RIGA TECHNICAL UNIVERSITY

Faculty of Power and Electrical Engineering
Institute of Industrial Electronics and Electrical Engineering
Department of Industrial Electronics and Electrical Technology

Andrew Mor-Yaroslavtsev
Computer Control of Electrical Technology Study Program

Modeling the Intelligent Electrical Transport Control Systems with Immune Algorithms

PhD Thesis Summary

Scientific advisor:
Dr. Sc. ing., professor
A. Levchenkov

Riga – 2013

UDK 656.2:004.89(043.2)
Mo 650 m

Mor-Yaroslavtsev A. Modeling the Intelligent Electrical Transport Control Systems with Immune Algorithms. PhD Thesis Summary.-R.: RTU, 2013.-32 lpp.

Published in accordance with RTU EEF IEEI resolution from 31st July, 2013, protocol No. 72.

This work was supported by the European Social Fund project „Support for the implementation of doctoral studies at Riga Technical University”.

ISBN 978-9934-10-498-5

DOCTORAL THESIS
IS SUBMITTED FOR THE PROMOTION OF DOCTOR DEGREE
OF ELECTRICAL ENGINEERING SCIENCES
AT RIGA TECHNICAL UNIVERSITY

The Doctoral thesis is openly defended on _____ ___ 2013 at Riga Technical University, Faculty of Electrical and Power Engineering, Kronvalda blvd. 1, …… room.

OFFICIAL REVIEWERS

Professor, Dr.habil.sc.ing. Imants Matīss
Latvian Academy of Sciences

Professor, Dr. habil.sc.ing. Andris Šnīders
Latvian University of Agriculture

Professor, Dr.sc.ing. Žilvinas Bazaras
Kaunas University of Technology (Lithuania)

ACKNOWLEDGMENT

I acknowledge that the work I submitted to Riga Technical University for the promotion to Doctoral degree in engineering sciences is my own. No part of this work is submitted to other universities or other institutions for any degree obtaining.

Andrew Mor-Yaroslavtsev ……………………………. (Signature)

Date: ……………………..

The thesis is written in Latvian, it contains an introduction, five chapters, conclusions, list of references, one appendix, 54 figures and illustrations, tables on the total of 169 pages. The list of literature refers to 71 sources.

Contents

TOPICALITY OF THE WORK

Nowadays the human factor plays an important role in the transport control system, including rail transport with electric drive and electrical signaling systems and rail crossings. In some cases, the existing safety systems are able to prevent accidents. Accidents on level crossings are taking place all over the world. In any country where there is a railway, there were accidents which caused death or injury to people. Therefore the improvement of railway safety is an actual topic for the national economy and welfare.

The artificial immune system (AIS) is a modern method based on the fundamental principles of biological immune systems, designed for the internal protection of the system. This method is implementable by means of computer control of electrical technology — programmable embedded devices, and will prevent accidents in the railways by reducing the human factor. It is necessary to develop specific algorithms for the artificial immune system for vehicles to warn people timely about the dangerous situation, to offer improvement solutions and to automatically stop the vehicle motion in order to avoid a collision.

THE GOAL OF THE RESEARCH

The objective of the Ph.D. thesis is to develop control methods based on artificial immune algorithms for the intelligent transport safety control system with embedded electronic devices, which help to prevent collisions of railway vehicles with other objects.

To achieve the goal the following tasks were formulated:

- Classify and compare AIS and evolutionary algorithms;
- Study abilities and working principles of AIS;
- Define the relevant possible disruptions to the electrical railway transport's movement;
- Develop the structure of the intelligent rolling stock safety control system and its working algorithm;
- Implement immune algorithms for running on computers and controllers;
- Choose modules and sensors for the controllers suitable for the task;
- Implement the data flow between controllers and control center computers, as well as recording this flow into the database;
- Compare the implementation and usage costs of the offered embedded devices;
- Conduct experiments in the laboratory and in real conditions;
- Analyze the experiments' results.

SCIENTIFIC NOVELTY OF THE RESEARCH

- A model of railway transport control using immune algorithms to avoid collisions was developed;
- A structure of intelligent rolling stock safety control system and its algorithm was designed;
- An immune algorithm for solving a multi-criteria railway transport safety problem was developed;
- A model of railway transport system functional interaction was developed, which describes how different separate transport elements interact and allows modeling the transport system's processes, including the intelligent control of electromechanical processes.

MAIN RESULTS

- Methods and means were developed to avoid collisions of railway transport with other objects;
- Prototypes of embedded devices for locomotives and traffic light and level crossing relay boxes were developed, as well as software for intelligent transport safety control, using data about vehicle position, speed and control relay status;
- Operation of algorithms and devices was experimentally tested in laboratory and real conditions;
- Computer and controller software, as well as connections to the railway objects was described.

APPLIED IMPORTANCE OF THE RESEARCH

The algorithms and railway transport safety control system design offered in the thesis can be used to enhance railway vehicles' safety control. Using the developed algorithms it is possible to avoid collisions between railway vehicles and other objects, or avoid crossing the red traffic light.

MEANS AND METHODS OF THE RESEARCH

The description of the system elements uses set theory, system and process analysis, theory of evolutionary algorithms, artificial immune systems; to evaluate the modeling results statistical analysis methods are used. To develop the embedded device prototypes and software for them, the author used Arduino Uno and Mega controllers and object-oriented programming concepts.

APPROBATION OF THE RESEARCH

1. 50th RTU International Scientific Conference, Riga, Latvia, October 14–15, 2009. Report
2. 8th International Symposium „Topical Problems in the Field of Electrical and Power Engineering“, Pärnu, Estonia, January 11–16, 2010. Report
3. RTU Innovation and New Technology Conference, Riga, Latvia, September 24, 2010. Poster report
4. 51st RTU International Scientific Conference, Riga, Latvia, Ocober 14, 2010. 2 reports
5. „Transport Systems Telematics 2010”, Katovice, Poland, October 19–24, 2010.
6. „VDE Congress 2010: E-Mobility”, Leipzig, Germany, November 8–9, 2010. Poster report
7. „2010 Second Global Congress on Intelligent Systems”, Wuhan, China, December 13–19, 2010. Report and chairing a session
8. „10th International Symposium „Topical Problems in the Field of Electrical and Power Engineering“”, Pärnu, Estonia, January 10–15, 2011. Report
9. „Intelligent Technologies in Logistics and Mechatronics Systems (ITELMS) 2011”, Panevezys, Lithuania, May 5, 2011. g. Report
10. 52nd RTU International Scientific Conference, Riga, Latvia, October 14, 2011. 2 reports
11. Transport Means 2011, Kaunas, Lithuania, October 20, 2011. 2 reports
12. TELFOR 2011, Belgrad, Serbia, November 22–24, 2011. Report

13. RTU Innovation and New Technology Conference, Riga, Latvia, April 3, 2012. Poster report
14. ITELMS 2012, Panevezys, Lithuania, May 4, 2012. Report
15. International Symposium of Electrical Engineering, Ronīši, Latvia, May 25, 2012. Report
16. 25th European Conference on Operational Research, Vilnius, Lithuania, July 8–11, 2012. Report
17. 53rd RTU RTU International Scientific Conference, Riga, Latvia, October 10, 2012. Report
18. ITELMS 2013, Panevezys, Lithuania, May 23, 2013. Report

AUTHOR'S PUBLICATIONS

1. Mors-Jaroslavcevs A. Electric Engine Diagnostics Using Artificial Immune Systems // Proceedings of 50th RTU International Scientific Conference, Riga, Latvia, October 14–15, 2009
2. Mors-Jaroslavcevs A., Ļevčenkovs A. Modeling of Artificial Immune Systems for Railway Electric Transport Control // Proceedings of 8th International Symposium „Topical Problems in the Field of Electrical and Power Engineering“,Pärnu, Estonia, January 11–16, 2010
3. Mors-Jaroslavcevs A., Ļevčenkovs A. Immune negative selection algorithm for railway electric vehicle fault detection system // Proceedings of 51st RTU International Scientific Conference, Riga, Latvia, October 14–15, 2010.
4. Mors-Jaroslavcevs A., Ļevčenkovs A., Ribickis L. Modeling of hybrid railway electric vehicle safety control system using artificial immune systems // Proceedings of VDE Congress 2010, Leipzig, Germany, November 8–9, 2010.
5. Mors-Jaroslavcevs A., Ļevčenkovs A., Ribickis L. Structure of automated railway electric vehicle safety control system // Proceedings of GCIS 2010, Wuhan, China, December 16.-17., 2010.
6. A. Mor-Yaroslavtsev, A. Levchenkov. Modeling the integration of expert systems into railway electric transport safety control. // Proceedings of 10th International Symposium „Topical Problems in the Field of Electrical and Power Engineering“, Pärnu, Estonia, January 10-15, 2011
7. Mors-Jaroslavcevs A., Ļevčenkovs A. Combining immune algorithms for an intelligent rolling stock safety system // Proceedings of 52nd RTU International Scientific Conference, Riga, Latvia, October 14, 2011
8. Mors-Jaroslavcevs A., Ļevčenkovs A. Railway electric vehicle diagnostics with an algorithm for self-nonself discrimination in artificial immune systems // Proceedings of Transport Means 2011, Kaunas, Lithuania, October 20, 2011
9. Mors-Jaroslavcevs A., Ļevčenkovs A. Intelligent Embedded Rolling Stock Safety Devices Using an Immune Clonal Selection Algorithm // Proceedings of Transport Means 2011, Kaunas, Lithuania, October 20, 2011
10. Mor-Yaroslavtsev A., Levchenkov A. Rolling Stock Location Data Analysis Using an Immune Algorithm on an Intelligent Embedded Device // Proceedings of TELFOR 2011, Belgrade, Serbia, November 22–24, 2011
11. Levchenkov A., Gorobetz M., Mor-Yaroslavtsev A. Evolutionary Algorithms in Embedded Intelligent Devices Using Satellite Navigation for Railway Transport// Infrastructure Design, Signalling and Security in Railway, Xavier Perpinya (Ed.), ISBN: 978-953-51-0448-3, InTech, 2012

12. Mor-Yaroslavtsev A., Levchenkov A. Self-learning algorithms for an embedded device using location data on a rolling stock// Proceedings of ITELMS 2012, Panevėžys, Lithuania, May 4, 2012
13. Potapovs A., Mor-Yaroslavtsev A., Levchenkov A., Gorobetz M. Smooth Braking of Train Using Adaptive Control Algorithms on Embedded Devices // Proceedings of 53rd RTU International Scientific Conference, Riga, Latvia, October 10, 2012

STRUCTURE OF THE THESIS

The thesis consists of introduction, five chapters, conclusions, list of literature sources and annexes.

The first chapter of the thesis the goal and tasks are formulated, an existing electrified railway is analyzed, and a mathematical model of the existing and proposed system is created that defines the possible disturbances in movement.

The second chapter of the thesis deals with the literature analysis of known evolutionary algorithms, including genetic algorithms, immune algorithms and neural networks. The chapter also describes some positioning devices, on-board equipment, and has an analysis of evolutionary algorithms.

The third chapter of the thesis contains the developed algorithms for rail and road collision avoidance, defined multiple criteria optimization objective function, as well as possibilities of immune algorithm implementation in the embedded devices.

In the fourth chapter there are the computer experiments with statistical hypothesis testing for evolutionary algorithms, a comparison of three artificial intelligence techniques for collision avoidance problem and proof of effectiveness of the immune algorithm.

The fifth chapter of the thesis describes experiments with the developed embedded device prototypes, analyzes and evaluates the results.

1. STATEMENT OF THE PROBLEM

During journey the rolling stock driver may experience many undesirable situations and have to make decisions on how solve them. The situations may include such examples as:

- the last car from the flow is still on the level crossing 25 seconds before the train arrival, while the safety regulations require the crossing to be cleared at least 35 seconds before train arrival;
- a daredevil is running across the tracks somewhere in the urban zone;
- there is a red signal on the railway traffic light;
- there is a wide but harmless rod lying between the tracks, etc.

Each of these situations requires different actions or no action at all. The driver may have to apply brakes, speed up, continue the steady movement and in any case communicate the information to the control center and other drivers.

The desired result conforms to at least two require-ments:

- there are no casualties;
- the train is on schedule.

A common situation is illustrated on Fig. 1.1, where L is a locomotive and I is an invading object on tracks.

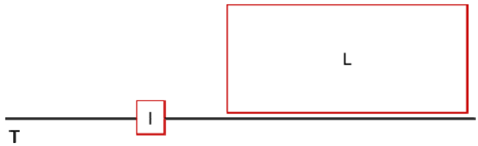

Fig. 1.1. A common unwanted situation on the railway tracks.

The author offers the intelligent rolling stock safety system functional design which is presented on Fig. 1.2.

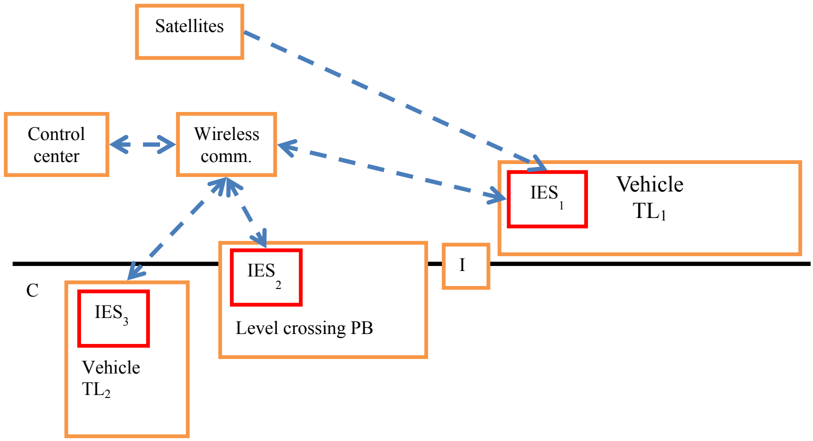

Fig. 1.2. The intelligent rolling stock safety system functional design

The vehicles TL and the level crossing PB host embedded devices IES, each of these contains a wireless modem M, positioning module G which receives data from positioning satellites ST, data processing module AIS, immune detector database DBD and control cell database DBC [author's publication 10].

The invading object I is picked up by sensors S and the data is transmitted to the nearest cell tower CT, which relays it to the control center CC and nearest locomotives wireless modems M. Through the same modem the locomotive receives data about closest neighbors' rolling stock position and status, railway segment profile and maximum allowed speed.

Depending on the results of control cell maturation the AIS module in the vehicle makes a decision and executes it by sending a control signal or displaying an alert to the driver. The information is also communicated to the device on a level crossing LC through a similar modem M.

The considered system is limited by two vehicles and an infrastructure object such as a traffic light or a level crossing [author's publication 12].

Let $U \in \Re$ be the problem space which includes all the possible sets of parameter values or "situations",

P — set of known safe situations,

$S(t)$ — current situation which changes with time t,

$D = \{D_1, D_2, \ldots, D_n\}$ — set of detectors which is the result of primary learning process,

$C = \{C_{D1(1)}, C_{D1(2)}, \ldots, C_{D1(p)}, C_{D2(1)}, \ldots, C_{Dn(p)}\}$ — set of control cells affiliated to detectors,

$E = \{E_1, E_2, \ldots, E_m\}$ — set of encountered situations,

$W = \{W_{E1D1}, W_{E1D2}, \ldots, W_{E1Dn}, W_{E2D1}, \ldots, W_{EmDn}\}$ — set of detector weights in different situations;

Distance between two points on a greater sphere:

$$d = 2R\arcsin\left(\sqrt{\sin^2\left(\frac{lat_A - lat_B}{2}\right) + \cos(lat_A)\times\cos(lat_B)\times\sin^2\left(\frac{lon_A - lon_B}{2}\right)}\right) \quad (1.1)$$

Embedded devicecs: $IES = \{ies_1, ies_2, \ldots, ies_n\}$ $ies_1, \ldots, ies_2 \in PB$

Conditions introduced by using embedded devices:

- Operation in real time: $T = \{t_1, t_2, \ldots, t_n\}$
- Size of the device: $V = \{v_1, v_2, \ldots, v_n\}$; $v_i = x_i \times y_i \times z_i$; V^* — min V
- Time to test the prototypes: $T_{test1} < T_{test2}$
- Energy consumption: $E = \{e_1, e_2, \ldots, e_n\}$; E^* — min E
- Installation and usage costs: $IZ = \{iz_1, iz_2, \ldots, iz_n\}$; IZ^* — min IZ
- Limited available computing speed and memory: $P = \{p_1, p_2, \ldots, p_n\}$; P^* — min P; $AT = \{at_1, at_2, \ldots, at_n\}$; AT^* — min AT

- A_n — a particular algorithm;
- $T_{An} \rightarrow \min$ — completion time for the algorithm;
- $Pr_{An} \rightarrow \min$ — share of avoided collisions as a result of using the algorithm;
- $E_{An} \rightarrow \min$ — energy consumption while running the algorithm;
- $V_B = (x_B \times y_B \times z_B) \rightarrow \min$ — embedded device dimensions;
- $ST_B \rightarrow \min$ — interruptions in the device or algorithm operation;
- $IP_B \rightarrow \min$ — device installation and usage costs.

- The hypothesis: the immune algorithm will be the fastest to complete while avoiding at least as much or more collisions than other algorithms and using as much or less energy.

The fitness function: $F(T_{An}, Pr_{An}, E_{An}, V_B, ST_B, IP_B) \rightarrow \min$

The intelligent rolling stock safety system general algorithm offered by the author:

1. Fill in the initial values D for DBD by running the negative selection training routine.
2. Run in real time the detection routine using the negative selection algorithm.
3. Determine the possible situation identifiers (detectors which matched above a given threshold).
4. Assign weights to the detectors based on their «distance» to the situation.
5. Retrieve a population of control cells C from DBC which are related to the activated detectors [author's publication 9].
6. Run the control cell maturation routine using the clonal selection algorithm [author's publication 9].
7. Execute the found optimal solution.
8. Communicate the information to the control centre CC.
9. Continue from step 2.

The most feasible way to implement such a safety system would be, like in case of hybrid IDS, through the two phases of anomaly detection and determination of their type to draw a conclusion. In this case the incoming data from the sensors is the set of antigens. The data includes but is not limited to speed, acceleration, voltage, rotation and operational temperature.

2. LITERATURE REVIEW

Artificial immune systems (AIS) use evolutionary data processing paradigm based on biological immune systems. It differs from computational immunology which models biological immune systems [author's publication 2].

Immune algorithms are mainly used to solve anomaly recognition, data collection and analysis tasks. From the computational point of view immune systems' most interesting features are self-learning, diversity mainte-nance and memory.

The problem is represented as an antigen and solution candidates as antibodies which are randomly generated from the library of available solutions or genes. The evaluation of affinity or degree of binding between the antigen and the antibody is similar to complementarity level in biological IS and it defines the fate of each indi-vidual antibody as well as termination of the whole algo-rithm.

Individual antibodies are replaced, cloned and hypermutated until satisfactory level of affinity is reached. Partial replacement of the solutions' population with fresh randomly generated candidates maintains diversity which allows solving a wider set of problems. The probability of cloning or hypermutating a candidate depends on its affinity.

The most relevant features of immune algorithms are:

- diversity maintenance,
- memory about the past decisions,
- detection of previously unknown but related el-ements,
- scalability,
- noise rejection,
- classifying ability.

As mentioned before, AIS can be used to solve different data analysis tasks. In the traveling salesman problem each city to be visited could be labeled as an antigen and the set iteratively combined with the antibody network, simulating antigen intrusion in the organism and driving through cities in the random order.

Network intrusion detection is related to unauthorized access to computer systems connected to the network and the problem is solved using anomaly or misuse patterns detection. Anomaly detection systems build a model of normal system activity and then regard deviations from this as potential intrusions, while misuse detection systems look for known attack patterns by signature matching. The key advantage of anomaly detection systems is their ability to detect novel attack patterns for which no signature exists, while their most notable disadvantage is a larger false positive rate. Already being close to the immune approach, by introducing its memory feature such systems could also provide further information about the consequences of the attack and possible future actions instead of simply reporting the actions.

AIS are modeled after biological IS and carry the terms of antigens and antibodies. They can be modeled using the shape-space concept (see Fig. 2.1). The shape-space *S* allows defining antigens, receptors and their interactions in a quantitative way [author's publication 1].

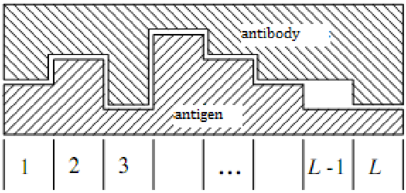

Fig. 2.1. A shape-space model of an antigen and an antibody

Like chromosomes in the evolutionary algorithms, the element's shape is defined by a string *m* which contains its coordinates:

$$m = \{m_1, m_2, \ldots, m_L\} \in S^L \qquad (2.1),$$

where *S* usually is defined on a set of real numbers – $S^L \in R$. Depending on a problem being solved it also could be a set of integers or binary numbers – $m \in Z^L$ or $m \in \{0,1\}^L$.

The affinity of an antigen–antibody pair is related to their distance in the shape-space *S* and can be estimated using any distance measure between the two attribute strings. The distance between an antigen, *Ag*, and an antibody, *Ab*, can be defined, for example, using a general class of Minkowski distance measures:

$$D_M(Ag, Ab) = \sqrt[p]{\sum_{i=1}^{L} |Ag_i - Ab_i|^p} \qquad (2.2).$$

By varying the value of the parameter *p* a suitable measure of distance can be obtained.

Negative selection is the paradigm describing the evolution of the T-lymphocytes where they are randomly generated and learn to recognize all except the self structures, specific to the host. Negative selection algorithms need training samples only from one class (self, normal), thus, they are especially suited for the tasks such as novelty, anomaly or change detection including those in engines and other devices.

The key advantage of anomaly detection systems is their ability to detect novel attack patterns for which no signature exists, while their most notable disadvantage is a larger false positive rate.

The algorithm:

1. Define a set *S* which needs to be monitored and the set *P* of known self elements in a feature space *U*. The set *U* corresponds to all the possible system states, *P* – normal states and *S* – the current state which changes in time.
2. Generate a set of candidate detectors $Ca = \{ca_1, ca_2, \ldots, ca_n\}$.
3. Compare each candidate ca_i to the set of known good elements *P*.
4. If a match occurs, discard the individual ca_i, otherwise store it in the mature detector set *D*. Or, to maximize the nonself space coverage with minimum number of detectors, move the matched candidate away from the closest element p_j, then store it in *D*.
5. Monitor *S* for changes by continually matching it against the detectors in *D*. If any detector matches, the change which has occurred most likely is dangerous, as *D* is designed not to match any normal system state.

This algorithm produces a set of detectors capable to recognize non-self patterns. The action following the recognition varies according to the problem under consideration. In the case of transport safety control system it could be an alarm or issue of an immediate stop signal depending on the detected situation.

The detectors and the caught dangerous conditions are stored in immune memory for further processing and to provide further information about the consequences of the attack and possible future actions instead of simply reporting the incidents.

The clonal selection algorithm [author's publication 6]:

1. Generate a population *P* of candidate solutions.
2. Determine the *n* most stimulated individuals.
3. Clone the selected cells. The number of clones is an increasing function of the stimulation level.
4. Submit the clones to a hypermutation scheme.
5. Reselect the improved clones to the population *P*.
6. Perform the suppression (remove from *P* the less stimulated elements).
7. Add a number of newly generated candidates (diversity introduction).
8. Repeat from step 2 until the terminating condition is satisfied.

While usually cloning is proportional and hypermutation is inversely proportional to each candidate's affinity, there's also a version of this algorithm which is a bit closer to a genetic algorithm [21] and where hypermutation happens on an independently random rate.

Several doctoral theses in the relevant fields were reviewed. They state that:

- Both Latvian and Russian railways suffer from having outdated fleet which is in operation since 1960–70s and was designed to be cheap in production but with higher maintenance costs;
- The new vibration and other sensors can be installed instead of the standard temperature sensors;
- Opetion of a diesel locomotive is severely impacted by its gas-air duct's status;
- To calculate a diesel locomotive'soperation process parameter values, one can use Grinevecki-Mazing, Vibe methods and the small movement method;
- There are such diagnostics systems on the market as DIANA (Germany), TORNAD (France), ACES and BHP Iron ORE OCCT (Australia);
- There are expert systems which are used in technical diagnostics, e.g., DELTA (General Electric locomotives) and SOPHIE (diagnostics of circuits and training).
- To diagnose an electrical train's most limiting elements, one can use such signs as:
  - Failure rate;
  - Amount of work to repair or change;
  - Importance weight coefficient.
- Starting time of an electrical drive is a d iagnostical parameter too, and it can help uncover such defects as rotor imbalance.

Diagnostical objects' (DO) status types:

- Not damaged—the system complies to all requirements stated in the normative documents and all its parameters are in the defined boundaries.
- Damaged—one or more of the system's parameters are outside the boundaries. The system still can be operable.
- Operable—the system's base parameters fit in the norm and it correctly solves its tasks. The system can be damaged.

- Non-operable—one or more of the system's base parameters are outside the normal boundaries. The loss of operable status is called a failure.

The full set of DO possible statuses:

$$W = A \cup B \cup C \qquad (2.3)$$

where A is a set of not damaged system statuses; B—damaged but operable; C—non-operable.

Operable and damaged system status sets (Fig. 2.2.):

$$W_1 = A \cup B,\ W_2 = B \cup C \qquad (2.4).$$

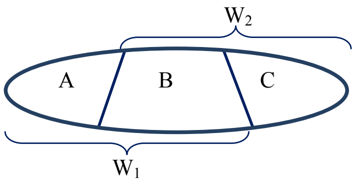

Fig. 2.2. DO status diagram

In the railway automatics, telemechanics and communications systems the basic parameters are voltage, current, frequence, circuit resistance, power etc. The most important task of designing these systems is creating controllable systems with an increased safety level. That means introducing a lot of statuses $Si \in B$, while doing it in such way that in all the cases of the most possible element failures the system as a whole would not transfer from A to C but would stay in B. Then the system continues to fulfill its function and allows to fix the damage.

3. EMBEDDED SYSTEMS AND DEVELOPMENT OF ALGORITHMS

To stop the locomotive one must connect to its electro-pneumatic valve (EPV) through a relay in the device. Disconnecting the circuit would cause the EPV to start its emergency stopping sequence—a loud signal for the first seven seconds and then the brakes are engaged. The device is powered from locomotive's internal 12V source (batteries). The connection schemes are available in appendices 3.4.2. and 3.4.3.

Reading signals from the level crossing and traffic light happens by connecting to the relevant relays in the relay box and checking the voltage read on the input pins.The device is powered from the 220V AC source available in the relay box through the power adapter. The connection schemes are available in appendices 1.3. and 3.4.4.

Both devices require 5–12V / 2A DC power.

To prevent collisions with road vehicles, embedded devices must be installed on them too, or as a more universal approach, the tracks and crossings should be augmented with additional sensors.

The delay of the locomotive embedded device since detecting the dangerous situation is 0 s, but the first seven seconds before the actual braking occurs are spent on the loud signal from the EPV.

The installation costs consist of production and installation labor. The usage costs consist of energy and data costs.

A satellite global positioning system (GPS) data processing algorithm for embedded devices was developed. The algorithm consists of the following steps:

Initializing. Include NewSoftwareSerial library. Include TinyGPS library. Define the GPS serial port and contacts. Run the serial data exchange with the GPS module. Run the serial data exchange with a computer through a USB port.

Step 1. Wait for 2 seconds and output string „Data";

Step 2. If there's new usable data from the GPS, switch to step 3, otherwise go to the step 4;

Step 3. Output string „OK";

Step 4. Run GPSdump subroutine to decode the GPS data and move on to the first step.

In the thesis there are described two additional subroutines—GPSdump for data decoding and feedGPS for getting new GPS data.

When connecting the controller to the computer it is possible to maintain a database and run the appropriate algorithm for data recording. This requires the MySQL database server and the PHP language interpreter. According to the data definition the string from the controller must have 7 information fields each separated by a comma, and a marker of data reliability—„OK" or „Bad". The data is recorded to the database only if it is reliable and contains all the required pieces of information.

A simpler variation of an immune algorithm was tested ona controller which reacted to changes in its environment. The prototype reacts to light and temperature changes.

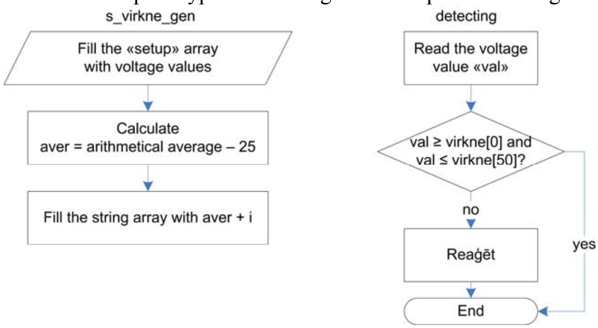

Fig. 3.1. Flowchart of two functions of the immune algorithm

There is aso a proposed algorithm for a neural network [20] which can be used as one of the tools for the immune system to improve the situation on a level crossing.

As in this case, it is not known if an estimated output of the network is correct, the trained neural networks cannot use backpropagation algorithm [32].

A random sequential delta rule self-learning algorithm with the objective function is developed for the neural network

Optimization is defined as a function with two criteria: the train and bus collision probability P and the minimization objective of the train speed to $\Sigma\Delta vi$—the minimization objective.

The first criterion relates to security. Situation is considered to be dangerous if the bus will be crossing at the same time when there is a train. If there are multiple trains and buses, the maximum value is selected from each of the i-th train and each j-th bus collision probability matrix. The second criterion relates to the specific organization of train schedules. Punctuality impacts the operation of the railway positively and train delays cause obstructions to other trains. Consequently, the objective is to minimize the train speed change.

Drawing from these criteria a fall-back optimization objective function F is such:

$$F(\Delta v) = \begin{cases} P = \max(P_{IJ}) \to \min \\ \sum \Delta v_i \to \min \end{cases} \quad (3.1)$$

where

Δv — vehicle speed changes, ie solution

P — the maximum collision probability

P_{IJ} — each i-train collision probability of the j-train,

Δv_i — i-train speed changes.

The developed algorithm consists of the following steps:

Initialization:

- learning set in the index e = 1,
- selected neuron correction sn = 1,
- selected weight correction sw = 1
- retrain = false

Step 1. Select element $e = \{d_1^e\ v_1^e,\ d_2^e,\ v_2^e\}$ from the training set

Step 2. Read x_{min} and x_{max} parameters that limit the output network

Step 4. Direct distribution to calculate the output unit totalizer values:

$$\Sigma_j = \left(\sum_{i=1}^{2n} x_i \cdot w_{ij} \right) + b_j \quad j = \overline{1..n} \quad (3.2)$$

Step 4. Generate the output layer neuron output value of the positives and negatives of saturated linear activation functions:

$$\Delta v_j = \begin{cases} x_{\min}, & \Sigma_j \le x_{\min} \\ \Sigma_j, & x_{\min} < \Sigma_j < x_{\max} \\ x_{\max}, & \Sigma_j \ge x_{\max} \end{cases} \quad (3.3).$$

Step 5. Save the value, if P^{iep} or $\Sigma\Delta v_i^{iep}$ have those.

Step 6. Rate the found solution with the objective function 3.1. $[P, \Sigma\Delta v_i] = F(\Delta v)$.

Step 7. If $P > P^{lim}$ or $\Sigma\Delta v_i > \Sigma\Delta v_i^{\ lim}$, then move on to the step 8.

Step 8. If the last element of the learning set e ≠ e emax is not reached,
then e = e +1 and go to the step 1.
else if the neural network is not to be retrained, then END,
else e = 1 and go to the first step.

Step 9. Sequentially adjust weights.

If ($s_n \neq 1$ and $s_w \neq 1$) or ($P^{iep} < P$ and $\Sigma\Delta v_i^{iep} < \Sigma\Delta v_i$), i.e. if this is not the first element and the situation got worse than before, then return to the previous weights $w_{sw,\,sn} = w_{sw,\,sn} - k$, and choose a different weight;

if $s_w > 2n$, then $s_n = s_n + 1$, otherwise $s_w = s_w + 1$, if $s_n > n$, then $s_n = 1$, $s_w = 1$

Step 10. Generate the correction value as a random number:

k = random (–1000, 1000) / 10,000.

Step 11. Perform weight adjustment: $w_{sw,\,sn} = w_{sn,\,sn} - k$

Step 12. If a weight correction has been made, then the whole neural network should be retrained. retrain = true

Step 13. Go to the step 3.

The modified algorithm is proposed for the train braking before the level crossing with the stationary vehicle on it. It is necessary to stop the train in an emergency, when another improvement is not possible.

4. LABORATORY EXPERIMENT WITH IMMUNE ALGORITHMS

The data set was collected from two controllers which were connected to the vehicle and to the level crossing. The diagram is on Fig. 4.1. [author's publication 11].

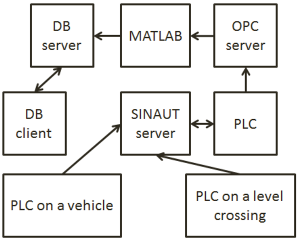

Fig.4.1. Data collection diagram.

Communication between controllers is performed using GPRS modules and a server which is working on the computer and records data rows into the database. This data serves as the basis data set for the algorithms' test [author's publication 13].

The real-valued negative selection (RNS) detector generation starts with a population of candidate detectors, which are then matured. It consists of following steps:

– In the problem space U which contains all possible parameter value combinations define the current situation S which will be continuously monitored. For the initial training purposes fill S with normal or secure situation parameter sets. $S = \{S_1, S_2, \ldots, S_n\}$, $S_i = \{p_1, p_2, \ldots, p_m\}$
– Generate a set of detectors F each of which does not match any element of S. Similarly to a biological immune system one could simply discard every candidate which matches an S element but a more optimal approach is to maximize the field coverage with a minimum number of detectors $F = \{F_1, F_2, \ldots, F_k\}$
– Continually check S for safety-threatening changes by comparing it to the set of detectors F. If any of the detectors is triggered it means that there is a change which is picked up by detectors which are specifically designed in a way that does not respond to safe situations.

This algorithm for the detection of candidates populations evolve within an iterative process [3]. The variables for each detector are its center and radius, which determines the size of the detector m-dimensional field — these values are selected randomly at first.

On every iteration each detector is examined whether it differs from the known "self" elements or other detectors. If this is the case, then the candidate is moved away from the matched or overlapped elements by changing the center coordinates. Then detectors are ranked according to their size (radius) — those which are larger and overlap with other detectors less are considered to be the best and selected for the next generation. The worst are replaced with clones of the best. Clones are moved from by a fixed distance, so they do not overlap with the original detector. Then a set of new random detectors is introduced in order to increase the nonself space coverage and maintain the diversity.

The whole detector generation process terminates when a set of mature (minimum overlapping) detectors are evolved and can provide significant coverage of the nonself space.

A detector is defined as $d = (c, r_d)$, where $c = (c_1, c_2, \ldots, c_m)$ is an m-dimensional point that corresponds to the center of a hypersphere with r_d as its radius. The following parameters are used (Fig. 1.7):

- r_s: threshold variation of a self point;

- α: variable movement of a detector away from a self sample or existing detectors;
- ξ: maximum allowable overlap among the detectors, allowing some overlap can reduce holes in the nonself coverage.

Movement of detector is described as following:

$$c^{jauns} = c + \alpha \frac{c - c^{ruv}}{\left\| c - c^{ruv} \right\|} , \tag{4.1}$$

where c^{tuv} is the closest canditate and || || is the norm of m-dimensional vector.

Movement of clones:

$$c^{clon} = c + r \frac{c - c^{tuv}}{\left\| c - c^{tuv} \right\|} . \tag{4.2}$$

Detectors overlap:

$$W(d) = \sum_{d \neq d'} w(d, d') , \tag{4.3}$$

where w(d, d') is the overlapping measure of two detectors $d = (c, r_d)$ and $d' = (c', r_d')$:

$$w(d, d') = \left(\exp(\delta) - 1\right)^m , \tag{4.4}$$

where m is dimension of the observed area or number of parameters and δ is:

$$\delta = \left(\frac{r_d + r_d' - D}{2 r_d} \right) , \tag{4.5}$$

where D is a distance between c and c'. δ value is within the bounds of 0 and 1.

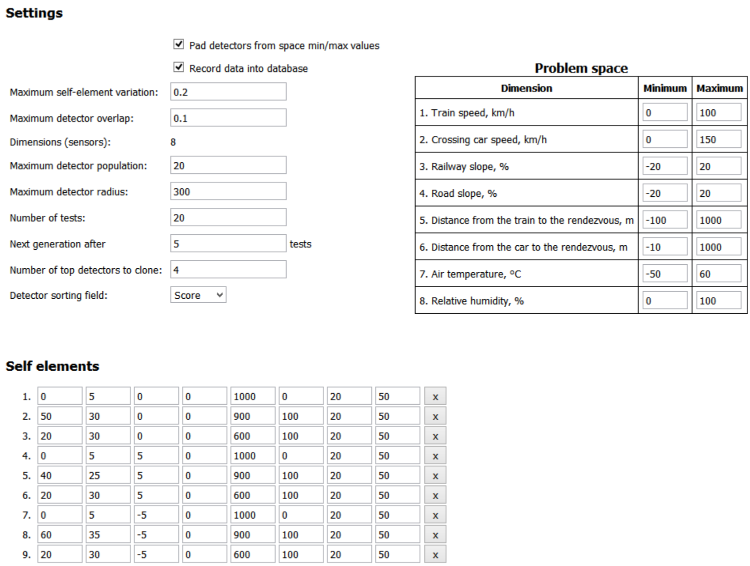

Fig. 4.2. Screenshot from the RNS algorithm PC implementation showing the initial settings for training the detector set.

Tests

#	Antigen	Result	Generation #	Detector #
1	[98.72, 26.37, 7.92, -13.67, 508.15, 512.51, -36.32, 47.57]	Alarm!	1	17
2	[46.94, 3.47, -2.07, -0.7, 153.35, 270.08, 49.57, 7.8]	Alarm!	1	11
3	[84, 128.61, -8.83, 6.11, 691.17, 277.47, -9.93, 98.54]	Alarm!	1	15
4	[93.67, 43.21, 11.44, -1.8, 57.42, 234.12, -49.12, 67.16]	Alarm!	1	11
5	[30.57, 5.21, -15.37, -19.56, 353.4, 710.79, -2.5, 2.27]	Alarm!	1	14
6	[10.87, 10.65, -19.86, 7.32, 304.03, 260.37, 1.09, 96.94]	Alarm!	1	11
7	[5.11, 77.52, 1.9, -7.67, 25.99, 537.38, -33.56, 16.44]	Alarm!	2	14
8	[63.64, 59.49, 8.22, 2.81, 181.57, 564.33, -3.25, 93.42]	Alarm!	2	14
9	[42.31, 114.83, -4.73, -0.78, 701.87, 804.13, 47.73, 48.11]	Alarm!	2	15
10	[52.22, 13.48, -3.01, 17.42, 788.05, 748.83, -20.12, 53.37]	Alarm!	2	15
11	[28.85, 4.42, 3.1, 14.4, 383.7, 897.68, 9.28, 49.83]	Alarm!	2	2
12	[4.44, 122.8, -9.2, -1.42, 396.72, 845.59, -12.04, 58.54]	OK	3	0

Fig. 4.3. Test runs with a sample of antigens on each detector generation with detection results.

During the straightforward detection process the matured detectors are continually compared to new test data samples [author's publications 3, 4]. The distance D between a sample pattern $p = (c_p, r_s)$ and a detector $d = (c_d, r_d)$ is computed in the same way as in the detector generation phase. If $D < (r_s + r_d)$ then the detector d gets activated indicating possible fault.

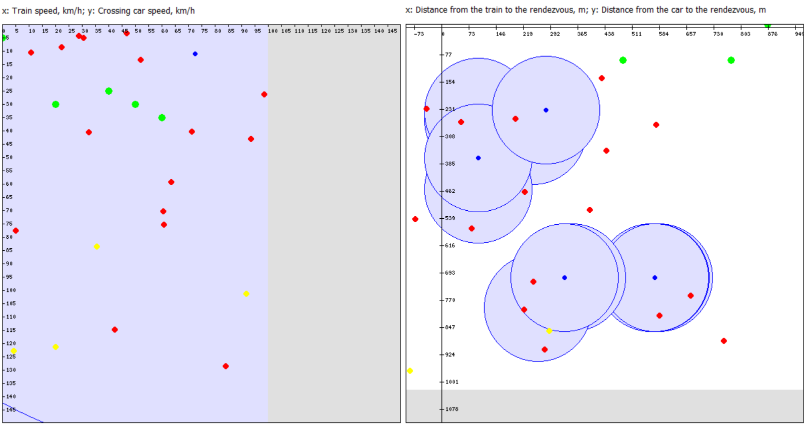

Fig. 4.4. Feature space U covered by the detectors, shown in two-dimensional projections of different parameter pairs. Green points are safe situations, blue circles—detectors, red points—detected emergency situations, yellow points—non-hazardous situations.

In 8-dimensional field the algorithm showed good field coverage with detectors and sound non-self antigen recognition. [author's publication 5]. Fig. 4.4. shows known safe "self" elements (green points) antigens (red points), non-hazardous antigens (yellow points) and detectors (blue circles with center points) projected on two different parameter pairs. The algorithm tuns quite fast, the time increases with an increase in the number of detectors, tests or parameters; in every case the most time is taken by the process of data logging after the main algorithm execution. The selected control activities did not differ as it is meant to be done by the second part of the system. The algorithm can be adapted fot multiprocessor systems which would significantly improve its speed.

The section also describes the experiment with the immune clonal selection algorithm and the genetic algorithm. The goal of the experiment was to test whether an immune optimization algorithm is able to improve the situation and prevent train collisions with road vehicles better than a genetic algorithm, and how quickly the immune algorithm can provide a solution for improving the situation.

To compare the immune algorithm with the genetic algorithm [author's publication 11, 4] a systemwas modeled which consists of a crossing, a train and a bus.

During the experiment some parameters of vehicles were changed: distance from the crossing, train average speed, distance to the bus crossing, average bus speed.

The algorithms used the optimization objective function 3.1.

To test the effectiveness of the immune and genetic algorithms with different parameters they were chosen so that they are similar for both algorithms [32].

Two groups of 10 experiments were planned. Each experiment had different initial model parameters although they are the same across each i^{th} experiment. In the first experimental group the crossing system can only slow down the vehicle to remedy the situation by reducing its average speed by up to 10 km/h. Consequently, the maximum speed change is zero and the minimum speed change is –10. The experiments in the second group could increase the speed by up to 5 km/h or reduce speed by up to 15 km/h.

Since evolutionary algorithms may give different solutions to the same parameters, each experiment was performed in six attempts. To gather the statistical data a total of 240 experiments were performed.

During the statistical analysis of the experimental results the following hypotheses were declared [5]:

H01: The immune algorithm can prevent the dangerous situation on the level crossing.

H02: The genetic algorithm can prevent the dangerous situation on the level crossing.

H03: The neural network can prevent the dangerous situation in the level crossing.

The summary contains the comparison table while full tables are found in the thesis.

In order to statistically test the hypothesis of immune and genetic algorithms' ability to improve the situation at the levelcrossingthe author used a statistical z-test evaluation [4, 32]. A confidence level is defined for the test interval. The value for verification was obtained from the initial situation assessment. These tables contain only the collision probability as a key parameter for accident prevention. In all experiments the immune algorithm found a solution (Table 4.1.). The results show that the hypotheses cannot be rejected with a probability of 0.999 (Tables 4.1. and 4.2.). This means that the immune algorithm can help avoid the dangerous situation on the level crossing. The genetic algorithm also finds good solutions which are better than the original state.

Table 4.1.

Fragment of immune algorithm hypothesis test results by a collision probability criterion

Exp. No.	Confidence leve	Test interval	Tested value	Result
1	99.90%	(0.0017; 0.00166)	0.62601	H01 cannot be denied with prob. 99.90%
2	99.90%	(0.0017; 0.0017)	0.5768	H01 cannot be denied with prob. 99.90%

Table 4.2.

Fragment of genetic algorithm hypothesis test results by a collision probability criterion

Exp. No.	Confidence level	Test interval	Tested value	Result
1	99.90%	(-0.0293; 0.26966)	0.62601	H02 cannot be denied with prob. 99.90%
2	99.90%	(0.0025; 0.02232)	0.5768	H02 cannot be denied with prob. 99.90%

Experiment with a neural network [20] used the target function defined in (3.1), but unlike evolutionary algorithms it was taught to decide on the change of speed to prevent a collision between trains and trucks. If the total number of trains and vehicles is *n*, then the neural network has 2*n* input pairs. Since this number can vary depending on the number of vehicles, the neural network input and the output neuron count is dynamic.

Each neural network input pair consists of two parameters: the vehicle's distance to the object and the crossing object average speed v_i.

According to the object count the neural network has n outputs which deliver the change in object's speed necessary to avoid a collision. Output layer neural inputs have weights W_{ij}, which determine a multiplier for each i^{th} input layer neuron output for each j^{th} output layer neuron and a bias b_j. Each output layer j^{th} neuron generates j^{th} vehicle speed change Δv_j. To decide on the need for training the neural network, the calculated Δv_1 ... Δv_n values are transferred to the self-learning multi-objective function F which measures the output efficiency (the collision probability) P and total train speed changes $\Sigma\Delta v_i$. The structure of the neural network is shown in Fig. 4.5.

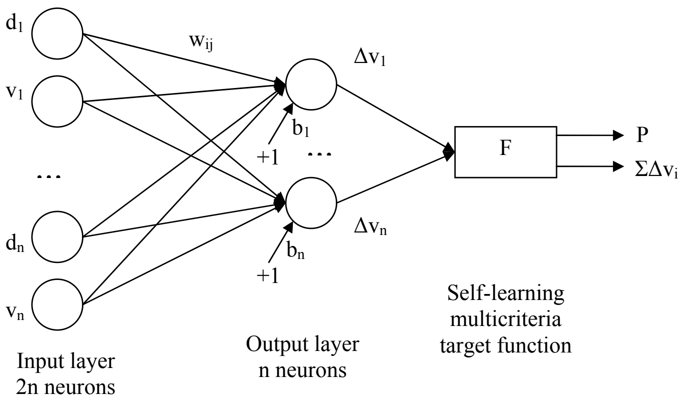

Fig. 4.5. Structure of the neural network

During the training process each element of the training set is put into the neural network input layer. The output — bus and train speed changes Δv_1 and Δv_2 — is measured by the objective function.

A trained neron is such that each of the speed change elements reduces the collision probability by at least 0.005 and the average change of train speed is less than 3 km/h.

The hypothesis H03 that neural networks can prevent dangerous situations at the level crossing, was statistically tested (Table 4.3).

Table 4.3.

Fragment of neural network hypothesis z-test results

Exp. No.	Confidence level	Test interval	Tested value	Result
1	99.90%	(0.0001; 0.0007)	0.62601	H03 cannot be denied with prob. 99.90%
2	99.90%	(0; 0.00048)	0.5768	H03 cannot be denied with prob. 99.90%

According to the analysis of experimental results it can be concluded that all the tested algorithms can improve the situation at the level crossing.

Immune algorithm performance is compared with the genetic algorithm and the neural network by following criteria: the average value of the collision probability, machine time for an algorithm and stability the results.

Table 4.4.

Immune algorithm criteria mean value comparison with genetic algorithm results

Criterion	Mean of Immune Algorithm (IA)	Mean of Genetic Algorithm (GA)	IA in comparison with the GA
Target function	0.00119	0.00962	708.40%
Collision probability	0.00086	0.00327	280.23%
Train speed change	0.04167	0.7	1579.87%
Machine time	0.35433	0.34762	-1.89%

Table 4.5.

Immune algorithm criteria mean value comparison with neural network results

Criterion	Mean of Immune Algorithm (IA)	Mean of Neural Network (NN)	IA in comparison with the NN
Target function	0.00119	0.00263	121.01%
Collision probability	0.00086	0.00088	2.33%
Train speed change	0.04167	0.15028	260.64%
Machine time	0.35433	0.15961	-54.95%

Table 4.6.

Immune algorithm stability comparison with the genetic algorithm

Criterion	Standard deviation of Immune Algorithm (IA)	Standard deviation of Genetic Algorithm (GA)	IA in comparison with the GA
Target function	0.00011	0.00088	700.00%
Collision probability	8.00E-05	0.0003	275.00%
Train speed change	0.00382	0.06417	1579.84%
Machine time	0.01604	0.01495	-6.80%

Table 4.7.

Immune algorithm stability comparison with the neural network

Criterion	Standard deviation of Immune Algorithm (IA)	Standard deviation of Neural Network (NN)	IA in comparison with the NN
Target function	0.00011	0.00024	118.18%
Collision probability	8.00E-05	8.00E-05	0.00%
Train speed change	0.00382	0.01378	260.73%
Machine time	0.01604	0.00634	-60.47%

Comparing the average values from tables 4.4. and 4.5, it can be seen that the best result by the collision probability criterion and the minimum train speed change belongs to the immune algorithm. The neural network result for collision avoidance was 2.33% worse while running 54.95% faster. Analysis of the algorithm results stability from the tables 4.6. and 4.7 shows that the immune algorithm finds solutions with the smallest standard deviation and the results are stable. Neural network shows similar stability by the collision probability criterion. By the measure of machine time consumed the immune algorithm loses by 6.80% to the GA and by 60.47% to the NN.

5. EXPERIMENT WITH EMBEDDED DEVICES AND IMMUNE SYSTEM

The aim of experiments with the embedded devices in the real world was to check and prove that the thesis developed mathematical models, methods and algorithms are capable of performing and technically applicable to real objects.

For this purpose, three level-crossings in Riga, as well as station Bolderāja and Lāčupe–Bolderāja railway section has been selected for tests in collaboration with JSC "Latvian Railway".

During experiments three types of microcontrollers were tested — Siemens Simatic S7-200 PLC, Arduino Uno/Mega and Waspmote; two communication options were tested — mobile communications GSM/GPRS and a separate radio channel, and the ability of the algorithm to remedy the situation at level crossings and equipment performance and function in the real world.

It is proposed to augment the existing real railway system with the following elements: satellite navigation receivers (GPS, Galileo, EGNOS, LATPOS) S_p for level-crossings, S_L for locomotives and S_A for road vehicles; wireless signal (GSM or radio channel) transmission antennas A_p for level crossings, A_L for locomotives, A_A for road vehicles and A_D for control center; the device for object detection on level-crossing K_p, braking control system on the locomotive K_L and the device of road vehicle K_A, device of the dispatching center K_D and the database DB of area control center.

Fig. 5.1. shows the apparatus for the data structure. Next to the name in brackets there is an example of the device's unique identification code. Bold solid lines show continuous permanent data exchange process, dashed lines show the single data exchange process:

1. Dispatching center is a common database DB for local railways with each level crossing or station located in its area. Zones do not intersect with each other.

2. KL-board equipment (10) continuously sends locomotives position of the latitude X_1 and longitude Y_1 to the dispatching center device K_D (5)

3. Dispatching center device K_D continuously compares the coordinates of the train with a database DB and when the train is in the area defined in database, then DB returns the zone ID number (for example, ID number 1000), and K_D transfers this code to the train unit K_L.

4. While the train device K_L (10) is continuously receiving the area code from the central controller (eg, ID number 1000) it starts connecting to the level crossing device K_P (1000) and registers there.

5. When a new train is registered in the area of the station or level-crossing K_P sends to the train device (for example, ID 10) the stations stations plan in a form of a graph, which describes the station node points (track circuits, points, points, traffic lights, level crossings), its geographical coordinates and lines between those points.

6. Station device K_p continuously sends node controlled conditions (crossings, traffic lights and switches) to the trains.

7. K_L-board device, taking into account the motion parameters, calculates the moment of time for the safe stop and sends a signal the the station or level-crossing device.

8. When the car is located in the crossing area and is stopped, the K_A sends its coordinates X_2 and Y_2 to the level-crossing device K_p

9. K_P device checks the location of the car, and if it is on a level crossing, generates a "busy" signal the train device K_L

10. KL-board equipment, when receiving "busy"signal starts braking process of the rolling stock using the braking control algorithm.

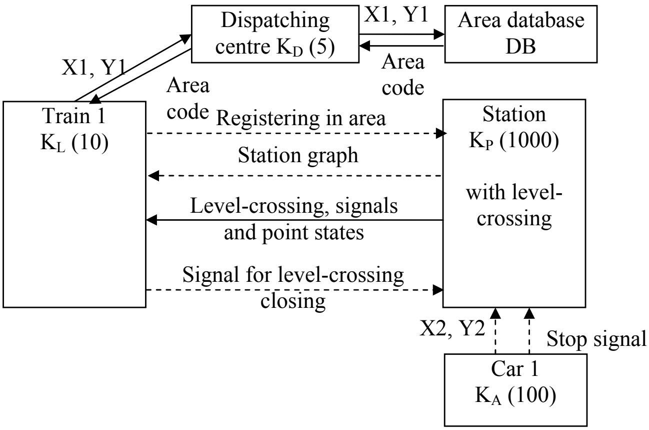

Fig. 5.1. Data exchange diagram

An experiment in recognition of dangerous situations was carried out with the detector generation phase with SIMATIC controllers. [author's publication 3]. Antigen recognition results were displayed on the screen or the LED and an appropriate for the activated detector control signal was sent to the output.

To test the system which can learn independently using the objective function, a prototype system was designed for athe experiment; the prototype is based on Waspmote microcontroller board and uses operator-independent radio channel.

The system is able to assess the situation and offer traffic participants — the train driver or the vehicle driver — to change the speed in order to avoid dangerous situations.

Experiment setup: a car which is equipped with the developed device is approaching the crossing and enters into an earshot of the radio channel, begins sending its speed and distance to the crossing, the train which is also entering the area, begins sending its speed and distance to the crossing; the crossing facility, which receives data from vehicles, enables the immune system based on neural networks which learn using the objective function, and finds the amount to reduce the speed of the train so that collision probability would be minimal.

The detection distance to the vehicle is 500 meters, and train detection distance is 1000 m. Speed of the train is 80 km/h and the car's speed is 40 km/h. After evaluation, the collision probability is defined as 0.469. Neural network training completed in 165 milliseconds time and gives an answer to remedy the situation up to a probability of 0.00014.

The embedded device of the car captures a report on the need to reduce the speed from the level crossing equipment with a self-training option.

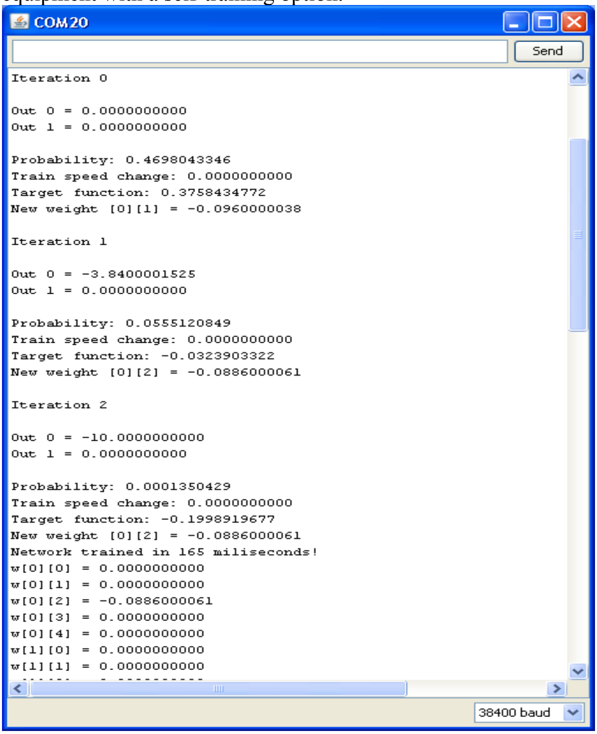

Fig. 5.2. System self-learning

The new railroad crossing wireless security system consists of 4 components: the locomotive device, the car device, the level crossing device and the command center. Proposed benefits of the system are such: it works in parallel to the existing ALS and does not interfere with the existing operation of the system, increasing the safety of trains; the equipment uses wireless communication networks and can work well in track segments with semi-automatic locking system.

The primary function of the locomotive device is to stop the train before crossing when crossing the road stands.

The main function of the crossing facilities is to check whether anything is blocking the crossing and warn the train approaching the crossing.

The vehicle device features are: determination of the vehicle location, direction of motion, speed using GPS and transfer the information to the command center using a wireless channel such as GSM or other radio; if an ID of the nearest crossing device is obtained, transfer the data to an appropriate level crossing device through the wireless communications (GSM or radio) channel, sending a warning to the crossing device through wireless communication (GSM or radio) channel when the vehicle is in the crossing area and its speed is around 0 km/h.

The command center device features are: receive locomotives' location, direction of movement, speed using wireless GSM channels, receive cars' location, direction of movement, speed using wireless GSM channel; find nearest crossings on the way of locomotive and transmit its identifier to the locomotive device using wireless communication channel, find the nearest level crossing to the car and transmit it to the device using a wireless channel, save the received data to the database.

Fig. 5.3. Devices of the locomotive, level-crossing and car.

Level crossing safety equipment had several trials in collaboration with JSC "Latvian Railway":

August 30, 2011 experiment at Jāņavārti–Zemitāni 2.37 km Vietalvas street two-way level-crossing of the railway section, equipped with automatic traffic lights, alarm systems and automatic barriers. The experiment showed that the level crossing equipment locomotive identification feature works well, but signal transfer time delays from 1 to 15 seconds were detected, which should be taken into account in the further calculations and proximity to the crossing zone definition.

November 24, 2011 at Lāčupe–Iļģuciems 0.94 km level-crossing (Dzirciema street). The experiment showed that the level crossing equipment is running smoothly in a stable manner and fulfills all the functions described in the protocol.

December 2, 2011 tests showed that the crossing facility is successfully cooperating with the car device, and the vehicle identification is performing and it can detect a standing car on a level crossing.

On March 23, 2012 experiments were made with the locomotive and level crossing devices. The locomotive device was installed on the M62 locomotive and the other in Bolderaja relay box. During the experiment communication between the controllers and computers was established through the open IRC protocol using string format for variables; the data was obtained with PHP software. It was concluded that continuous output of all data when it is not the most important, is sub-optimal and there is a need to develop a position and velocity prediction algorithms in order to continue the calculation, if the data is received late.

On February 27-28, 2013 the new rail safety equipment prototypes “SAFE-R 8” and “SAFE-R 9” with their software were tested in real operating conditions at Bolderāja station and Lāčupe-Bolderāja railway section. SAFE-R 8 prototypes were installed in the locomotive and railcar, and SAFE-R 9 prototype in the relay box of the station.

The following equipment abilities were tested: receiving GPS signal and determining the position and exact track of the locomotive and railcar; SAFE-R 8 receiving control signals from SAFE-R 9 via the radio channel; correct data display on SAFE-R 8 screen which includes speed, braking distance, braking and emergency information mode indicator, traffic lights at the front and rear of the locomotive, distances to these points and their positions; receiving full data about the station on demand;correctly backing up traffic light and track switching signals; receiving a request through radio communication and transmitting the required information without any loss of data.

During experiments, the locomotive leaves the station on a specified segment. Several trips were made moving away and approaching the entrance of the station equipped with a traffic light with prohibitive signal. The signal switches to green, the locomotive enters the station and stops. Then the railcar departs from the station on a specified segment with several trips moving away and approaching the entrance to the station with a prohibitive signal. When the signal switched to an allowing signal the railcar enters the station and stops.

Experimental results showed that the SAFE-R 8 device can work in various movable units: locomotive or railcar without configuration changes, SAFE-R 8 and SAFE-R 9 devices do not use public wireless communications network and are working on a separate radio frequency channel. The SAFE-R system is able to stop the train safely before passing the red signal, SAFE-R 8 device is able to determine the service and emergency braking distance to warn the driver of the need for braking and automatically stop the train if the driver did not respond in timely manner. The devices are compatible with the SCB system and do not interfere with the existing system. SAFE-R 8 displays the corresponding traffic light readings from the SAFE-R 9 device. Train devices can work in the track sections which are equipped with field SAFE-R 9 devices, on other sections they do not apply automatic braking and work in the information display mode. Area is limited by the communication equipment capacity. During the tests the maximum distance was limited by 12 km in ideal conditions. During the experiment, a stable uninterruptable data exchange between train and field SAFE-R 9 devices was recorded at 2600 meters. The maximum claimed distance of 12 km was not verified due to the test area limitations. The field device SAFE-R 9 is able to read busy tracks, switch positions and signals from an existing SCB system and is able to provide data about the station and train speed limits for SAFE-R 8 devices. SAFE-R 8 device can accurately determine the proper traffic lights and signals despite the satellite navigation signal uncertainty.

RESULTS AND CONCLUSIONS OF THE THESIS

- The term „evolutionary algorithm" includes such artificial intelligence systems as artificial neural networks, genetic algorithms and immune algorithms. They are inspired by evolutionary processes of biological systems and work with data as „populations".
- The most important features of immune algorithms are self-learning, diversity maintenance, memory about the past decisions, recognition of previously unknown but similar elements, noise reduction and classification ability.
- The research looks into such disruptions to the electrical railway transport movement as collisions with other objects and ignoring the prohibitive trafic light signal.
- A structure of the intelligent rolling stock safety control system and its algorithm was designed.
- The intelligent railway transport safety control system must be structured in two phases—anomaly recognition (implemented with the immune negative selection algorithm) and selection of the best reaction based on the type of the anomaly (implemented with the immune clonal selection algorithm).
- Examples of immune algorithm implementations for computers and embedded devices were developed.
- Controller modules and sensors were selected for the experiments.
- The data flow between the embedded devices and control center computers was implemented, as well as its logging into the database.
- Experiments in the laboratory and in the field were conducted; the embedded device prototypes were connected to the locomotives, traffic light and level crossing relay boxes.
- Analysis of the experiments' results shows that: it's necessary to limit the data output, leaving only the most important data for each time moment and precisely stating the addressee; imprecise or late data can cause wrong decisions by the algorithm.
- Limited memory and processing power is the reason for controller to implement the immune algorithm in a simpler form.
- Analysis of experimental results showed that the amount of data output should be limited, leaving only the most relevant data, and a more precise addressing system should be used; inaccurate or late data can cause wrong or late decision.
- The developed prototypes of embedded devices can be connected and are able to cooperate with the railway facilities, and the locomotive, traffic lights and crossing relay cabinets that allow them to use the existing railway infrastructure and rolling stock.
- Neural networks can be used as a mechanism in the immune system that can self-train and improve the situation of the transport system by reducing the collision probability.
- The defined multi-objective function can be used in evolutionary algorithms, i.e. neural networks, immune and genetic algorithms.
- The modified braking control algorithm makes it possible to stop the train and prevent a collision where a correction of the situation is not possible, and that provides an additional security feature for the level crossing's AIS.
- Laboratory computer experiments compared three evolutionary computing methods—an immune algorithm, a genetic algorithm and a neural network—that made it possible to assess the immune algorithms by rail and road vehicle collision avoidance criterion, the train speed to benchmark and machine time consumed in the operation of the algorithm;

- The immune negative selection algorithm showed good space coverage with detection and sound “foreign” antigen recognition, which can effectively find a dangerous situation on a level crossing and run the optimization mechanism to remedy the situation;
- The immune algorithm decreases the probability of collisions up to 280% better than the genetic algorithm and 2.88% better than the neural network.
- Immune algorithm kept the train speed up to 1579% closer to the one defined at the beginning than the genetic algorithm and up to 260.64% closer than the neural network.
- Despite the stochastical behavior, the immune algorithm shows an average of 850% more stable solutions compared with the genetic algorithm and 126.30% compared to the neural network.
- Neural network training has the smallest machine time consumed which is 60.47% less than the immune algorithm.
- The genetic algorithm works 1.89% faster than the immune algorithm which can be considered negligible in comparison with the outcome of the immune algorithm found measured by the objective function criteria.
- After the performed experiments with the additional level crossing safety devices it can be concluded that:
    - the additional level crossing safety equipment is in working order;
    - the devices can provide identification of approaching cars and locomotives;
    - the devices can transfer current status of control relays to the locomotives and cars;
    - the devices can provide the location of cars parked in the crossing area and pass the information on to the locomotive using the wireless communication channel;
    - the devices are connected in parallel to an existing railway control system and do not disturb functionality of the existing control system increasing the existing safety level of the railway system.
- During the experiment, it was found that:
    - a continuous output all the data, if not the most important, is not optimal;
    - it is necessary to develop a position and velocity prediction algorithms in order to continue the calculation if the data is received too late;
    - to stabilize GPS data the Kalman filter can be used but its implementation in controllers will take up a lot of memory and computational resources.

INFORMATION SOURCES

1. A. Klūga. Satelītu radionavigācijas sistēmas. RTU, Rīga, 2010.
2. A. Lewiński, T. Perzyński. The Reliability and Safety of Railway Control Systems Based on New Information Technologies. Selected papers of 10th Conference, TST 2010
3. A. Watkins, J. Timmis, L. Boggess. Artificial Immune Recognition System (AIRS): An Immune-Inspired Supervised Learning Algorithm. 2003. Kluver Academic Publishers.
4. Alps I. Promocijas darbs: Intelektuālā elektrotransporta vadības sistēmu sarakstu problēmu modelēšana neparedzētos gadījumos. RTU, Rīga, 2012.
5. Arhipova I., Bāliņa S., „Statistika ekonomikā”, mācību līdzeklis, Datorzinību centrs, Rīga, 2008, 352 lpp.

6. C. Hunga, Y. Suda, M. Aki, T. Tsuji, M. Morikawa, T. Yamashita, T. Kawanabe, T. Kunimi. Study on early signs detection of derailment for railway vehicles. // Proceedings of 21st International Symposium on Dynamics and Vehicles on Roads and Tracks, Stockholm, August 17-21 2009
7. D. Dasgupta, K. Krishna Kumar, D. Wong, M. Berry. Negative Selection Algorithm for Aircraft Fault Detection. In proceedings of ICARIS 2004.
8. D. Eidukas, R. Kalnius. Models Quality of Electronics Products. 2010. No. 2(98)
9. D. Eidukas, R. Kalnius. Models Quality of Mechatronic Products. 2009. No. 7(95)
10. D. Eidukas, R. Kalnius. Two-Parameter Electronic Devices Quality Models. 2010. No. 7(103)
11. D. Eidukas. Modeling of Level of Defects in Electronics Systems. 2010. No. 3(99)
12. D. Eidukas. Stochastic Models of Quality Level of Mechatronic Products. 2008. No. 3(83)
13. D. Eidukas. Stochastic Models Quality of Electronics Systems. 2008. No. 5(85)
14. Das S., Gui M., Pahwa A. Artificial Immune Systems for Self-Nonself Discrimination: Application to Anomaly Detection. Advances of Computational Intelligence in Industrial Systems. Springer, 2008.
15. E. Bareiša, V. Jusas, K. Motiejūnas, R. Šeinauskas. Functional Test Generation Procedures. 2006. No. 8(72)
16. E. Bareiša, V. Jusas, K. Motiejūnas, R. Šeinauskas. Functional Test Transformation to Improve Compaction. 2008. No. 3(83)
17. E. Bareiša, V. Jusas, K. Motiejūnas, R. Šeinauskas. Transition Fault Coverage For Different Implementations Of The Circuit. ELEKTRONIKA IR ELEKTROTECHNIKA. 2005. Nr.3(59)
18. Goldberg. Computer Aided Gas Pipeline Operation Using Genetic Algorithms and Rule Learning. 1983.
19. H. Tsunashima, H. Mori. Fault detection of railway vehicle suspension using multiple model approach. // Proceedings of 21st International Symposium on Dynamics and Vehicles on Roads and Tracks, Stockholm, August 17-21 2009
20. Haykin S. Neural Networks. A Comprehensive Foundation. Second Edition. Prentice Hall, 2006 – 842 p.
21. Holland H. John. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. University of Michigan Press, 1975 - 183 p.
22. I.Beinarts. Promocijas darbs: Izplūdušās loģikas kontrolleru algoritmu izpēte pasažieru transporta salona siltuma komforta vadībai. RTU, Rīga, 2011
23. J.D. Farmer, N.H. Packard, A.S. Perelson. The Immune System, Adaptation, and Machine Learning
24. J.O. Kephart. A biologically inspired immune system for computers. In proceedings of Artificial Life IV: The Fourth International Workshop on the Synthesis and Simulation of Living Systems, 1994. MIT Press.
25. Janota, K. Rástočný, J. Zahradník. Modelling of safety properties of a control system. // Proceedings of Transport Systems Telematics 2006, Katowice, October 2006
26. K. KrishnaKumar, J. Neidhoefer. Immunized Adaptive Critic for an Autonomous Aircraft Control Application. In Artificial Immune Systems and Their Applications. Ed. by D. Dasgupta.Springer, 1999.
27. K. Rástočný, J. Ždánský. Application of programmable logic controllers in safety systems. // Proceedings of Transport Systems Telematics 2007, Katowice, October 2007

28. L. Wolski, P. Zaprawa, A. Komkowski, A. Adolph. A remote diagnosis & monitoring system using internet technology. // Proceedings of Transport Systems Telematics 2005, Katowice, October 2005
29. A. Lewiński, T. Perzyński. The modeling of multicomputer structure in railway control applications. // Proceedings of Transport Systems Telematics 2006, Katowice, 2006
30. M. A. Kamarposhti. Contingencies Ranking for Voltage Stability Analysis using Continuation Power Flow Method. 2010. No. 3(99)
31. M. Bereta, T. Burczynski. Immune K-means and negative selection algorithms for data analysis. Elsevier IS 179, 2009.
32. M. Gorobecs. Promocijas darbs: Ģenētisko algoritmu izpēte elektriskā transporta optimālai vadībai. RTU, Rīga, 2008
33. M. Gorobetz, I. Alps, A. Levchenkov. Mathematical Formulation of Public Electric Transport Scheduling Task for Artificial Immune Systems. Proceedings of ITELMS'2009, Kaunas.
34. M. P. Groover. Automation, Production Systems, and Computer-Integrated Manufacturing. - Prentice Hall, NJ, 2008.
35. Mitsuo Gen, Runwei Cheng. Genetic Algorithms & Engineering Design. Wiley, 1997
36. Mitsuo Gen, Runwei Cheng. Genetic Algorithms & Engineering Optimization. Wiley, 2000
37. O. Krūmiņš. Promocijas darbs: Inteliģento transporta sistēmu izveides principi un realizācijas iespējas. RTU, Rīga, 2010
38. P. Gorczyca, J. Młyńczak. Operation and diagnostic database of devices. // Proceedings of Transport Systems Telematics 2005, Katowice, October 2005
39. P. Musilek, A. Lau, M. Reformat, L. Wyard-Scott. Immune Programming. Elsevier IS 179, 2009.
40. P. Negi. Artificial Immune System Based Urban Traffic Control. Texas A&M University, 2006.
41. R. Tavakkoli-Moghaddam, A. Rahimi-Vahed, A. H. Mirzaei. A hybrid multi-objective immune algorithm for a flow shop scheduling problem with bi-objectives: Weighted mean completion time and weighted mean tardiness. Elsevier IS 179, 2009.
42. S. Forrest, A.S. Perelson, L. Allen, R. Cherukuri. Self-nonself discrimination in a computer. In proceedings of the 1994 IEEE Symposium on Research in Security and Privacy. Los Alamitos, CA.
43. S. T. Powers, J. He. A hybrid artificial immune system and Self Organising Map for network intrusion detection. Elsevier IS 179, 2009.
44. T. A. S. Masutti, L. N. de Castro. A self-organizing neural network using ideas from the immune system to solve the traveling salesman problem. Elsevier IS 179, 2009.
45. T. Fukuda, K. Mori and M. Tsukiyama. Parallel Search forMulti-Modal Function Optimization with Diversity and Learning of Immune Algorithm. In Artificial Immune Systems and Their Applications. Ed. by D. Dasgupta.Springer, 1999.
46. Uteševs I. Promocijas darbs: Elektriskā sliežu transporta iebūvēto bezvadu bremzēšanas iekārtas vadības algoritmu izstrāde izmantojot pozicionēšanas informācijas sistēmas. RTU, Rīga, 2010
47. V. Abraitis, Ž. Tamoševičius. The Transition Fault Model of Programmable Logic. 2008. No. 1(81)
48. V. Knyva, M. Knyva. Algorithm for Estimation of Measurement Devices Software's Metrological Reliability. 2010. No. 4(100)

49. Y. Hasegawa, H. Tsunashima, Y. Marumo, T. Kojima. Detection of unusual braking behaviour of train driver. // Proceedings of 21st International Symposium on Dynamics and Vehicles on Roads and Tracks, Stockholm, August 17-21 2009
50. Мартиросов Д.С. Диагностирование технического состояния ЖРД на основе математических моделей рабочих процессов и измеряемых параметров методом структурного исключения. автореферат дис. кандидата технических наук: Научно-производ. объединение энергетич. машиностроения, Москва, 1998
51. Сапожников В. В., Сапожников Вл. В. Основы технической диагностики. М.: Маршрут, 2004.
52. Титанаков Д.А. Оценка технического состояния газовоздушного тракта тепловозных дизелей. автореферат дис. кандидата технических наук. Ом. гос. ун-т путей сообщ., Омск, 2005
53. Тычков А.С. Диагностирование тяговых электродвигателей грузовых электровозов по параметрам магнитного поля. автореферат дис. кандидата технических наук: Сам. гос. ун-т путей сообщения, Самара, 2009
54. Чернов Д.В. Функциональная диагностика асинхронных электродвигателей в переходных режимах работы. автореферат дис. кандидата технических наук: Ульян. гос. техн. ун-т, Ульяновск, 2005
55. Швалов Д.В., Шаповалов В.В. — Системы диагностики подвижного состава. Маршрут, 2005
56. Юдина О. Диагностика состояния электродвигателя. Proceedings of the International conference on system analysis and information technologies. Kyiv, 2009.