

# **A System of Intelligent Algorithms for a Module of Onboard Equipment of Mobile Vehicles**

**Anna Gorbenko**

Department of Intelligent Systems and Robotics  
Ural Federal University  
620083 Ekaterinburg, Russia  
[gorbenko.ann@gmail.com](mailto:gorbenko.ann@gmail.com)

Copyright © 2013 Anna Gorbenko. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## **Abstract**

The area of intelligent robotics is moving from the single robot control problem to that of controlling multiple robots operating together and even collaborating in dynamic and unstructured intelligent environments. In such conditions, an intelligent robot control system is only part of general intelligent system. In this paper, we consider a model of such system.

**Mathematics Subject Classification:** 68T05, 68T40

**Keywords:** intelligent systems, mobile robots, Runge Kutta neural network, GSAT, robot control

## **1 Introduction**

Intelligent control systems has been extensively studied in robotics (see e.g. [1, 2, 3]). It is well-known that the area of intelligent robotics is moving from the single robot control problem to that of controlling multiple robots operating together and even collaborating in dynamic and unstructured intelligent

environments (see e.g. [4, 5]). Therefore, it is natural to consider an intelligent robot control system as a part of general intelligent system. In particular, for mobile robots, we need to consider a module of onboard equipment of mobile vehicles. Note that an usage of an unmanned vehicle is preferable during solving many practical tasks. In some cases, even for low quality of motion unmanned vehicles have an advantages, which is associated with impossibility of presence of a human or danger for human life. Perhaps, the most simple and inexpensive way for creation of unmanned vehicle is an usage of some onboard equipment for traditional vehicles that provides a remote operator control system for such vehicles. However, in many cases, full transference of control to the operator is impossible. For many applications, there is a large number of different problems with remote operator control systems. In particular, we can mention problems with connections: noise, delays, active interferences. It is clear that partial or full disconnection does not should lead to loss of controllability. We can not allow unpredictable behaviors of vehicles. For this purposes, a system of intelligent algorithms is needed. In particular, such system can allow us to obtain some local control system during communication failures. It is clear that we can use some similar ideas for a module of onboard equipment of traditional vehicles and onboard modules of robots in intelligent environments. We use this approach to consider a general model of a module of onboard equipment of mobile vehicles.

## 2 Measurement of Noises

It should be noted that a system of intelligent algorithms can be used in different modes according to traffic capacity of data channel and levels of noises in it. To determine correct values of these parameters and to measure degrees of reliance of incoming signals it is necessary a separate intelligent algorithm. Functioning of this algorithm is based on analysis of time series by combinatorial algorithms and intelligent system of prediction of anomalies in incoming data.

## 3 Information about Environment

Measurements of the traffic capacity of a data channel, levels of noises, and degrees of the reliance of control signals allow us to estimate a proper level of the vehicle control autonomy. However, these data are insufficient to choose a final model of an intelligent module. The local control system should have some information about environment. Therefore, the system of algorithms of onboard equipment have subsystems for gathering and analysis of sensor data.

It should be noted that sensor data transmitted to operator can become

unavailable. On the other hand, it is economically feasible to store all sensor data on vehicle board. Therefore, a specialized intelligent system in real time analyzes sensor data and performs preparation of onboard information during the functioning of vehicle. This information can be used in the case of disconnection.

## 4 Selection of Simple Intelligent Algorithms

According to selected autonomy mode an onboard navigation system uses selection of simple intelligent algorithms. In particular the system of such algorithms includes

- installation of the route from point to point;
- pursuit of object;
- avoidance of objects;
- object surveillance;
- maintenance of distance;
- maintenance of formation with other vehicles;
- rounding of obstacles;
- collision avoidance;
- installation of route for independent return to base.

## 5 Module of Self-Awareness

In parallel with the system of simple navigation algorithms operates a module of self-awareness of vehicle control system (see e.g. [6] – [10]) and anticipation of simple facts (see e.g. [11] – [14]). This system should perform following tasks:

- anticipation of different dangers;
- anticipation of changes in the environment;
- malfunction detection of vehicle.

Moreover, a module of self-awareness with module of simple algorithms implements task of construction of more complex navigation algorithms. In particular, we can mention such algorithms as

- search of safe shelter;
- finding of advantageous point for surveillance;
- movement through rugged terrain and environment that is dangerous for navigation.

Another important function is estimating of efficiency of current algorithms of navigation. This function is performed by module of self-awareness. In this case, if some algorithm of navigation will be recognized as ineffective, then a module of self-awareness generates a request for later regeneration of this algorithm.

It is should to be noted that regeneration of navigation algorithms is an expensive task from the computational point of view. Therefore, it is preferable to transmit a request for regeneration to the operator on the base. If a connection with the operator is impossible, then a module of self-awareness puts the control system to the mode of reduced operability for releasing of computational resources.

## 6 Generating System

There are three problems that are associated with the development of navigation systems,

- automatization of creation of simple navigation systems;
- automatization of method of selection of appropriate navigation system;
- accumulation and generalization of knowledge for facilitation of transition from construction of simplest navigation systems to more complex.

A natural way to solve mentioned above problems is a creation of some generating system (see e.g. [15]) which is able to generate modules of navigation by some data stream automatically, construct from these modules new navigation systems and generate control modules, providing a correct selection of navigation system for the specific task. Herewith such system can have knowledge base for storing statistical information. Generation of new modules and navigation systems on the basis of existing ones is a classical genetic programming task. In the simplest case, for the same type of navigation modules it is possible to construct a generalizing module for account of simultaneously connected initial modules.

Direct usage of the simultaneous connection of initial modules leads to a successive increase of sizes of generated modules, and in the end to the decrease

of its computational efficiency. Therefore, for efficient functioning of generating system a library of specialized algorithms is needed. These algorithms should provide a maintenance of computational efficiency of constructed modules. In particular, this library should contain

- minimization algorithms (e.g. simulated annealing, construct minimal automata, algorithm of minimization of neural network through learning a network of specified architecture, different methods of extraction of common part);
- algorithms of intelligent parallelization of neural networks and genetic algorithms;
- algorithms of transformation and embedding methods (e.g. methods of construction a neural networks by genetic algorithms, algorithms of embedding of finite state automaton in neural networks, methods of construction a genetic algorithms over some specified grammar etc.).

In the process, navigation system of vehicle keeps statistics of incoming information. Herewith functioning is considered within the framework of model “onboard computer – control computer – supercomputer”. Thus, generation of new navigation modules occurs on supercomputer and not in the module of onboard equipment. In this case, transmission of tasks to a supercomputer, keeping statistics, performing of auxiliary algorithms redistribution and processing of statistical data is implemented on control computer. This makes it possible to significantly reduce cost of module of onboard equipment, herewith all computations are performed on supercomputer. Thus, if it is necessary to change navigation algorithm or operating conditions of a system are changed, the system sends a signal for regeneration of algorithm. A supercomputer implements analysis of current conditions and generates an efficient algorithm. Then it sends the new algorithm to a vehicle.

## 7 Autonomy of a Vehicle

One of the important elements in the remote control is an autonomy of a vehicle. Naturally, requirements to a vehicle control system are determined by its architecture, environment, the range of solved task, and an autonomy mode, in which operates the vehicle. An autonomy mode is a some state of robot control system, which shows how the robot depends from human or other robot during performing the task.

When robot performs under a human command or a human control, there are several kinds of control that depend from degree of human participation in control. Among them, three main types can be allocated. First is described full

autonomy of system work, when robot performs a some specified task without any guidance. The next type involves semiautonomous robots, that is a robot partially depends from a human control. In this case, a human can have only limited set of commands. Last type is a mode in which robot performs actions under a full human control.

It is important to note that different algorithms can be used depending from the level of autonomy. This algorithms provide to a vehicle more or less freedom in choice of action. For example, decision making during navigation in typical simple conditions can be fully transferred on the module of onboard equipment. From selected autonomy mode can also depend a mode of computational resources distribution and different kinds of treatment of sensor information. For example, in the case of disconnection with a control computer a vehicle can independently generate some simple navigation modules on board. Otherwise in complex undefined situations the vehicle can wait instructions from control center or independently return to base.

## **8 Correspondence Between the System of Self-Awareness and the Navigation System**

It is clear that we can consider a correspondence between the navigation system and the self-awareness system. For example, from the one hand, a vehicle should follow the instructions of navigation algorithms, and not take into account own knowledge about what the vehicle should do. From the other hand, it is possible that following to the instructions of navigation algorithms is disadvantageously for the vehicle, and it should intensify a self-awareness system. For example, it is possible that a some heavy object fell into the vehicle. When this happens, the vehicle can conclude about slipping of wheels and tracks herewith onboard camera captures the same visual picture, which is almost unchanged, and process of motion continues. In this case, the vehicle can for some time not accept instructions from navigation algorithm and can try independently get out from under the object, analyzing dependence of own motions from sensor data.

## **9 The General System of Motion Planning**

The general system of motion planning can be constructed from external and internal modules. To create the external module of motion planning we can use system of multi-agent path planning [16] and system of reconfiguration planning for modular robots [1]. These algorithms can be used as a base part of module. To improve the module we can use self-learning system that utilizes

the module of safe planning [2] and the model of Hamiltonian path with fixed number of color repetitions [3].

Internal module needed for single robot motion planning. It is natural to consider motion planning for humanoid robots separately (see e.g. [17, 18]). For mobile robots, the problem of motion planning can be considered as the problem of generation of sequences of motor primitives. To solve this problem we can use different string problems and algorithms for these problems (see [19] – [32]). To improve the internal module we can use the model of one shot learning by imitation.

## 10 One Shot Learning by Imitation and Feature Distortion Warping

An algorithm for one shot learning by imitation was proposed in [33] for humanoid robots. It is clear that the same idea can be applied to any mobile robot. This algorithm creates the task path from the cloud of possible waypoints using minimum-energy strategy. In particular, the idea of feature distortion warping is used.

Following [33], we can define the  $k$  Cartesian coordinates  $a_i$  from the template as the invariant control points  $\mathbf{P}$  and the corresponding invariant control points  $a_j$  in the task  $\mathbf{P}'$ . Note that if we can define each mapping from  $\mathbf{P}$  to  $\mathbf{P}'$  as a function  $f$ , to minimize the distortion of feature in space is equivalent to minimize the following energy function [34]:

$$E = \sum_{w=1}^k \| \mathbf{P}'_w - \mathbf{P}_w \| + \lambda E_f$$

where

$$E_f = \int \int_{R^2} (f''_{xx} + 2f''_{xy} + f''_{yy}) dx dy,$$

$\lambda$  is the regularization parameter.

Note that the regularization parameter  $\lambda$  is a trade-off between the exact matching of points and the smoothness. This parameter is particularly useful in the presence of a noise. It is clear that the quality of the algorithm is critically dependent on the values of the regularization parameter. However, the exact way to calculate values of  $\lambda$  is not obvious.

It is natural to consider different intelligent algorithms for the prediction values of the regularization parameter. Since  $\lambda$  is useful in the presence of a noise, we can also consider different Kalman filters. In our computational experiments, we consider genetic algorithms (GA), multilayer perceptron networks (MPN) with gradient learning algorithm, recurrent neural networks

(RN) gradient learning algorithm, 4-order Runge Kutta neural networks (see e.g. [35]), unscented Kalman filters (UKF) (see e.g. [36]), extended Kalman filters (EKF) (see e.g. [36]), adaptive unscented Kalman filters (AUKF) (see e.g. [36]), adaptive neuro-fuzzy extended Kalman filter (ANFEKF) (see e.g. [37]). Runge Kutta neural networks we consider for multilayer perceptron networks with the gradient learning algorithm (RKN-P-GLA) (see [35]), the nonlinear recursive least square learning algorithm (RKN-P-NLSLA) (see [35]), the zero-order nonlinear recursive least square learning algorithm (RKN-P-0-NLSLA) (see [35]), the first-order nonlinear recursive least square learning algorithm (RKN-P-1-NLSLA) (see [35]) and for recurrent neural networks with the gradient learning algorithm (RKN-R-GLA) (see [35]), the nonlinear recursive least square learning algorithm (RKN-R-NLSLA) (see [35]), the zero-order nonlinear recursive least square learning algorithm (RKN-R-0-NLSLA) (see [35]), the first-order nonlinear recursive least square learning algorithm (RKN-R-1-NLSLA) (see [35]).

Let

$$M = \{m_l \mid m_l = (x_l, y_l), 1 \leq l \leq p\}$$

be the set of spatial feature points that represents the template path. Let

$$M_0 = \{m_l^0 \mid m_l^0 = (x_l^0, y_l^0), 1 \leq l \leq p\}$$

be the set of spatial feature points that represents the path for  $\lambda = \text{const}$ . Let

$$M_A = \{m_l^A \mid m_l^A = (x_l^A, y_l^A), 1 \leq l \leq p\}$$

be the set of spatial feature points that represents the path for  $\lambda$  controlled by the algorithm  $A$ . It is clear that we can use the average value  $\text{avr}V[A]$  of

$$V[A] = \frac{\sum_{l=1}^p (x_l^A - x_l)^2 + \sum_{l=1}^p (y_l^A - y_l)^2}{\sum_{l=1}^p (x_l^0 - x_l)^2 + \sum_{l=1}^p (y_l^0 - y_l)^2}$$

as a measure of the quality of the algorithm  $A$ . Selected experimental results are given in Tab. 1.

## 11 SAT Solvers

It is well-known that usage of efficient satisfiability algorithms and encoding different hard problems as instances of satisfiability has caused considerable interest recently (see e.g. [38, 39, 40, 41]). However, most of them require the use of a supercomputer. For the module of onboard equipment of mobile vehicles we need a SAT-solver that requires significantly less computing resources. In particular, GSAT is a well-known satisfiability algorithm [42]



$p$	$10^2$	$10^3$	$10^4$	$10^5$
GA	0.271	0.306	0.324	0.019
MPN	0.442	0.439	0.431	0.443
RN	0.445	0.426	0.411	0.398
UKF	0.892	0.843	0.813	0.768
EKF	0.724	0.714	0.699	0.691
AUKF	0.753	0.704	0.641	0.521
ANFEKF	0.762	0.695	0.633	0.508
RKN-P-GLA	0.322	0.165	0.139	0.118
RKN-P-NLSLA	0.314	0.158	0.129	0.107
RKN-P-0-NLSLA	0.316	0.161	0.132	0.104
RKN-P-1-NLSLA	0.296	0.144	0.111	0.097
RKN-R-GLA	0.196	0.114	0.091	0.083
RKN-R-NLSLA	0.175	0.099	0.081	0.066
RKN-R-0-NLSLA	0.173	0.093	0.077	0.061
RKN-R-1-NLSLA	0.171	0.078	0.064	0.059

Table 1: The dependence of the quality of different algorithms from the number  $p$  of spatial feature points.

(see also [43, 44]). This algorithm is extensively used for the solution of the satisfiability problem. It should be noted that GSAT applies only to clausal formulas. The notion of score plays a key role in GSAT.

Note that some modification of the score function are considered. However, usually, the score function is the number of clauses of formula which are falsified by the truth assignment (see e.g. [45]). A modification of GSAT with adaptive score function was proposed in [44]. In particular, a genetic algorithm was used for prediction of values of score function. In this paper we consider a new modification of GSAT. In particular, we use Runge Kutta neural networks (see e.g. [35]) for the calculation of values of the adaptive score function.

Let  $f(z[1], \dots, z[n])$  be a CNF. Let  $T$  be a truth assignment for the variables of  $f$ . Let  $\#occ(k, f, z[i])$  be the number of positive occurrences of  $z[i]$  in clauses of type  $x_1 \vee \dots \vee x_k$ ,  $\#occ(k, f, \neg z[i])$  be the number of occurrences of  $\neg z[i]$  in clauses of type  $x_1 \vee \dots \vee x_k$ . Let  $S(T, f)$  be the set of clauses of  $f$  which are falsified by  $T$ . Following [33], we can assume that

$$score(T, f) = \sum_{x_1 \vee \dots \vee x_p \in S(T, f)} \sum_{1 \leq i \leq q, 1 \leq j \leq p} (\alpha(\#occ(i, f, x_j)) + \beta(\#occ(i, f, \neg x_j))).$$

Also, following [44], we can use a genetic algorithm for prediction of values of  $\alpha$  and  $\beta$ .

It is clear that we can consider  $\alpha$  and  $\beta$  as functions of time. In this case, we can consider  $\alpha(t)$  and  $\beta(t)$  as dynamical systems  $\alpha' = F(\alpha, t)$  and  $\beta' = G(\beta, t)$  for some unknown functions  $F$  and  $G$ .

Clearly, a genetic algorithm for prediction of values of  $\alpha$  and  $\beta$  can be used to calculate initial conditions  $\alpha(t) = \alpha_0$  and  $\beta(t) = \beta_0$ .

Let  $a(t)$  be a solution trajectory of the system  $\alpha' = F(\alpha, t)$ . To predict  $a(t)$ , we consider the 4-order Runge Kutta neural network that described by

$$a(t+1) = a(t) + \frac{1}{6}h(k_\alpha[0] + 2k_\alpha[1] + 2k_\alpha[2] + k_\alpha[3]),$$

where

$$k_\alpha[0] = N_\alpha(a(t); w_\alpha);$$

$$k_\alpha[1] = N_\alpha(a(t) + \frac{1}{2}h_\alpha k_\alpha[0]; w_\alpha);$$

$$k_\alpha[2] = N_\alpha(a(t) + \frac{1}{2}h_\alpha k_\alpha[1]; w_\alpha);$$

$$k_\alpha[3] = N_\alpha(a(t) + h_\alpha k_\alpha[2]; w_\alpha),$$

where the neural network  $N_\alpha(y, w_\alpha)$ , with input  $y$  and weights  $w_\alpha$ , is the multilayer perceptron network,  $h_\alpha$  is the time step (see [35]). Similarly, let  $b(t)$  be a solution trajectory of the system  $\beta' = G(\beta, t)$ . To predict  $b(t)$ , we consider the 4-order Runge Kutta neural network that described by

$$b(t+1) = b(t) + \frac{1}{6}h(k_\beta[0] + 2k_\beta[1] + 2k_\beta[2] + k_\beta[3]),$$

where

$$k_\beta[0] = N_\beta(b(t); w_\beta);$$

$$k_\beta[1] = N_\beta(b(t) + \frac{1}{2}h_\beta k_\beta[0]; w_\beta);$$

$$k_\beta[2] = N_\beta(b(t) + \frac{1}{2}h_\beta k_\beta[1]; w_\beta);$$

$$k_\beta[3] = N_\beta(b(t) + h_\beta k_\beta[2]; w_\beta),$$

where the neural network  $N_\beta(y, w_\beta)$ , with input  $y$  and weights  $w_\beta$ , is the multilayer perceptron network,  $h_\beta$  is the time step. For 4-order Runge Kutta neural networks, we consider a gradient learning algorithm.

In our experiments, we consider GSAT with score function  $score(T, f)$  (GSAT-ASF) and GSAT with score function

$$Score(T, f, 1) = \sum_{x_1 \vee \dots \vee x_p \in S(T, f)} \sum_{1 \leq i \leq q, 1 \leq j \leq p} (\alpha(\#occ(i, f, x_j)) + \beta(\#occ(i, f, \neg x_j)));$$

$$Score(T, f, t+1) = \sum_{x_1 \vee \dots \vee x_p \in S(T, f)} \sum_{1 \leq i \leq q, 1 \leq j \leq p} ((a(t) + \frac{1}{6}h(k_\alpha[0] + 2k_\alpha[1] + 2k_\alpha[2] + k_\alpha[3]))(\#occ(i, f, x_j)) + (b(t) + \frac{1}{6}h(k_\beta[0] + 2k_\beta[1] + 2k_\beta[2] + k_\beta[3]))(\#occ(i, f, \neg x_j)))$$

(GSAT-ASF-RK) for 3-CNFs. Selected experimental results are given in Tables 2 and 3.

	$G = 10^4$	$G = 10^5$	$G = 10^6$
GSAT-ASF	91 %	92 %	93 %
GSAT-ASF-RK	87 %	90 %	92 %

Table 2: A number of solved formulas for GSAT-ASF and GSAT-ASF-RK where  $G$  is a number of generations of genetic algorithm.

	$G = 10^4$	$G = 10^5$	$G = 10^6$
$\frac{T_{GSAT-ASF-RK}}{T_{GSAT-ASF}}$	84 %	33 %	14 %

Table 3: The performance of GSAT-ASF and GSAT-ASF-RK where  $G$  is a number of generations of genetic algorithm,  $T_{GSAT-ASF}$  is the average time of solution of GSAT-ASF,  $T_{GSAT-ASF-RK}$  is the average time of solution of GSAT-ASF-RK.

**ACKNOWLEDGEMENTS.** The work was partially supported by Analytical Departmental Program “Developing the scientific potential of high school”, RFBR, research project No. 13-01-00048 a, and Ural Federal University development program with the financial support of young scientists.

## References

- [1] A. Gorbenko and V. Popov, Programming for Modular Reconfigurable Robots, *Programming and Computer Software*, 38 (2012), 13–23.
- [2] A. Gorbenko, Graph-Theoretic Models for the Module of Safe Planning for Control Systems of Mobile Robots, *Advanced Materials Research*, 683 (2013), 737–740.
- [3] A. Gorbenko and V. Popov, Self-Learning of Robots and the Model of Hamiltonian Path with Fixed Number of Color Repetitions for Systems of Scenarios Creation, *Advanced Materials Research*, 683 (2013), 909–912.
- [4] R. J. Duro, M. Grana, and J. de Lope, On the Need of Hybrid Intelligent Systems in Modular and Multi Robotics, *Lecture Notes in Computer Science*, 5271 (2008), 641–648.
- [5] B. Sights, G. Ahuja, G. Kogut, E.B. Pacis, H.R. Everett, D. Fellars, and S. Hardjadinata, Modular robotic intelligence system based on fuzzy reasoning and state machine sequencing, *Proceedings of the SPIE*, 6561 (2007), 65610T.
- [6] A. Gorbenko and V. Popov, Robot Self-Awareness: Usage of Co-training for Distance Functions for Sequences of Images, *Advanced Studies in Theoretical Physics*, 6 (2012), 1243–1246.
- [7] A. Gorbenko and V. Popov, Robot Self-Awareness: Formulation of Hypotheses Based on the Discovered Regularities, *Applied Mathematical Sciences*, 6 (2012), 6583–6585.
- [8] A. Gorbenko and V. Popov, Robot Self-Awareness: Occam’s Razor for Fluents, *International Journal of Mathematical Analysis*, 6 (2012), 1453–1455.
- [9] A. Gorbenko, V. Popov, and A. Sheka, Robot Self-Awareness: Exploration of Internal States, *Applied Mathematical Sciences*, 6 (2012), 675–688.
- [10] A. Gorbenko, V. Popov, and A. Sheka, Robot Self-Awareness: Temporal Relation Based Data Mining, *Engineering Letters*, 19 (2011), 169–178.
- [11] A. Gorbenko and V. Popov, Robot’s Actions and Automatic Generation of Distance Functions for Sequences of Images, *Advanced Studies in Theoretical Physics*, 6 (2012), 1247–1251.

- [12] A. Gorbenko and V. Popov, Anticipation in Simple Robot Navigation and Finding Regularities, *Applied Mathematical Sciences*, 6 (2012), 6577–6581.
- [13] A. Gorbenko and V. Popov, Anticipation in Simple Robot Navigation and Learning of Effects of Robot's Actions and Changes of the Environment, *International Journal of Mathematical Analysis*, 6 (2012), 2747–2751.
- [14] A. Gorbenko and V. Popov, The Force Law Design of Artificial Physics Optimization for Robot Anticipation of Motion, *Advanced Studies in Theoretical Physics*, 6 (2012), 625–628.
- [15] A. Gorbenko and V. Popov, Self-Learning Algorithm for Visual Recognition and Object Categorization for Autonomous Mobile Robots, *Lecture Notes in Electrical Engineering*, 107 (2012), 1289–1295.
- [16] A. Gorbenko and V. Popov, Multi-agent Path Planning, *Applied Mathematical Sciences*, 6 (2012), 6733–6737.
- [17] A. Gorbenko and V. Popov, Footstep Planning for Humanoid Robots, *Applied Mathematical Sciences*, 6 (2012), 6567–6571.
- [18] A. Gorbenko and V. Popov, Computational Experiments for the Problem of Footstep Planning for Humanoid Robots, *Applied Mathematical Sciences*, 7 (2013), 2357–2372.
- [19] A. Gorbenko and V. Popov, Restricted Common Superstrings, *Applied Mathematical Sciences*, 7 (2013), 2335–2339.
- [20] A. Gorbenko and V. Popov, The Minimum k-Cover Problem, *Applied Mathematical Sciences*, 7 (2013), 2347–2352.
- [21] A. Gorbenko and V. Popov, The Shortest Common Superstring Problem, *Applied Mathematical Sciences*, 7 (2013), 2353–2356.
- [22] A. Gorbenko and V. Popov, The Shortest Common Parameterized Supersequence Problem, *Applied Mathematical Sciences*, 7 (2013), 2373–2380.
- [23] V. Popov, On Reoptimization of the Shortest Common Superstring Problem, *Applied Mathematical Sciences*, 7 (2013), 1195–1197.
- [24] A. Gorbenko and V. Popov, The Swap Common Superstring Problem, *Applied Mathematical Sciences*, 7 (2013), 609–614.
- [25] A. Gorbenko and V. Popov, On Multiple Occurrences Shortest Common Superstring Problem, *Applied Mathematical Sciences*, 7 (2013), 641–644.

- [26] A. Gorbenko and V. Popov, Multiple Occurrences Shortest Common Superstring Problem, *Applied Mathematical Sciences*, 6 (2012), 6573–6576.
- [27] V. Popov, Approximate Periods of Strings for Absolute Distances, *Applied Mathematical Sciences*, 6 (2012), 6713–6717.
- [28] A. Gorbenko and V. Popov, The set of parameterized k-covers problem, *Theoretical Computer Science*, 423 (2012), 19–24.
- [29] V. Popov, The Approximate Period Problem, *IAENG International Journal of Computer Science*, 36 (2009), 268–274.
- [30] V. Popov, Multiple genome rearrangement by swaps and by element duplications, *Theoretical Computer Science*, 385 (2007), 115–126.
- [31] V. Yu. Popov, Computational complexity of problems related to DNA sequencing by hybridization, *Doklady Mathematics*, 72 (2005), 642–644.
- [32] V. Popov, The approximate period problem for DNA alphabet, *Theoretical Computer Science*, 304 (2003), 443–447.
- [33] Y. Wu and Y. Demiris, Towards One Shot Learning by Imitation for Humanoid Robots, *IEEE International Conference on Robotics and Automation*, (2010), 2889–2894.
- [34] F. Bookstein, Principal warps: Thin-plate splines and the decomposition of deformations, *IEEE Transactions on pattern analysis and machine intelligence*, 11 (1989), 567–585.
- [35] Y.-J. Wang and C.-T. Lin, Runge Kutta neural network for identification of continuous systems, *Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics*, (1998), 3277–3282.
- [36] J. Han, Q. Song, and Y. He, Adaptive Unscented Kalman Filter and Its Applications in Nonlinear Control, in: V. M. Moreno and A. Pigazo (eds.), *Kalman Filter: Recent Advances and Applications*, I-Tech, Vienna, Austria, 2009, 1–24.
- [37] R. Havangi, M. A. Nekoui, and M. Teshnehlab, Adaptive Neuro-Fuzzy Extended Kalman Filtering for Robot Localization, *International Journal of Computer Science Issues*, 7 (2010), 15–23.
- [38] A. Gorbenko and V. Popov, SAT Solvers for the Problem of Sensor Placement, *Advanced Studies in Theoretical Physics*, 6 (2012), 1235–1238.
- [39] V. Popov, Genetic Algorithms with Exons and Introns for the Satisfiability Problem, *Advanced Studies in Theoretical Physics*, 7 (2013), 355–358.

- [40] V. Popov, A Genetic Algorithm with Expansion Operator for the 3-Satisfiability Problem, *Advanced Studies in Theoretical Physics*, 7 (2013), 359-361.
- [41] A. Gorbenko and V. Popov, Task-resource Scheduling Problem, *International Journal of Automation and Computing*, 9 (2012), 429-441.
- [42] B. Selman, H. Levesque, and D. Mitchell, A new method for solving hard satisfiability problems, *Proceedings of the Tenth National Conference on Artificial Intelligence*, (1992), 440-446.
- [43] I.P. Gent and T. Walsh, Towards an Understanding of Hill-Climbing Procedures for SAT, *Proceedings of the Eleventh National Conference on Artificial Intelligence*, (1993), 28-33.
- [44] V. Popov, GSAT with Adaptive Score Function, *Advanced Studies in Theoretical Physics*, 7 (2013), 363-366.
- [45] R. Sebastiani, Applying GSAT to non-clausal formulas, *Journal of Artificial Intelligence Research*, 1 (1993), 309-314.

**Received: June 15, 2013**