u}_t^1, \ldots, \tilde{u}_t^m$.

If undirected graph $\mathcal{G}_A$ has a spanning tree, the load balancing problem can be reformulated as a minimization problem of a Laplacian potential associated with graph $\mathcal{G}_A$ (see Olfati-Saber and Murray (2004))

$$\Phi_t(\mathbf{x}_t) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{m} a^{i,j}(x_t^j - x_t^i)^2 \rightarrow \min_{\mathbf{x}_t}, \qquad (7)$$

$$subject\ to \sum_{i=1}^{m} x_t^i \theta_t^i = \sum_{i=1}^{m} q_{t-1}^i, \qquad (8)$$

since $\Phi_t(\mathbf{x}_t) = 0$ for the case $x_t^1 = x_t^2 = \ldots = x_t^m$ and $\Phi_t(\mathbf{x}_t) > 0$ for all other cases. Is is also mentioned in Olfati-Saber and Murray (2004) that local voting protocol (see, e.g., Amelina et al. (2015)) is equivalent to gradient descent for Laplacian potential. Linear constrain (8) is natural for problems of tasks redistribution because we cannot loss the tasks during a redistribution process.

To solve the problem (7),(8) we could use the algorithm and result from Granichin (2015).

4. MEAN-RISK OPTIMIZATION PROBLEM UNDER LINEAR CONSTRAINS

Consider a set of differentiable functions $\{f_w(\theta)\}_{w\in\mathbb{W}}$, $f_w(\theta) : \mathbb{R}^m \to \mathbb{R}$, let $\mathbf{x}_1, \mathbf{x}_2, \ldots$ be the set of observation points chosen by experimenter. For each $t = 1, 2, \ldots$ we get measurements $y_1, y_2, \ldots$ of $f_w(\cdot)$ with additive external noise v_t

$$y_t = f_{w_t}(\mathbf{x}_t) + v_t, \qquad (9)$$

where $\{w_t\}$ is an uncontrollable sequence, $w_t \in \mathbb{W}$.

Let $(\Omega, \mathcal{F}, P)$ be the underlying probability space, and let $\mathcal{F}_{t-1}$ be the σ-algebra of all probabilistic events occurred before $t = 1, 2, \ldots$.

The problem is to find optimal $\theta_t^\star$ that minimizes mean-risk functional

$$F_t(\theta) = E_{\mathcal{F}_{t-1}} f_{w_t}(\theta) \to \min_\theta \qquad (10)$$

subject to linear constrains

$$H_t\theta = \mathbf{q}_{t-1} \qquad (11)$$

with matrices H_t of dimension $k \times m$ and vectors $\mathbf{q}_{t-1} \in \mathbb{R}^k$, $0 \le k < m$ (with $k = 0$ it is assumed that there is no constrains).

Hereandafter E is a symbol for mean value and $E_{\mathcal{F}_{t-1}}$ is a symbol for conditional mathematical expectation with respect to $\mathcal{F}_{t-1}$, $\langle\cdot,\cdot\rangle$ is a scalar product of two vectors, $\|\cdot\|$ is an Euclidean norm of a vector.

If $rankH_t = k$ then there exists linear function $h_t : \mathbb{R}^m \to \mathbb{R}^{m-k}$ and its reverse function $g_t : \mathbb{R}^{m-k} \to \mathbb{R}^m$ such as

$$\mathbf{x} = g_t(h_t(\mathbf{x})),\ \forall \mathbf{x} \in \mathbb{M}_t = \{H_t\mathbf{x} = \mathbf{q}_{t-1}\}.$$

We assume that $h_t(\cdot)$ could always be chosen.

Let $\Delta_n,\ n = 1, 2, \ldots$ be an observed sequence of independent random variables in $\mathbb{R}^{m-k}$, called the simultaneous test perturbation, with Bernoulli distribution which elements equal ±1 with probabilities $\frac{1}{2}$.

Let us take a fixed initial vector $\widehat{\theta}_0 \in \mathbb{R}^m$ and choose positive numbers α and β. Consider the algorithm

$$\begin{cases} \mathbf{x}_n^\pm = g_{2n-1\pm\frac{1}{2}}(h_{2n-1\pm\frac{1}{2}}(\widehat{\theta}_{2n-2}) \pm \beta\mathbf{\Delta}_n), \\ \widehat{\theta}_{2n-1} = g_{2n-1}(h_{2n-1}(\widehat{\theta}_{2n-2})), \\ \widehat{\theta}_{2n} = g_{2n}(h_{2n}(\widehat{\theta}_{2n-1}) - \alpha\mathbf{\Delta}_n \dfrac{y_n^+ - y_n^-}{2\beta}), \end{cases} \qquad (12)$$

which is similar to one proposed in Granichin (2015) when $H_t(\cdot)$ does not depend on t.

Next, we assume the following about $f_{w_t}(\mathbf{x})$, $F_t(\mathbf{x})$ and uncertainties in the model:

A3: Function $F_t(\cdot)$ has unique minimum point $\theta_t^\star$ and $\forall \mathbf{z} \in \mathbb{R}^{m-k}$

$$\langle \mathbf{z} - h_t(\theta_t^\star), E_{\mathcal{F}_{t-1}} \nabla_\mathbf{z} f_{w_t}(g_t(\mathbf{z}))\rangle \ge \mu\|\mathbf{z} - h_t(\theta_t^\star)\|^2$$

with a constant $\mu > 0$.

A4: $\forall w_t \in \mathbb{W}$ gradient $\nabla_\mathbf{z} f_{w_t}(g_t(\mathbf{z}))$ satisfies the Lipschitz condition: $\forall \mathbf{z}', \mathbf{z}'' \in \mathbb{R}^{d-k}$

$$\|\nabla_\mathbf{z} f_{w_t}(\mathbf{z}') - \nabla_\mathbf{z} f_{w_t}(\mathbf{z}'')\| \le M\|\mathbf{z}' - \mathbf{z}''\|$$

with a constant $M \ge \mu$.

A5: Vector-gradient $\nabla\tilde{f}_t(\cdot)$ is uniformly bounded in point $h_t(\theta_t^\star)$: $\|E\nabla\tilde{f}_t(h(\theta_t^\star))\| \le c_1$, $E\|\nabla\tilde{f}_t(h(\theta_t^\star))\|^2 \le c_2$, $E\langle\nabla\tilde{f}_t(h(\theta_t^\star)), \nabla\tilde{f}_{t-1}(h(\theta_{t-1}^\star))\rangle \le c_2$ ($c_1 = c_2 = 0$ if w_t is nonrandom, i.e. $f_{w_t}(\mathbf{x}) = F_t(\mathbf{x})$).

A6: For $n = 1, 2, \ldots,$

a) $\mathbf{\Delta}_n$ and w_{2n-1}, w_{2n} (if they are random) do not depend on σ-algebra $\mathcal{F}_{2n-2}$.

b) If w_{2n-1}, w_{2n} are random then random vectors $\mathbf{\Delta}_n$ and elements w_{2n-1}, w_{2n} are independent.

c) the successive differences $\bar{v}_n = v_{2n} - v_{2n-1}$ of observation noises are bounded:

$|\bar{v}_n| \le c_v < \infty$, or $E\bar{v}_n^2 \le c_v^2$, if a sequence $\{v_t\}$ is random.

d) If $\bar{v}_n$ is random then $\bar{v}_n$ and vector $\mathbf{\Delta}_n$ are independent.

A7: Matrices H_{2n-1} and H_{2n} (if they are random) do not depend on σ-algebra $\mathcal{F}_{2n-2}$.

A8: The drift is bounded: $\|h_t(\theta_t^\star - \theta_{t-1}^\star))\| \le \delta_\theta < \infty$, or $E\|h_t(\theta_t^\star - \theta_{t-1}^\star)\|^2 \le \delta_\theta^2$ and $E\|h_t(\theta_t^\star - \theta_{t-1}^\star)\|\|h(\theta_{t-1}^\star - \theta_{t-2}^\star)\| \le \delta_\theta^2$, if a sequence $\{w_t\}$ is random.

The rate of drift is bounded in a such way that $\forall \mathbf{z} \in \mathbb{R}^{d-k}$: $E_{\mathcal{F}_{2n-2}}\varphi_n(\mathbf{z})^2 \le c_3\|\mathbf{z} - h_t(\theta_{2n-2}^\star)\|^2 + c_4$, where $\varphi_n(\mathbf{x}) = f_{w_{2n}}(\mathbf{x}) - f_{w_{2n-1}}(\mathbf{x})$.

Denote $\kappa = 2(\mu - \alpha\gamma)$, $b = 2\beta M c_\Delta^3(1 + 6\alpha M c_\Delta^2) + \delta_\theta(M + 2\mu + 6\alpha M^2 c_\Delta^4)$, $\bar{l} = 2\alpha c_\Delta{}^2(c_v^2 + 3(\max_n \frac{c_4}{2\beta} + c_\Delta^2(c_2 + M^2(\delta_\theta + 2\beta c_\Delta)^2))) + 2\delta_\theta(4\beta M c_\Delta^3 + M\delta_\theta + c_1 + 3\mu\delta_\theta^2)$, where $\gamma = 3c_\Delta{}^2(M^2 c_\Delta^2 + \frac{c_3}{2\beta})$.

The following Theorem shows the asymptotically efficient mean-squared weak upper bound of estimation residuals by algorithm (12).

Theorem 1. If $rankH_t = k$, assumptions **A3-A8** hold, and α is sufficiently small: $\alpha \in (0; \mu/\gamma)$ if $\mu^2 > 2\gamma$, or $\alpha \in (0; \frac{\mu - \sqrt{\mu^2 - 2\gamma}}{2\gamma}) \cup (\frac{\mu + \sqrt{\mu^2 - 2\gamma}}{2\gamma}; \mu/\gamma)$ otherwise,

then the sequence of estimates provided by the algorithm (12) has asymptotically efficient mean-squared weak upper bound of estimation residuals

$$\bar{L} = (b + \sqrt{b^2 + \kappa\bar{l}})/\kappa, \qquad (13)$$

i.e. $\forall\varepsilon > 0\ \exists N$ such that $\forall n > N$

$$\sqrt{E\|\widehat{\theta}_{2n} - \theta_{2n}^\star\|^2} \le \bar{L} + \varepsilon.$$

Proof of Theorem 1 is slightly different from the correspondence proof in Granichin (2015) since we consider more complicated problem setting and additional Assumption **A5**.

5. TASK REDISTRIBUTION PROTOCOL

Generally, to ensure load balancing across a network (in order to increase the overall throughput of a system and to reduce execution time) it is naturally to use the redistribution protocol over time.

Minimum point $\mathbf{x}_t^\star$ of (7) vary over time due to the system dynamics (6). Consider SPSA algorithm (12) with nonvanishing step-sizes for tracking the changes $\mathbf{x}_t^\star$ using $h_t(\mathbf{x}) = h(\mathbf{x}) = col(x^1, \dots, x^{m-1})$

$$g_t(\mathbf{z}) = col(z^1, \dots, z^{m-1}, \frac{\sum_{i=1}^m q_{t-1}^i - \sum_{j=1}^{d-1} z^j \theta_t^i}{\theta_t^m}).$$

We have initial guess $\widehat{\mathbf{x}}_0$ which is formed by q_0^i/θ_0^i, $i = 1, \dots, m$. Let $\alpha > 0$ and $\beta > 0$ be fairly small step-sizes.

The iteration step consists of

- Compute two values

$$y_n^\pm = \Phi_{2n-1\pm\frac{1}{2}}(g_{2n-1\pm\frac{1}{2}}(h(\widehat{\mathbf{x}}_{t-1}) \pm \beta\mathbf{\Delta}_t)); \tag{14}$$

- Compute quasigradient vector

$$\widehat{\nabla}_n = \mathbf{\Delta}_n \frac{y_n^+ - y_n^-}{2\beta}; \tag{15}$$

- Get new estimate

$$\widehat{\mathbf{x}}_{2n-1} = g_{2n-1}(h(\widehat{\mathbf{x}}_{2n-2});$$
$$\widehat{\mathbf{x}}_{2n} = g_{2n}(h(\widehat{\mathbf{x}}_{2n-2}) - \alpha\widehat{\nabla}_n). \tag{16}$$

We cannot use (16) in decentralized load balancing problem since each agent is able to use information about its neighbors only.

Consider the ith component of the quasigradient vector from (15). By virtue (7) and (14) we have

$$\widehat{\nabla}_t^i = \Delta_t^i \frac{\Phi_t(\widehat{\mathbf{x}}_{t-1} + \beta\mathbf{\Delta}_t) - \Phi_t(\widehat{\mathbf{x}}_{t-1} - \beta\mathbf{\Delta}_t)}{2\beta} =$$
$$\Delta_t^i \frac{1}{4\beta} \sum_{k=1}^n \sum_{j=1}^n a^{k,j} \times$$
$$\times \left((x_t^j + \beta\Delta_t^j - x_t^k - \beta\Delta_t^k)^2 - (x_t^j - \beta\Delta_t^j - x_t^k + \beta\Delta_t^k)^2 \right).$$

By using the difference of squares (formula: $a^2 - b^2 = (a-b)(a+b)$), we derive

$$\widehat{\nabla}_t^i = \Delta_t^i \sum_{k=1}^n \sum_{j=1}^n a^{k,j} (\Delta_t^j - \Delta_t^k)(x_t^j - x_t^k) =$$
$$\sum_{j \in N^i} (a^{i,j} + a^{j,i})(1 - \Delta_t^i\Delta_t^j)(x_t^j - x_t^i) +$$
$$\Delta_t^i \sum_{k \neq i}^n \sum_{j \neq i}^n a^{k,j} (\Delta_t^j - \Delta_t^k)(x_t^j - x_t^k),$$

since $(\Delta_t^i)^2 = 1$. Denoting $\eta_t^i = \sum_{k\neq i}^n \sum_{j\neq i}^n a^{k,j}(\Delta_t^j - \Delta_t^k)(x_t^j - x_t^k)$ we get

$$\widehat{\nabla}_t^i = 2 \sum_{j \in N^i} a^{i,j}(1 - \Delta_t^i\Delta_t^j)(x_t^j - x_t^i) + \Delta_t^i \eta_t^i.$$

Following by the SPSA iteration step (16) we could consider decentralized control protocol

$$u_t^i = \alpha \sum_{j \in N^i} a^{i,j}(1 - \Delta_t^i\Delta_t^j) \left(\frac{\theta_t^i}{\theta_t^j} y_t^{i,j} - y_t^{i,i} \right),\ i \in N, \tag{17}$$

where $\alpha > 0$ is a step-size of control protocol (17).

For each $i \in N$ the dynamics of the closed loop system with protocol (17) is as follows

$$x_{t+1}^i = x_t^i + \tilde{f}_t^i + \alpha \sum_{j \in N^i} a^{i,j}(1 - \Delta_i\Delta_j) \left(\frac{y_t^{i,j}}{\theta_t^j} - \frac{y_t^{i,i}}{\theta_t^i} \right). \tag{18}$$

If we denote matrix $B_t = [b_t^{i,j}]$, where $b_t^{i,j} = a^{i,j}(1 - \Delta_t^i\Delta_t^j)$, then properties of a similar control algorithm, called a local voting protocol, for a load balancing problem were studied in Amelina et al. (2015). The common feature is that the control value of the local voting protocol for each agent was determined by the weighted sum of differences between the information about the state of the agent and the information about its neighbors' states. However, the analysis in Amelina et al. (2015) was done only for the case of statistical noise (noise with Gaussian distribution) with standard zero-mean and bounded covariance properties. Here we consider the randomized modification (the special case) of the local voting protocol, which was inspired by SPSA methods. Probably we could use weaker conditions about observation noise $\{v_t^{i,j}\}$ and disturbances $\tilde{f}_t^i$ if we assume the independence of simultaneous test perturbation $\mathbf{\Delta}_t$ on noise and disturbances (see Granichin et al. (2015)).

5.1 Connection to gossip algorithms

For considered SPSA algorithm we used Bernoulli distributed simultaneous perturbation $\mathbf{\Delta}$ but in fact we can use different $\mathbf{\Delta}$ without losing core properties of the SPSA. In Granichin et al. (2015) required conditions are presented (chapter 3, conditions (3.8)). One possible distribution is as follows:

$$\Delta_n^k = \begin{cases} 0, & k \neq i, j \\ \pm 1, \text{ w.p. } \frac{1}{2}, & k = i \text{ or } k = j \\ \mp 1, \text{ Assuring } \Delta_n^i = -\Delta_n^j & k = i \text{ or } k = j \end{cases}$$

If we apply such Δ to (18) there will be only two nonzero coordinates and whole sum contains single nonzero term. Multiplier $(\Delta_n^j - \Delta_n^k)$ is ± 2 for nonzero terms and the sign doesn't affect summary value. With that, considered stochastic approximation procedure can be described as "iteratively pick random pair of indices and average values of corresponding coordinates" (value of β affects how much the values are drawn to their average). Such algorithms was previously studied in Boyd et al. (2006) and are known as "gossip" algorithms. As we mentioned in previous sections, SPSA algorithm works with arbitrary bounded noises. With that, gossip algorithms should work with arbitrary noises as well *if choice of a pair of indices does not correlate with external noise*.

6. ASYMPTOTIC MEAN SQUARE ε-CONSENSUS

Definition 1. n agents are said to achieve *the asymptotic mean square ε-consensus*, if $\mathrm{E}\|x_0^i\|^2 < \infty,\ i \in N$, and there exists a sequence $\{x_t^\star\}$ such that

$$\overline{\lim}_{t\to\infty} \mathrm{E}\|x_t^i - x_t^\star\|^2 \leq \varepsilon$$

for all $i \in N$.

Assume that the following assumptions are satisfied:

A9: a) For each $i, j = 1, \dots, m$ vector $\mathbf{\Delta}_t$ and $\xi_t^{i,j}$ (if it is random) are independent.
b) For each $i = 1, \dots, m$ vector $\mathbf{\Delta}_t$ and z_t^i (if it is random) are independent.

c) For each $i = 1, \dots, m$ vector $\boldsymbol{\Delta}_t$ and θ_t^i (if it is random) are independent.

d) For all $i, j \in N$, $t = 1, 2, \dots$ observation noise $\xi_t^{i,j}$ is bounded: $|\xi_t^{i,j}| \leq c_\xi < \infty$, or $E(\xi_t^{i,j})^2 \leq c_\xi^2$ if $\xi_t^{i,j}$ is random.

e) For each $i = 1, \dots, m$ z_t^i is bounded: $|z_t^i| \leq c_z < \infty$, or $E(z_t^i)^2 \leq c_z^2$ if z_t^i is random.

Let $x_0^\star$ be the average of the initial data

$$x_0^\star = \frac{1}{n}\sum_{i=1}^{n} x_0^i$$

and $\{x_t^\star\}$ is the trajectory of the averaged system

$$x_{t+1}^\star = x_t^\star + \frac{1}{n}\sum_{i=1}^{n} \tilde{f}_t^i. \qquad (19)$$

Theorem 1 allows to derive the level of upper bound of the asymptotic mean square ε-consensus:

$$\overline{\lim}_{t\to\infty} \mathrm{E}\|\mathbf{x}_t - x_t^\star \mathbf{1}_m\|^2 \leq \varepsilon$$

with some ε which can be calculated using Theorem 1result. Here $\mathbf{1}_m$ is m-vector of ones,

For considered case

$$\mu = Re(\lambda_2(A)),\ M = 4d_{\max}(A).$$

To verify the applicability of Theorem 1 we need to check that:

1: The function $\Phi_t(\cdot)$ is strongly convex on subspace $\mathbb{X}_t = \{\mathbf{x} \in \mathbb{R}^m : \mathbf{x}^{\mathrm{T}}\mathbf{1}_m = \mathbf{x}_t^{\mathrm{T}}\mathbf{1}_m\}$, i.e. it has a unique minimum point $x_t^\star \mathbf{1}_m$ and

$$(\mathbf{x} - x_t^\star \mathbf{1}_m)^{\mathrm{T}} \nabla\Phi_t(\mathbf{x}) \geq \mu\|\mathbf{x} - x_t^\star \mathbf{1}_m\|^2,\ \ \forall \mathbf{x} \in \mathbb{X}_t$$

with a constant $\mu = Re(\lambda_2(A)) > 0$.

Calculating the derivatives, we get

$$\frac{\partial\Phi_t(\mathbf{x})}{\partial i} = -\sum_{j=1}^{n} a^{i,j}(x^j - x^i) + -\sum_{j=1}^{n} a^{j,i}(x^i - x^j).$$

Hence, gradient-vector $\nabla\Phi_t(\mathbf{x})$ equals to $2\mathcal{L}(A)\mathbf{x}$.

The vector $\mathbf{1}_m$ is the right eigenvector of Laplacian matrix $\mathcal{L}(A)$ and corresponding to the zero eigenvalue:

$$\mathcal{L}(A)\mathbf{1}_m = 0.$$

Sums of all elements in rows of matrix $\mathcal{L}(A)$ is equal to zero and, moreover, all the diagonal elements are positive and equal to the absolute value of the sum of all other elements in the row.

By virtue Assumption **A1** matrix A has a spanning tree. By Lemma 2.10 from Ren and Beard (2008) the rank of matrix $\mathcal{L}(A)$ equals $m-1$. Hence we can derive

$$(\mathbf{x} - x_t^\star \mathbf{1}_m)^{\mathrm{T}} \nabla\Phi_t(\mathbf{x}) = 2(\mathbf{x} - x_t^\star \mathbf{1}_m)^{\mathrm{T}} \mathcal{L}(A)\mathbf{x} =$$
$$2(\mathbf{x} - x_t^\star \mathbf{1}_m)^{\mathrm{T}} \mathcal{L}(A)(\mathbf{x} - x_t^\star \mathbf{1}_m) \geq Re(\lambda_2(A))\|\mathbf{x} - x_t^\star \mathbf{1}_m\|^2,$$

2: By using Gershgorin criteria (see Lewis et al. (2014)), we get that the gradient $\nabla\Phi_t(\mathbf{x})$ satisfies the Lipschitz condition: $\forall \mathbf{x}', \mathbf{x}'' \in \mathbb{R}^m$

$$\|\nabla\Phi_t(\mathbf{x}') - \nabla\Phi_t(\mathbf{x}'')\| = 2\|\mathcal{L}(A)(\mathbf{x}' - \mathbf{x}'')\| \leq$$
$$4d_{\max}(A)\|\mathbf{x}' - \mathbf{x}''\|$$

with a constant $M = 4d_{\max}(A) \geq \mu = Re(\lambda_2(A))$.

3: By virtue Assumption **A9a** the vector $\boldsymbol{\Delta}_t$ does not depend on observation noise and drift.

4: By virtue Assumption **A9d** the observation noise satisfies:

$$\left|\sum_{j\in N^i} a^{i,j}(1 - \Delta_i\Delta_j)\left(\frac{\xi_t^{i,j}}{\theta_t^j} - \frac{\xi_t^{i,i}}{\theta_t^i}\right)\right| \leq$$
$$4d_{\max}(A)c_v/\theta_{\min} < \infty.$$

5: By virtue Assumption **A9e** the drift is bounded:

$$\|x_t^\star \mathbf{1}_m - x_{t-1}^\star \mathbf{1}_m\| = |\frac{1}{m}\sum_{i=1}^{m} \tilde{f}_t^i|\sqrt{m} \leq$$
$$\max\{1, c_z/\theta_{\min} - 1\}\sqrt{n} = \delta_{\mathbf{x}} < \infty.$$

7. SIMULATION RESULTS

To illustrate the theoretical results we consider the decentralized computer network of $m = 50$ computing nodes. We will show that the proposed randomized control algorithm (17) provides load balancing of the network similar to the one presented in Fig. 1.

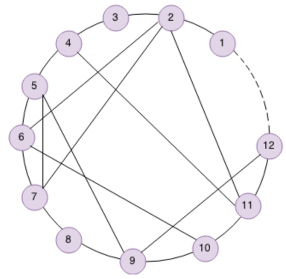

Fig. 1. The network topology.

The network topology is a ring with chords which are randomly chosen by the following rule for every node:

(1) simulate a number of added chords by a Poisson distribution with mean value $m/2$
(2) randomly select nodes that attach to the current (the number of such units is equal to the value obtained in step 1)

We generate the initial productivities $\theta_t^1, \theta_t^2, \dots, \theta_t^m$ randomly by the uniform distribution over the interval $(10; 50)$. We assume that productivity measure in our case is the number of available jobs in time instant $t = 0, 1, \dots$, the productivities do not change over time and $\theta_t^i \neq 0\ \forall\ i$.

The tasks are divided into two sets: regular and burst. The first one is served on each tact to a randomly chosen node and the second one at any given time. During system operation we will be adding regular tasks from the interval $(12; 100)$ and burst tasks from $(10000; 25000)$.

Fig. 2 shows the dependence of algorithm convergence rate on choosing of coefficient α.

In Fig. 3, we can see the system of $m = 50$ nodes operating in nonstationary case with the control protocol (17). Each line indicates how the load x_t^i evolves over time. For clarity, the chart displays 3 maximum and 3 minimum values. These lines also show how the system evolves to reach load-balancing or consensus. We can see that even when the new burst task set is received during the system work, it does not affect the quality of load balancing. During the simulation we have set the coefficient $\alpha = 0.007$, which is the most suitable value for the current topology and chosen parameters (see Fig. 2). In addition

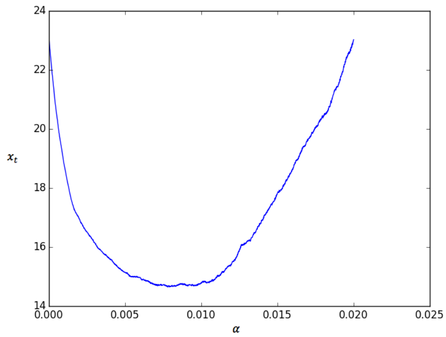

Fig. 2. Rate of convergence based on α.

to the obtained results, it is planned to study the possibility of SPSA application for tracking the optimal value of α.

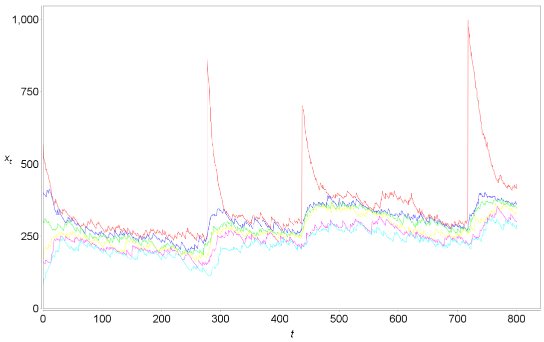

Fig. 3. Perfomance of the system with $m = 50$ nodes x_t^i for the nonstationary case.

8. CONCLUSION

In this paper the problem of load balancing in a multi-agent system under *unknown but bounded* disturbances was examined. To solve the load balancing problem the new randomized local voting protocol with nonvanishing step-size was proposed. Conditions for achieving an approximate consensus (balance of the network load) were obtained. To illustrate the theoretical results we presented the simulations for the computing network.

REFERENCES

Amelin, K., Amelina, N., Granichin, O., Granichina, O., and Andrievsky, B. (2013). Randomized algorithm for uavs group flight optimization. *11th IFAC International Workshop on Adaptation and Learning in Control and Signal Processing*, 205–208.

Amelina, N. and Fradkov, A. (2012). Approximate consensus in the dynamic stochastic network with incomplete information and measurement delays. *Automation and Remote Control*, 73(11), 1765–1783.

Amelina, N., Fradkov, A., Jiang, Y., and Vergados, D. (2015). Approximate consensus in stochastic networks with application to load balancing. *IEEE Transactions on Information Theory*, 61(4), 1739–1752.

Boyd, S., Ghosh, A., Prabhakar, B., and Shah, D. (2006). Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6), 2508–2530.

Chebotarev, P.Y. and Agaev, R.P. (2009). Coordination in multiagent systems and laplacian spectra of digraphs. *Automation and Remote Control*, 70(3), 469–483.

Granichin, O. and Amelina, N. (2015). Simultaneous perturbation stochastic approximation for tracking under unknown but bounded disturbances. *IEEE Transactions on Automatic Control*, 60(5).

Granichin, O., Skobelev, P., Lada, A., Mayorov, I., and Tsarev, A. (2012). Comparing adaptive and non-adaptive models of cargo transportation in multi-agent system for real time truck scheduling. *Proceedings of the 4th International Joint Conference on Computational Intelligence*, 282–285.

Granichin, O., Volkovich, Z.V., and Toledano-Kitai, D. (2015). *Randomized Algorithms in Automatic Control and Data Mining*. Springer.

Granichin, O. (1989). A stochastic recursive procedure with correlated noise in the observation, that employs trial perturbations at the input. *Vestnik Leningrad University: Math*, 22(1), 27–31.

Granichin, O. (1992). Unknown function minimum point estimation under dependent noise. *Problems of Information Transmission*, 28(2), 16–20.

Granichin, O. (2015). Stochastic approximation search algorithms with randomization at the input. *Automation and Remote Control*, 76(5), 761–774.

Huang, M. and Manton, J. (2009). Coordination and consensus of networked agents with noisy measurements: stochastic algorithms and asymptotic behavior. *SIAM Journal on Control and Optimization*, 48(1), 134–161.

Kar, S. and Moura, J.M. (2009). Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise. *IEEE Transactions on Signal Processing*, 57(1), 355–369.

Lewis, F.L., Zhang, H., Hengster-Movric, K., and Das, A. (2014). *Cooperative control of multi-agent systems: optimal and adaptive design approaches*. Springer Publishing Company, Incorporated.

Li, T. and Zhang, J. (2009). Mean square average-consensus under measurement noises and fixed topologies: Necessary and sufficient conditions. *Automatica*, 45(8), 1929–1936.

Olfati-Saber, R., Fax, J., and Murray, R. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1), 215–233.

Olfati-Saber, R. and Murray, R. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9), 1520–1533.

Polyak, B.T. and Tsybakov, A.B. (1990). Optimal order of accuracy of search algorithms in stochastic optimization. *Problemy Peredachi Informatsii*, 26(2), 45–53.

Rastrigin, L. (1963). The convergence of the random search method in the extremal control of a many-parameter system. *Automation and Remote Control*, 24(10), 1337–1342.

Ren, W. and Beard, R. (2008). *Distributed Consensus in Multi-vehicle Cooperative Control. Communications and Control Engineering*. Springer.

Ren, W., Beard, R., and Atkins, E. (2007). Information consensus in multivehicle cooperative control. *Control Systems, IEEE*, 27(2), 71–82.

Spall, J.C. (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3), 332–341.

Tsitsiklis, J., Bertsekas, D., and Athans, M. (1986). Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9), 803–812.