TELKOMNIKA, Vol.14, No.4, December 2016, pp. 1408~1416
ISSN: 1693-6930, accredited A by DIKTI, Decree No: 58/DIKTI/Kep/2013
DOI: 10.12928/TELKOMNIKA.v14i4.4997

Planning and Coordination in Hierarchie of Intelligent Dynamic Systems

Alexander Ya. Fridman
Institute for Informatics and Mathematical Modelling,
Kola Science Centre of the Russian Academy of Sciences,
24A Fersman str., 184209 Apatity Murmansk reg., Russia
Tel./fax +7 81555 74050, e-mail: fridman@iimm.ru

Abstract

On the basis of the known principle of interactions prediction (Mesarovic), our earlier proposed incremental coordination principle is extended over hierarchical collectives of intelligent dynamic systems (IDSs) after Gennady Osipov. Such systems admit arbitrary types of variables in their state vector and thereby allow investigating more general dynamic systems than "classical" ones defined in numerical state spaces. Using the concept of effective N-attainability (Osipov), a straightforward procedure of planning for hierarchical collectives of IDS is developed. As soon as a plan for reaching a goal state from the current one is found, effective implementation of this plan requires for coordination of IDSs taking their parts in the collective. We consider both aspects of coordination (coordinability with respect to the coordinator's task and coordinability in relation to the global task) and infer necessary conditions of the coordinability for a locally organized hierarchy of IDSs.

Keywords: *intelligent dynamic system, incremental coordination, direct planning, locally organized hierarchy*

1. Introduction

Hierarchical control became standard for complex systems due to increasing difficulty of centralized management for such systems. It was necessary to divide the decision-making process into several levels to get reasonable complexity of optimization tasks on each of them. However, advent of multilevel hierarchical systems raised a new problem of matching and coordinating the decisions made on different control levels (see, for instance, [1-3]). The key problems in the development of such systems are: specialization of subsystems within their inherent problems and coordination of control impacts at different levels of the hierarchy. In other words, the tasks for sub-systems and their quality criteria are necessary to form such a way that the shared performance of their tasks allows subsystems to perform a global task for entire hierarchy (compatibility postulate [4]).

In general, the construction of the coordination principles (in particular, the interactions prediction principle [4]) requires for seeking satisfactory solutions at the level of the lower decisive elements, which is consistent with the modern methods of decentralized control. Before this, it is necessary to solve three problems: first, to build a metric in the state space of the system; second, to specify a coordination principle, and third, to plan combine activities of the lower-level elements for reaching the general goal of the whole system. Let us consider the existing ways of solving these problems.

To solve the first of the mentioned problems, there are known methods to metrize spaces of data and knowledge in order to build their hierarchical taxonomies (e.g., [5, 6]). Such metrics are based on plant characteristic tables of the taxonomy elements that are not suitable for dynamic systems. In economic applications of hierarchical systems (for example, [7]), they only use financial indicators. This limits the generality of estimation and control for the state elements characterized by different units of measure. Methods of expert judgment processing like [8] are neither focused on the dynamic control problems.

As for the second problem, coordination is mostly investigated with regard to information coordination [9, 10], behaviors coordination [11] or coordination of peer-rank objects [12, 13], while most complex systems are hierarchical and need dynamic coordination of

Received September 21, 2016; Revised November 8, 2016; Accepted November 23, 2016

interactions among subsystems of different levels in decision making.

In the literature, the third problem (planning) concerns mechanical aspects of robots functioning [14-16] and does not consider informational interlinks among subsystems, which include their own decision makers.

In connection with the above-described, a change-based (gradient for continuous states and incremental – for discrete states) generalized criterion was proposed in [17] for the state estimation in hierarchical dynamic systems that allows to analyze numerical state elements. In terms of multi-objective optimization (e.g., [18]), this criterion belongs to the global criteria with weight coefficients inversely proportional to the tolerances of scalar criteria. This idea looks reasonable since the more important is a criterion for the whole system, the less its devations are admissible from the Coordinator's point of view.

On this basis, we have developed a coordination principle for hierarchical systems [17] that implements the interactions prediction principle [4] for elements of a hierarchical system, taking into account the method of ensuring stability of local control signals in the collectives of automata [19]. Our coordination technique uses the necessary and sufficient conditions of coordinability for a locally organized hierarchy of dynamic systems.

In terms of system analysis, the proposed principle to coordinate hierarchical systems corresponds to the external (objective) approach to assessing the effectiveness of subsystems within a metasystem. This principle states as follows: sub-objects tasks will be coordinated with respect to the Coordinator's task, if the sign of the gradient of the generalized Coordinator's criterion for its current dominant scalar criterion will coincide with signs of gradients of this generalized criterion for all current values of scalar criteria for sub-objects. Efficiency of this technique was illustrated by its simulation for a network object [17].

Thus, the earlier developed coordination principle fits only to hierarchical systems with quantitative quality criteria and numerical state spaces. In this paper, we extend the idea of incremental coordination to the systems admitting other types of variables as well. The approach we propose is implemented below for intelligent dynamic systems (IDSs) [20] designed for modeling complex dynamic systems in the state spaces with arbitrary types of state elements.

To illustrate our idea of coordination, we introduce IDSs paradihm [21] first, and then describe means for planning and coordination in this formalism since existing a plan to reach a goal state is a necessary condition for coordinating.

2. Synopsis of IDS [13, 20, 21]

An IDS is described as a discrete dynamic system.

$$D = \langle X, N, \varphi, \psi \rangle, \tag{1}$$

where: X is a topological state space $x \in X$ with the proximity relation σ; N is the set of natural numbers, which mark discrete points of time; ϑ is the set of all subsets of X;

φ: $\vartheta \rightarrow \vartheta$ is a closure function with the following property:

$$\text{if } x \in \vartheta, \text{ then } x \subseteq \varphi(x); \tag{2}$$

ψ: $\vartheta \times N \rightarrow \vartheta$ is a transition function with the properties corresponding to requirements for transition functions in the "classical" control theory:

$$\psi(x, 0) = x \text{ for any } x \in \vartheta,$$
$$\psi(\psi(x, t_1), t_2) = \psi(x, t_1 + t_2). \tag{3}$$

If an IDS is based on rules that contain sets of formulas of a certain language L, dynamics of this IDS in markovian case is described by the following equation:

$$x(t + 1) = \varphi(\psi(\varphi(x(t) \cup u(t) \cup \delta(t))), \tag{4}$$

Where: $u(t) \subseteq U(t) \subseteq U \subseteq L$ is a set of facts that are added to the state $x(t)$ (control signals); $\delta(t) \subseteq \Delta(t) \subseteq \Delta \subseteq L$ is a set of facts that appear as a result of unpredictable changes in IDS's environment (disturbances).

The trajectory (4) is stable, if the function φ is monotonic and ψ is monotonic with respect to the state vector (4) [21]. In what follows we assume that any referenced IDS meets these conditions.

2.1. IDSs Architecture

IDSs allow for knowledge representation both by rules and semantic networks [13]. Further on, we will consider the rule-based IDSs. In such an IDS, its knowledge base (KB) comprises a set of rules of the following format:

$$\Pi = < C, A, D > , \quad (5)$$

Where: C is a precondition (condition) of a rule;

A is a set of facts added after application of the rule Π;

D is a set of facts deleted after application of the rule Π.

C, A и D are sets of formulas of the language L.

Any rule must meet the relation.

$$A \cap D = \varnothing. \quad (6)$$

Any rule belongs to only one of the two classes: RD or RS.

Every rule of the RD class contains an action applied to the external environment by an executive body or a procedure that computes and assigns a variable with certain values of some database attributes considering their values available in the current state. These actions result in changes of the IDS's database state. This group of rules describes changes of the system's state in time and is called "rules of (diachronic) transition".

Rules of the RS class are bind with no actions, they do not change the environment; rather, they change the knowledge of it. In other words, they represent the theory of the subject domain.

Then the IDS's knowledge base is:

$$R = <\text{RS}, \text{RD}>. \quad (7)$$

2.2. IDSs Goal-Seeking Behaviour

As noted above, an IDS can be described by the relations (1) – (4). To comply with the property (2), no RS-class rules are supposed to be capable of removing facts [13].

Then $\psi(\bullet, 1)$ is the transition function, and $\{\varphi(\psi(x, i)) \mid i \in N\}$ describes an orbit or a trajectory of the dynamic system.

Dynamics of a rule-based IDS (in the Markov case) is described by equation (4).

Let us consider the relationship between architecture of the knowledge base R (7) and properties of the model (1) – (4) [20].

Suppose, $L(R)$ is a set of formulas from the language L, which occur in the rules of R.

Definition 1. If X is a set of IDS's states, then the pair of points (x_0, x_1) in the space $X \times X$ is called N-attainable one, if there exist control signals $U(j)$ ($j = 0, 1,\ldots, N - 1$), for which $x_1 \subseteq x(N)$ with the initial conditions $x(0) \subseteq x_0 \cup U(0)$, where $x(t)$, $t \in N$ are solutions of the IDS's state equation (4).

Definition 2. If a pair of points (x_0, x_1) is N-attainable and every fact of $x(N)$ does not occur in more than one rule within the corresponding trajectory, then the pair of points (x_0, x_1) is called effectively N-attainable.

Let $\tau \subset L(R)$ be a set of facts. If a sequence of rules $\Pi_1, \Pi_2, \ldots, \Pi_k$ from R is given, the set of facts $S(\Pi_1, \Pi_2, \ldots, \Pi_k)$ derived after application of τ rules from this sequence is defined by induction:

$$S(\Pi_1) = \tau \setminus D_1 \cup A_1;$$

$$S(\Pi_1, \Pi_2,\ldots, \Pi_i) = S(\Pi_1, \Pi_2 ,\ldots, \Pi_{i-1}) \setminus D(\Pi_i) \cup A(\Pi_i). \quad (8)$$

Definition 3. A rule Π_i is called admissible one, if there is a control U_{i-1}, for which:

$$C_i \subseteq (S(\Pi_1, \Pi_2,\ldots, \Pi_{i-1}) \cup U_{i-1}.) \quad (9)$$

Definition 4. The sequence of rules and controls $\Pi = <(\Pi_1, U_1), (\Pi_2, U_2), \ldots, (\Pi_k, U_k)>$ is called a plan to achieve the state ω from the current state ι, if:

1) each rule from this sequence is admissible;

2) $\omega \subseteq S(\Pi_1, \Pi_2 , \ldots, \Pi_k)$.

Theorem [20]. For every pair of points $(x_0, x_1) \in X \times X$, the plan $\Pi = <(\Pi_1, U_1), (\Pi_2, U_2), \ldots, (\Pi_k, U_k)>$ exists, if and only if the pair (x_0, x_1) is N-attainable.

In [13], an algorithm is proposed to search for a sequence of admissible rules and their relevant controls that make up the plan to achieve the state ω from the current state ι. According to the principles of dynamic programming, this algorithm works "backward in time" (starting from the target state). However, this approach is difficult to consider the rules of the RD class since they not always have an inverse operator. In connection with the written above, we now propose a planning procedure that works in "live time" starting from the initial IDS's state.

2.3. Direct Planning for IDSs

The above-cited theorem determines the necessary and sufficient conditions for existence a plan to transfer a system from an initial state x_0 to the end state x_1. If we toughen conditions of this theorem and require for effective N-attainability of a pair of points (x_0, x_1) rather than their N-attainability (see Definition 2), we can obtain a direct planning algorithm similar to the ideas of derivative-based control implemented in the "classical" automatic control theory.

Let $x(t + 1)$ be an IDS's state vector obtained by solving the equation (2) at the step t + 1. Then $x(t + 1) \setminus x(t)$ are the new facts that have appeared on this stage of inference. If the IDS is on a trajectory, which has the property of the effective N-attainability, then the emerging facts should not repeat all along this trajectory. Consequently, the following relation is true:

$$\bigcap_{t=1}^{N-1} x(t+1) \setminus x(t) = \varnothing . \quad (10)$$

Then for synthesis a plan by direct inference, it is possible to calculate the intersection of already appeared new facts at each step k:

$$\Sigma k = \bigcup_{t=0}^{k-1} x(t+1) \setminus x(t) \quad (11)$$

And choose the current control so as:

$$\Sigma k \cap (x(k+1) \setminus x(k)) = \varnothing . \quad (12)$$

Hence, the procedure of the direct planning will look as follows:

1) Let x_1 is a target state and x_0 is the initial state of an IDS.

2) $k := 0$, $\Sigma k := \varnothing$.

3) Let $x(k)$ be the current state. If $x_1 \subseteq x(k)$, then stop. Also, let $\Pi^k := \{\Pi_1^k, \Pi_2^k, \ldots, \Pi_l^k\}$ is the set of admissible rules at the step k.

4) Apply a rule from Π^k and check the condition (12). If it holds, then $k := k + 1$ and go to Step 3, otherwise select another (not applied yet) rule from Π^k and return to the beginning of Step 4. If none of the admissible rules result in fulfilling the condition (12), then the step k is considered a failure; the rule led to it is marked as a dead-end; backtrack to the previous step of inference; $k := k - 1$ and go to Step 3.

It is possible to show that, if there exist an effective (in the sense of Definition 2) plan to achieve the state x_1 from the state x_0, then the described procedure will complete construction of this plan. The speed of implementing the plan will increase, if the algorithm will look through the entire set of admissible rules at Step 4 and, if there are several rules that satisfy the condition (12), choose one of them, Π^{*k}, for which the following relation is true:

$$x_1 \setminus x^*(k) \subseteq x_1 \setminus x_i(k), \tag{13}$$

Where $x^*(k)$ is the state after application the rule Π^{*k} and $x_i(k)$ are the states after application of any other rules meeting the condition (12).

After obtaining a plan, we have to proceed with coordination during fulfilling this plan. In [13], the possibility to control interactions within a team of "peer" IDSs (the ones with identical knowledge bases) is described. An example of such a group can serve vehicles involved in road traffic. We believe, it is of interest to investigate hierarchical systems, which include IDSs as elements of different levels. This will be done in the next section.

3. Coordination in a Collective of IDSs

As in [4] and without loss of generality, we consider a two-level IDSs system (Figure 1) where the top-level desision maker (Coordinator) DM_0 sends coordinating signals (adjusting parameters of the quality criteria of the lower-level IDSs) γ_i to the subordinated IDSs $DM_1 – DM_n$ and receives their feedback signals w_i. For simplicity, we assume that all IDSs in the lower level are of the same type, i.e. have the same KBs (7). Such a team may take part, for example, in the solution of any construction task by a collective of robots, one of which serves as a job coordinator. Subordinated IDSs interact only via the controlled process P and have no information concerning the states of other IDSs in the same level, that is, the entire system is locally organized.

According to the principles of decentralized control, the purpose of coordinating signals γ_i, coming from the Coordinator to systems of the lower level, is a specification of such conditions for the tasks they have to manage that they would issue proper control signals m_i, which ensure fulfillment of both their own goals and the task of the whole system. Correspondingly, there exist two concepts of coordinability for the lower-level systems: coordinability with respect to the task of the Coordinator and coordinability in relation to the global task. There are several modes of coordination [4], the method of interactions prediction looks the most suitable for a collective of IDSs. In implementing this method, the Coordinator informs the subordinate systems about its desired values of interactions among them and each of the lower-level systems is trying to reach the corresponding predetermined value, assuming that the other subsystems will operate properly as well.

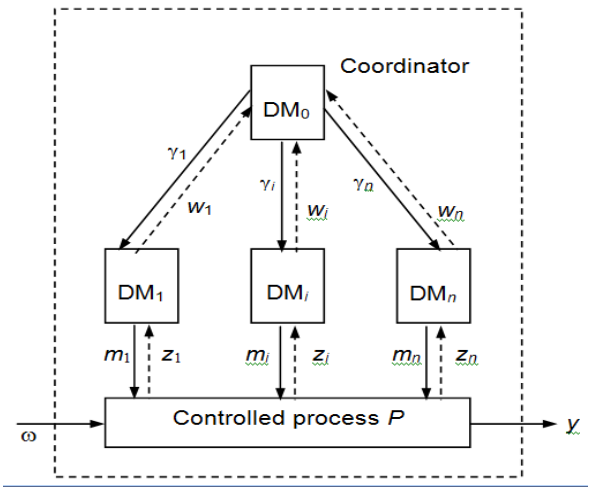

Figure 1. A two-level system of decision-making

3.1. Coordinability with Respect to the Coordinator's Task

When interpreting the method of interactions prediction for the two-level collective of IDSs, we assume that at the beginning of every stage of the operation t_j, the Coordinator informs DM_i about their desired states $\underline{x}_j$ to be achieved at this stage. Thus, we should take $\gamma_i = \underline{x}_j$ in Figure 1. As feedback signals from the lower-level IDSs, the Coordinator uses their states x_i as the most comprehensive sets of local information. Then natural formulation for coordinability conditions with respect to the Coordinator's task requires that each subordinated IDS can achieve the desired state from its current state x_i. Obviously, this requires for existence a plan to achieve $\underline{x}_i(t_j)$ from $x_i(t_j)$ [13], i.e. existence a sequence of control impacts on the environment, which would allow to come closer to the desired state (in general, in the presence of disturbances). It is also clear that, generally speaking, every DM_i may need different time to achieve a given state; therefore it is possible to propose two approaches to the organization of interactions among subsystems in time. Either the Coordinator shall generate the specified states, taking into account the potential of all lower-level IDSs to achieve them in a single cycle of the system, then you can synchronize the internal time of the IDSs, or the IDSs shall have the event-driven planning. Preference for one of the above approaches is determined by the specificity of the subject area. For simplicity, it is further assumed that the Coordinator's time and local times of all DM_i are synchronized, and its increment is accepted equal to 1. Thus, $t \in T = \{0, 1, 2,\ldots\}$.

Under coordination by the method of interactions prediction, disturbances will occur in a system DM_i, if the states of the lower-level systems are different from the ones set by the Coordinator. More detail description of the disturbances is only possible after a more specific description of the problem to be solved by a collective of IDSs and the environment; that is beyond the purpose of the given paper. However, in view of (4), we can interpret a necessary coordinability condition with respect to the Coordinator's task as a requirement to move "as close as possible" to end states for all dominated IDSs by the end of the current control step:

$$\forall \delta_i(t) \subseteq \Delta(t),\ \forall \underline{x}_i(t+1) \subseteq X,\ \forall i \in I;$$

$$\exists u_i^*(t) \subseteq U(t)\colon x_i^*(t+1) \setminus \underline{x}_i(t+1) \subseteq x_i(t+1) \setminus \underline{x}_i(t+1), \tag{14}$$

Where: $x_i^*(t+1) = \varphi(\psi(\varphi(x_i(t) \cup u_i^*(t) \cup \delta_i(t)))$ is the best possible state for the DM_i by the end of a control step;

$x_i(t+1) = \varphi(\psi(\varphi(x_i(t) \cup u_i(t) \cup \delta_i(t)))$ is any other attainable state for the DM_i by this instant.

In [13], it is shown that for stabilization of a trajectory of an IDS, i.e. for compensation of disturbances influence, it is enough to apply the following control:

$$u(x(t+1), t+1)) = \varphi(\psi(x(t+1)), t+1)) \setminus \varphi(\psi(\varphi(\psi(\varphi(x(t) \cup \delta(t)), t)), t+1)). \tag{15}$$

Considering (4), the relation (15) will look like:

$$u(x(t+1), t+1)) = \underline{\underline{x}}(t+2 / t+1)) \setminus \underline{x}(t+2 / t). \tag{16}$$

Where: $\underline{\underline{x}}(t+2 / t+1)$ is the prediction for the value of the state x at the instant t + 2 made at the time t + 1 in absence of disturbances; $\underline{x}(t+2 / t)$ is the prediction for the value of the state x at the instant t + 2 made at the time t considering disturbances, which existed at that point, and assuming absence of disturbances at the time t + 1.

Formulas (15), (16) allow to take into account unavoidable delay of control signals because of unpredictable disturbances from the environment [4].

Given (4) and (16), the necessary condition for coordinability with respect to the problem of the Coordinator (14) takes the form:

$$\forall \delta_i(t) \subseteq \Delta(t),\ \forall \underline{x}_i(t+1) \subseteq X,\ \forall i \subseteq I$$
$$\exists u_i^*(t) \subseteq U(t)\colon \underline{\underline{x}}_i^*(t+2 / t+1)) \setminus \underline{x}_i(t+2 / t) \subseteq \underline{\underline{x}}_i(t+2 / t+1)) \setminus \underline{x}_i(t+2 / t), \tag{17}$$

3.2. Coordinability in Relation to the Global Task

Let us assume first that the two-level system of IDSs is single-purpose, and this purpose is to achieve a given external state of the Coordinator $\underline{x}_0 \in X_0$. In the above-mentioned example with the construction, the purpose $\underline{x}_0$ can be formalized as a clear description of the expected results (e.g., a drawing of the building). Then the system will be coordinated in relation to the task of achieving $\underline{x}_0$, if the Coordinator will be able to find a set of predicted values $\underline{x}_i \subseteq X$, $\forall i \in I$ at the time t so that (after their issuance to the subordinated IDSs and subsequent impact of these systems on the environment) the Coordinator's state will move closer to $\underline{x}_0$. To present this statement more formally, let us detail the task of the Coordinator.

For the Coordinator, disturbances are deviations of the current states of its subordinated IDSs from their given values, and control signals are expected states of the lower-level IDSs. Accordingly, after receiving the feedback signals from the subordinated IDSs (a one-step delay is assumed to exist for each IDS and for reaction of the environment), the Coordinator's state equation can be represented similar to (4):

$$x_0(t + 4) = \varphi_0(\psi_0(\varphi_0(x_0(t) \cup u_0(t) \cup \delta_0(t))), \tag{18}$$

Where: $\delta_0(t)$: $X_1 \times X_2 \times \ldots \times X_n \rightarrow \Delta_0$ is the functional mapping of the impact of deviations of the current states of subordinates IDSs from their given values upon the general state of the job; $u_0(t)$ is a general description of the predicted job's state for the next step.

Relation (18) shows that the Coordinator DM_0 should solve two disparate tasks: first, to evaluate the current progress and to develop a strategy for further solving the problem on the basis of this assessment; second, to allocate tasks for the next step among the subordinated IDSs. Therefore, the structure of the Coordinator should be presented in the form shown in Figure 2.

Here: BOSS stands for the Block to Objectify the current State of solving the problem and to develop Strategies for further action; its state equation is described by the relation (18);

BCA_i denotes the Blocks to Correct Actions, which are responsible for mapping a generalized description of the job state predicted for the next step to the anticipated states of their subordinated DM_i.

Then, by analogy with (16), the formulated above coordinability statement for a collective of IDSs in relation to the global task can be written in the form:

$$\forall t \in T,\ \forall x_i(t) \subseteq X_i,\ \forall i \in I,\ \forall \underline{x}_0 \subseteq X_0$$

$$\exists x_i^*(t+4) \subseteq X_i\colon x_0^*(t+4) \setminus \underline{x}_0 \subseteq x_0^*(t) \setminus \underline{x}_0. \tag{19}$$

If you do not separate BCAs within the Coordinator, it should receive disturbances as a vector with the following components:

$$\delta_{i0} = x_i(t + 1) \setminus \underline{x}_i(t + 1),\ i \subseteq I, \tag{20}$$

And DM_0 will directly generate $\underline{x}_i$ as its output signals.

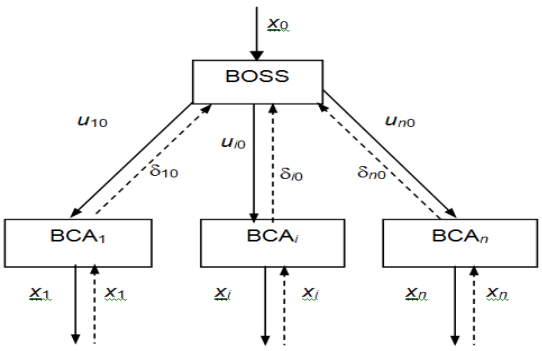

Figure 2. The Coordinator's structure

4. Results and Discussion

Change-based management procedures seem to be effective in different applications for control systems. In the prevoius and the given publications, we have proposed such methods to coordinate interactions among components of a hierarchical system and to plan its general behavior for both numeric and non-numeric metrics upon the state spaces of these components.

The possibility to construct hierarchies for solving multi-purpose tasks is evident as well. Then, the goal state of the Coordinator ($\underline{x}_0$ in Figure 2) will depend on time and should be selected within the system with the use of a preference relation on the set of goals [13]. Such problems can arise, for example, in problems of structure control for the virtual enterprises [22, 23]. However, they require for a separate consideration.

5. Conclusion

For locally organized hierarchical collectives of intelligent dynamic systems, we have found necessary conditions of coordinability both with respect to the Coordinator's task and in relation to the global task of the whole hierarchy. Besides, we have proposed a procedure for direct synthesis a plan to control such a hierarchy. Further research in this field ought to cover at least the following directions: search for necessary conditions of coordinability and planning; development of specific coordination algorithms for certain classes of jobs; looking for ways to prevent conflicts among the decision makers responsible for different subsystems within the hierarchy; extending ideas of incremental coordination to other problems like logistics, virtual enterprises, etc.

Acknowledgements

The author would like to thank the Russian Foundation for Basic Researches (grants 14-07-00257, 15-07-04760, 15-07-02757, 16-29-04424, and 16-29-12901) for partial funding of this research.

References

[1] Takahashi S, Kijima K, Sato R. Applied General Systems Research on Organizations. Springer. 2004.
[2] Delgado J, Pujol J, Sanguesa R. Emergence of Coordination in Scale-Free Networks. *Web Intelligence and Agent Systems*. 2003; 1: 131-138.
[3] Lee G, Chong NY. Self-organizing Mobile Sensor Network: Distributed Topology Control Framework. In: Handbook of Research on Ambient Intelligence and Smart Environments: Trends and Perspectives. IGI Global. 2011: 558-584.
[4] Mesarovic MD, Macko D, Takahara Y. Theory of Hierarchical Multilevel Systems. New-York and London: Acad. Press. 1970.
[5] Baroni M, Dinu G, Kruszewski G. *Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors*. 52nd Annual Meeting of the Association for Computational Linguistics. 2014: 1: 238-247.
[6] Beydoun G, Garcia-Sanchez F, Vincent-Torres CM, Lopez-Lorca AA, Martinez-Bejar R. Providing metrics and automatic enhancement for hierarchical taxonomies. *Information Processing & Management.* 2013; 49(1): 67-82.
[7] Mantegna RN. Hierarchical Structure in Financial Markets. *The European Physical Journal B.* 1999; 11: 193-197.
[8] Jonson RA, Wichern DW. Applied Multivariate Statistical Analysis. 6th edn. Pearson Prentice Hall. 2007.
[9] EB Foroutan, H Khotanlou. Improving semantic clustering using with Ontology and rules. *International Journal of Electrical and Computer Engineering (IJECE)*. 2014; 4(1): 7-15.
[10] Jiang W, Zhang J, Li J. A Multi-agent Supply Chain Information Coordination Mode Based on Cloud Computing. *TELKOMNIKA Indonesian Journal of Electrical Engineering.* 2013; 11(11): 6427-6433.
[11] Wicaksono H, Khoswanto H, Kuswadi S. Behaviors Coordination and Learning on Autonomous Navigation of Physical Robot. *TELKOMNIKA Telecommunication Computing Electronics and Control.* 2011; 9(3): 473-482.
[12] Qian Li, Limin Jia, Changxu Ji, Jianling H, Zhongyuan J. Simulating on Passengers Coordination with Distribution Service in Railway Station. *TELKOMNIKA Indonesian Journal of Electrical Engineering.* 2013; 11(1): 392-398.

[13] Osipov G. Intelligent dynamic systems. *Scientific and Technical Information Processing*. 2010; 37(5): 259–264.
[14] Iswanto, Wahyunggoro O, Cahyadi AI. Quadrotor Path Planning Based on Modified Fuzzy Cell Decomposition Algorithm. *TELKOMNIKA Telecommunication Computing Electronics and Control*. 2016; 14 (2): 655-664.
[15] Jingwei L, Yifei T, Shaofeng W, Qingmeng T, Dongbo L. Welding Robot Kinematics Analysis and Trajectory Planning. *TELKOMNIKA Telecommunication Computing Electronics and Control*. 2016; 14(2A): 92-100.
[16] Xuesong Y, Qinghua W, Hammin L. An Improved Robot Path Planning Algorithm. *TELKOMNIKA Telecommunication Computing Electronics and Control.* December 2012; 10(4): 629-636.
[17] Fridman OV, Fridman A Ya. Decreasing Dissemination of Disturbances within Network Systems by Neural Networks. *TELKOMNIKA Indonesian Journal of Electrical Engineering.* 2013; 11(9): 4942-4948.
[18] Marler RT, Arora JS. Survey of multi-objective optimization methods for engineering. *Struct Multidisc Optim*. 2004; 26: 369-395.
[19] Stefanuk VL. *Stability of Local Control in a System with Contralinear Interaction of Subsystems*. European Control Conference (ECC'93), Groningen. 1993; 1: 117-119.
[20] Osipov G. *Attainable Sets and Knowledge Base Architecture in Discrete Dynamic Knowledge-based Systems*. Workshop "Applied Semiotics: Control Problems (ASC 2000)". ECAI2000. 14th European Conference of Artificial Intelligence. Berlin. 2000: 39-43.
[21] Vinogradov AN, Zhilyakova L Yu, Osipov GS. Dynamic Intelligent Systems: I. Knowledge Representation and Basic Algorithms. *Computer and Systems Sciences International.* 2002; 41(6): 953-960.
[22] Sokolov B, Fridman A. *Integrated Situational Modelling of Industry-Business Processes for Every Stage of Their Life Cycle*. 4th International IEEE Conference “Intelligent Systems” (IS 2008). Varna, Bulgaria. 2008; 1: 8-40.
[23] Sokolov B, Ivanov D, Fridman A. Situational Modelling for Structural Dynamics Control of Industry-Business Processes and Supply Chains. In: Sgurev V, Hadjiski M, Kacprzyk Ja. *Editors*. Intelligent Systems: From Theory to Practice. Berlin Heidelberg: Springer-Verlag. 2010: 279-308.