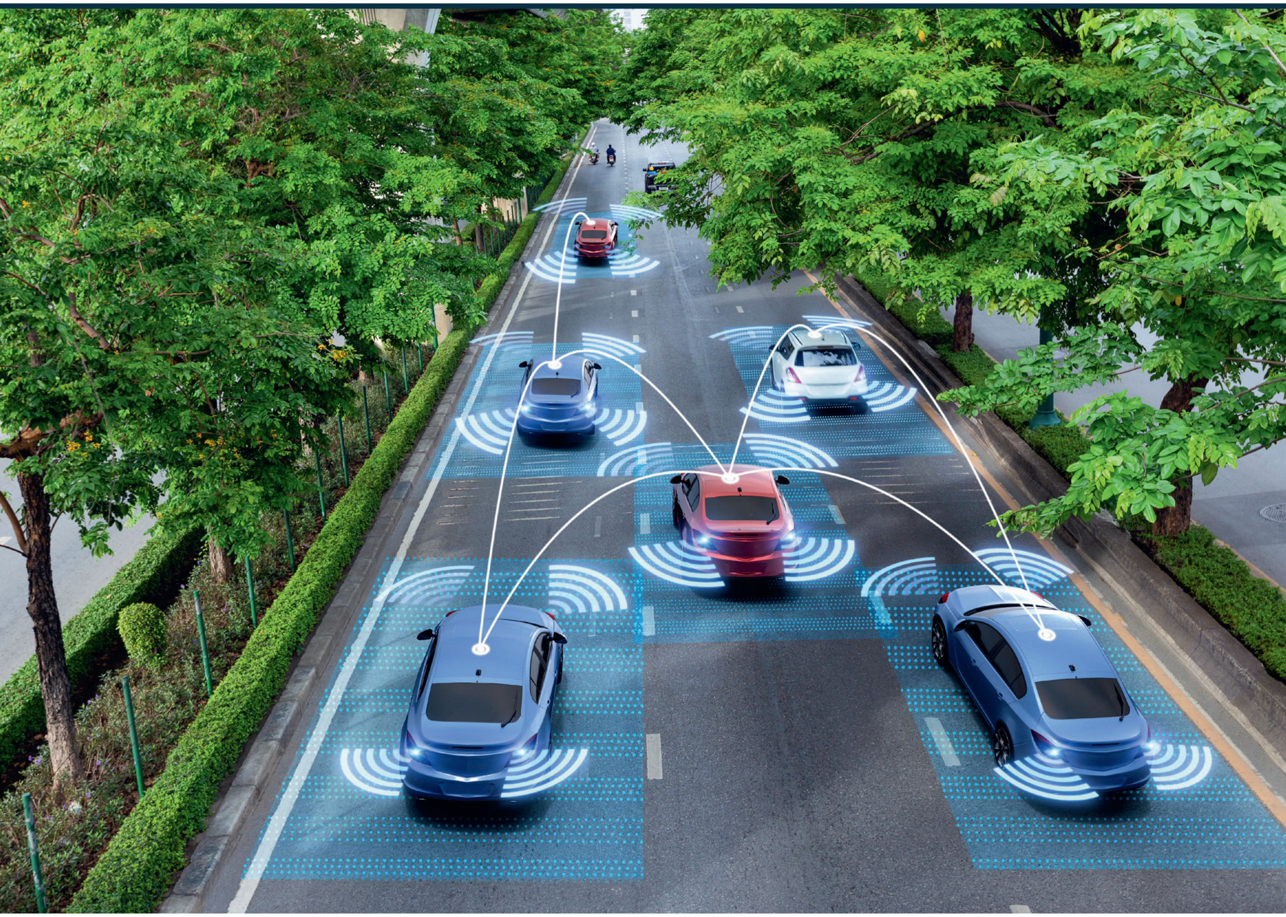
1862

RIGA TECHNICAL UNIVERSITY

Anna Beinaroviča

RESEARCH AND DEVELOPMENT OF IMMUNE NEURAL NETWORK ALGORITHMS FOR ELECTRICAL TRANSPORT DANGEROUS SITUATION RECOGNITION AND PREVENTION

Doctoral Thesis



RTU Press
Riga 2023

RIGA TECHNICAL UNIVERSITY
Faculty of Electrical and Environmental Engineering
Institute of Industrial Electronics and Electrical Engineering

Anna Beinarovica
Doctoral Student of the Study Programme "Computerised Control of Electrical Technologies"

RESEARCH AND DEVELOPMENT OF IMMUNE NEURAL NETWORK ALGORITHMS FOR ELECTRICAL TRANSPORT DANGEROUS SITUATION RECOGNITION AND PREVENTION

Doctoral Thesis

Scientific supervisor
Professor, Dr. sc. ing.
MIKHAIL GOROBETZ

Riga 2023

ACKNOWLEDGEMENTS

I would like to express my gratitude to Professor Mikhail Gorobetz for all the guidance, support, and instruction he provided me. I could not have embarked on this trip without his outstanding knowledge and expertise, which made my academic studies possible.

I thank all my colleagues at the RTU Institute of Industrial Electronics and Electrical Engineering for the friendly and motivating environment, especially Professor Dr. habil. sc.ing Leonīds Ribickis for his time, effort, and understanding in helping me succeed in my studies, Professor Pēteris Apse-Apsītis and Professor Ivars Raņķis, for their insightful comments and encouragement, but also for the questions which incented me to widen my research from various perspectives, lead researcher Andrejs Potapovs for technical and perceptive assistance with the research, research assistant Aija Laicāne for support and advice during doctoral studies.

Also, gratitude is expressed to RTU Doctoral study units and Doctoral School employees, Doctoral study units head Lauris Bisenieks, coordinator of promotion councils Alina Galkina, senior project manager Krista Papēde for support and advice during doctoral studies.

My big gratitude also to other colleagues in the university and institute for their advice, positivism, understanding and encouragement throughout these years.

To conclude, I would like to thank my husband and my parents. It would have been impossible to finish my studies without their unwavering support over the past few years.

NATIONAL DEVELOPMENT PLAN 2020 EUROPEAN UNION European Social Fund
INVESTING IN YOUR FUTURE

1862
RIGA TECHNICAL UNIVERSITY

This work has been supported by the European Social Fund within the Project No 8.2.2.0/20/I/008 «Strengthening of PhD students and academic personnel of Riga Technical University and BA School of Business and Finance in the strategic fields of specialization» of the Specific Objective 8.2.2 «To Strengthen Academic Staff of Higher Education Institutions in Strategic Specialization Areas» of the Operational Programme «Growth and Employment».
This research was supported by Riga Technical University's Doctoral Grant programme.

ABSTRACT

The number of used vehicles is growing very fast. This causes the bigger number of accidents. A lot of accidents and crashes are caused by driver factor. The solution might be an unmanned transport control. But use of such transport leads to another problem – how to provide safety drive and to prevent collisions of unmanned transport.

The scietific novelty of the doctoral research is the technology for the embedded system, which is able to learn in real time to avoid crashes without any datasets, preliminary training and a teacher. This technology is universal for different transport types.

Author named it as Immune Neural Network (INN). It is a symbiosis that uses best features of the artificial neural networks (ANN) and artificial immune system (AIS). Separately each of them has a bunch of disadvantages. ANN requires preliminary training, may be overtrained etc. AIS does not require training, but the optimization requires time and may not be used for real time control. Novel INN system combines the possibilities of ANN and AIS is adapted for real time safety control.

The main goal of the doctoral work is to develop immune neural network based technology of machine learning for unsupervised safe vehicle control.

The main hypothesis is that an immune neural network can make control decisions to prevent vehicle collisions with better performance than a traditional neural network in this task.

Following tasks were defined:

- To study the objects of electric transport traffic movement control and their interaction.
- To study the existing solutions for the recognition and prevention of dangerous situations in electric transport, which are based on the algorithms of ANN.
- To compare centralized, decentralized and distributed system structures, to choose the most suitable one for the proposed task and to develop the novel system structure, which could help to make the proposed system cheaper, faster and easier to implement.
- To develop mathematical models and algorithms, that could help to solve different types of transport safety and collision prevention tasks, such as object recognition task, traffic light signal recognition task, possible crossing point detection task, collision probability evaluation task, collision prevention task.
- To develop novel INN based algorithm for UEV dangerous situation recognition and prevention task.
- To develop electrical circuit diagram with an immune memory (IM) for UEV based on a single board computer.
- To make computer simulations and to prove the efficiency of the proposed algorithms.

The introduction of the dissertation contains the task statement, the analysis of the existing types of solutions of the problem, grounding and topicality of the chosen topic, practical application of the proposed algorithms and methods.

First chapter of the doctoral thesis is devoted to the comparing centralized, decentralized and distributed system models and developing the novel system structure, which could help to make the proposed system cheaper, faster and easier to implement.

Mathematical models and target function are described in the second chapter of the doctoral thesis. Mathematical models are based on the algorithms, such as convolutional neural network (CNN) and artificial neural network (ANN), and on the novel INN based technology of machine learning for unsupervised safe vehicle control.

The third chapter is devoted to the methods and algorithms used in the dissertation. The choice of electric transport communication and control methods are studied and described. Several algorithms are developed:

- CNN based algorithm for object recognition task;
- algorithm for traffic light red signal recognition task;
- ANN based algorithm for collision probability evaluation and minimization task;
- novel INN based algorithm for collision probability evaluation and minimization task.

The fourth chapter presents the developed prototypes and computer models for testing the proposed algorithms. The electrotechnical description of the developed system with the explanation of the electrical scheme are proposed in this part of doctoral thesis also. The last chapter of the dissertation is devoted to the results of testing the efficiency of the developed algorithms. Developed algorithms were tested with the help of computer modeling.

ANOTĀCIJA

Lietoto transportlīdzekļu skaits pieaug ļoti strauji. Tas izraisa lielāku negadījumu skaitu. Daudzu negadījumu un avāriju cēlonis ir autovadītāja faktors. Risinājums varētu būt bezpilota transporta kontrole. Taču šāda transporta izmantošana noved pie citas problēmas – kā nodrošināt drošu braukšanu un izvairīties no bezpilota transporta sadursmēm.

Promocijas darba zinātniskais jaunums ir iebūvētas sistēmas tehnoloģija, kas spēj mācīties reāllaikā, lai izvairītos no avārijām bez iepriekšējas apmācības, datu kopām un skolotāja. Šī tehnoloģija ir universāla dažādiem transporta veidiem.

Autors to nosauca par imūno neironu tīklu (INN). Tā ir simbioze, kurā tiek izmantoti mākslīga neironu tīkla (ANN) un mākslīgas imūnas sistēmas (AIS) labākās īpašības. Atsevišķi katram no tiem ir virkne trūkumu. ANN nepieciešama iepriekšēja apmācība, tas var būt pārtrenēts utt. AIS nav nepieciešama apmācība, bet optimizācija prasa laiku, un to nevar izmantot reāllaika kontrolei. Jaunā INN sistēma apvieno ANN iespējas un AIS ir pielāgota reāllaika drošības kontrolei.

Promocijas darba galvenais mērķis ir izstrādāt uz imūnneironu tīklu balstītu mašīnmācības tehnoloģiju bez uzraudzības drošai transportlīdzekļa vadībai.

Galvenā hipotēze ir tāda, ka imūnais neironu tīkls var pieņemt vadības lēmumus, lai novērstu transportlīdzekļu sadursmes ar labāku veiktspēju nekā tradicionālais neironu tīkls šajā uzdevumā.

Tika definēti šādi uzdevumi:

- Izpētīt elektrotransporta satiksmes kustības vadības objektus un to mijiedarbību.
- Izpētīt esošos risinājumus bīstamo situāciju atpazīšanai un novēršanai elektrotransportā, kas balstīti uz ANN algoritmiem.
- Salīdzināt centralizētās, decentralizētās un dalītās sistēmas struktūras, izvēlēties definētam uzdevumam piemērotāko un izstrādāt jaunu sistēmas struktūru, kas varētu palīdzēt piedāvāto sistēmu padarīt lētāku, ātrāku un vieglāk ieviešamu.
- Izstrādāt matemātiskos modeļus un algoritmus, kas varētu palīdzēt risināt dažāda veida transporta drošības un sadursmju novēršanas uzdevumus, piemēram, objektu atpazīšanas uzdevumu, luksoforu signālu atpazīšanas uzdevumu, iespējamo krustojuma punktu noteikšanas uzdevumu, sadursmes varbūtības novērtēšanas uzdevumu, sadursmju novēršanas uzdevumu.
- Izstrādāt jaunu uz INN balstītu algoritmu UEV bīstamo situāciju atpazīšanas un novēršanas uzdevumam.
- Izstrādāt UEV elektriskās ķēdes diagrammu ar imūnatmiņu, pamatojoties uz viena borta datoru.
- Veikt datorsimulācijas un pierādīt piedāvāto algoritmu efektivitāti.

Promocijas darba ievads satur uzdevuma formulējumu, esošo problēmas risinājumu veidu analīzi, izvēlētās tēmas pamatojumu un atbilstību, piedāvāto algoritmu un metožu praktisko pielietojumu.

Promocijas darba pirmā daļa ir veltīta centralizēto, decentralizēto un dalīto sistēmu modeļu salīdzināšanai un jaunas sistēmas struktūras izstrādei, kas varētu palīdzēt piedāvāto sistēmu padarīt lētāku, ātrāku un vieglāk ieviešamu.

Matemātiskie modeļi un mērķa funkcija ir aprakstīti promocijas darba otrajā daļā. Matemātiskie modeļi ir balstīti uz tādiem algoritmiem kā konvolucionālais neironu tīkls (CNN) un mākslīgais neironu tīkls (ANN), kā arī uz jauno uz INN balstītu mašīnmācības tehnoloģiju bez uzraudzības drošai transportlīdzekļa vadībai.

Trešā daļa ir veltīta promocijas darbā izmantotajām metodēm un algoritmiem. Tika pētīta un aprakstīta elektrotransporta sakaru un vadības metožu izvēle. Ir izstrādāti vairāki algoritmi:

- CNN algoritms objektu atpazīšanas uzdevumam;
- luksoforu sarkanā signāla atpazīšanas algoritms;
- ANN balstīts algoritms sadursmes varbūtības novērtēšanai un minimizēšanai;
- jauns uz INN balstīts algoritms sadursmes varbūtības novērtēšanai un minimizēšanai.

Ceturtajā daļā ir parādīti izstrādātie prototipi un datormodeļi piedāvāto algoritmu testēšanai. Izstrādātās sistēmas elektrotehniskais apraksts ar elektriskās shēmas skaidrojumu piedāvāts arī šajā promocijas darba daļā.

Promocijas darba pēdējā daļa ir veltīta izstrādāto algoritmu efektivitātes pārbaudes rezultātiem. Izstrādātie algoritmi tika pārbaudīti ar datormodelēšanas palīdzību.

CONTENT

INTRODUCTION ... 10
Situation description ... 10
Industrial and scientific research review ... 10
Review of the industrial research results for different vehicle types safety ... 10
Review of the scientific research and literature about traffic safety ... 13
Topicality of the problem ... 18
Goal and tasks ... 19
Scientific novelty ... 19
Practical application ... 20
Approbation ... 20
Publications ... 21
Summary of review ... 23
1. PROBLEM FORMULATION OF ELECTRIC TRANSPORT SAFETY CONTROL TASK. ... 24
1.1. Situation description ... 24
1.2. Developed system structure ... 24
1.2.1. General safety system structure for electric transport dangerous situation recognition and prevention task ... 24
1.2.2. Subsystem structure for object recognition task ... 33
1.2.3. Subsystem structure for electrical vehicle collision probability evaluation and minimization task ... 34
1.2.4. Novel immune neural network based technology of machine learning for unsupervised safe vehicle control ... 35
1.3. Chapter 1 summary ... 41
2. DEVELOPED MATHEMATICAL MODELS ... 42
2.1. Situation description ... 42
2.2. Abbreviations ... 42
2.3. Mathematical sets of system objects ... 45
2.4. Mathematical model for traffic light red signal recognition task ... 50
2.5. Mathematical model for object recognition task ... 51
2.6. Mathematical model for possible crossing point detection and collision probability evaluation task ... 54
2.7. Mathematical model for the neural network ... 63
2.8. Mathematical model for the immune neural network ... 64
2.9. Chapter 2 summary ... 72
3. DEVELOPED ALGORITHMS FOR ELECTRIC VEHICLE DANGEROUS SITUATION RECOGNITION AND PREVENTION TASK ... 73
3.1. Situation description ... 73

3.2. Developed algorithm for traffic light red signal recognition method for electric transport dangerous situation recognition and prevention task ... 73
3.3. Developed algorithm for convolutional neural network for object recognition for electric transport dangerous situation recognition and prevention task ... 74
3.4. Developed algorithm for electrical transport collision probability evaluation task ... 77
3.5. Developed algorithm for neural network for collision probability evaluation and minimization for electric transport dangerous situation recognition and prevention task ... 81
3.6. Novel developed algorithm for immune neural network for unsupervised collision probability evaluation and minimization for electrical vehicle dangerous situation recognition and prevention task ... 82
3.7. Chapter 3 summary ... 86
4. DEVELOPED PROTOTYPE AND COMPUTER MODEL FOR TESTING PROPOSED ALGORITHMS ... 87
4.1. Situation description ... 87
4.2. Computer model for testing the algorithm of traffic light red signal recognition method for electric transport dangerous situation recognition and prevention task ... 87
4.3. Computer model for testing the algorithm of convolutional neural network for object recognition for electric transport dangerous situation recognition and prevention task ... 88
4.4. Computer model for testing the algorithm of electrical transport collision probability evaluation task ... 90
4.5. Computer model for testing the algorithm of neural network for collision probability evaluation and minimization for electric transport dangerous situation recognition and prevention task ... 92
4.6. Computer model for testing the novel algorithm of immune neural network for unsupervised collision probability evaluation and minimization for electrical vehicle dangerous situation recognition and prevention task ... 94
4.7. Electric scheme with an unsupervised immune memory for unmanned electrical vehicle based on a single board computer ... 98
4.8. Chapter 4 summary ... 101
5. EXPERIMENTAL TESTING OF THE PROPOSED ALGORITHMS ... 102
5.1. Situation description ... 102
5.2. Experimental testing of the proposed algorithm of traffic light red signal recognition method for electric transport dangerous situation recognition and prevention task ... 102
5.3. Experimental testing of the proposed algorithm of convolutional neural network for object recognition for electric transport dangerous situation recognition and prevention task…… ... 102
5.3.1. Experiment Nr.1 ... 103
5.3.2. Experiment Nr.2 ... 104
5.3.3. Experiment Nr.3 ... 105
5.3.4. Experiment Nr.4 ... 108

5.4. Experimental testing of the proposed algorithm of electrical transport collision probability evaluation task ... 111
5.5. Experimental testing of the proposed algorithm of neural network for collision probability evaluation and minimization for electric transport dangerous situation recognition and prevention task ... 114
5.5.1. Neural network experiment with training ... 114
5.5.2. Neural network self-training experiment ... 115
5.5.3. Experimental testing of the algorithm of multiple unmanned electrical vehicles for their collision prevention task ... 116
5.6. Experimental testing of the proposed novel algorithm of the immune neural network for unsupervised collision probability evaluation and minimization for electrical vehicle dangerous situation recognition and prevention task ... 118
5.6.1. Experiment Nr.1 ... 118
5.6.2. Experiment Nr.2 ... 120
5.6.3. Experiment Nr.3 ... 126
5.7. Chapter 5 summary ... 162
CONCLUSIONS ... 165
REFERENCES ... 168

INTRODUCTION

Situation description

Safety is very important for any type of unmanned transport systems. Therefore intelligent computer systems are needed to recognize and prevent dangerous situations.

To prove this statement the introduction provides reviews of such systems at different aspects - scientific and practical and for different modes of transport.

A review of the industrial and scientific research and results are provided in the first part of the introduction. The review of the industrial results is divided according to the type of vehicle: road, railway and aerial vehicles. The review of the scientific research and literature is also divided into various subsections, according to the task, which is solved in the exact research: data analyses and process for dangerous situation recognition and collision probability evaluation task; transport collision probability evaluation and minimization task; electrical vehicle control task.

Topicality of the topic of the doctoral thesis is provided in the third part of the introduction.

Goal and tasks are defined and scientific novelty of the research and practical application of research are also described in the introduction.

Approbation of the scientific work, as well as authors publications are given in the seventh and eighth parts of the introduction.

Industrial and scientific research review

Review of the industrial research results for different vehicle types safety

The number of used electric vehicles is growing very fast because of the fast industry development and high demand. This causes a bigger number of road accidents. That is why transportation safety is one of the most studied subjects nowadays. Road accidents can lead to terrible consequences. Safety is one of the priority tasks in transport domain. A lot of road accidents and crashes are caused by driver factor, which depends on the mood, physical and psychical condition of the driver [1]. Many scientists and world companies are interested in inventing the Autopilot or intelligent system, which could reduce the human factor by smart analysing the situation on the road, making the decisions to avoid dangerous situations, and transmitting the solution from the infrastructure to the vehicles [2]. For example:

- An American automaker, energy storage company Tesla Motors already developed AutoPilot, which provides semi-autonomous driver assist in all Tesla vehicles manufactured since October 2014 [11].
- In January 2019, the Amazon Company introduced the Amazon Scout, a six-wheeled electric-powered delivery robot. Six of the robots are currently making deliveries in a Washington neighborhood during daylight hours, under the supervision of a human associate [6].

- In June 2019, Apple Company confirmed that it had acquired Drive.ai, a self-driving startup. Apple's 'Project Titan' self-driving car could include course-correction systems to assist with bad weather or road conditions [7].
- At the International Automobilausstellung (IAA) 2019 in Frankfurt, Marc Lichte, Head of Design at Audi, presents the Audi AI:TRAIL. The futuristic visionary vehicle is the final car of the AI concept car series and is supposed to function as an electric off-roader [8].
- Tapping into decades-long experience in AI, NVIDIA DRIVE™ hardware and software solutions deliver industry-leading performance to help automakers, truck makers, tier 1 suppliers, and startups make autonomous driving a reality [9]. etc. [10].

Review of the industrial research and results about the road traffic safety

Accidents with transport equipped with Autopilot are happening, despite the new technologies and huge number of inventions. For example, Tesla’s Autopilot system was activated, and the driver’s hands were off the steering wheel when the 2014 Model S crashed into a parked fire truck on a Los Angeles freeway in January 2018. Investigators determined that the car’s Autopilot system had been operating for nearly 14 minutes before the crash occurred. During that period the driver had not touched the steering wheel for more than three minutes before the crash. Teslas, and systems like them, have problems with vehicles stopped on the road ahead of them, especially when they are revealed with little warning [12]. The National Transportation Safety Board determined the probable cause for the crash was the Tesla driver’s lack of response to the fire truck parked in his lane, due to his inattention and overreliance on the car’s advanced driver assistance system [13]. Several Tesla crashes follow a common scenario. A Tesla vehicle operating on cruise control is following another vehicle. The lead vehicle suddenly leaves the lane to avoid something ahead that’s stationary or moving slowly. The Tesla driver assist systems don’t have time to react to the object suddenly in its path, such as a stopped fire truck, and there’s a collision [5]. Also, the statistical data about deaths, related to the use of Tesla cars, proves that there are still a lot of situations, that need to be studied and prevented [17]. Commenting on these situations, representatives of the company said that every time autopilot is activated, the system advises drivers to keep their hands on the wheel and “be prepared to take over at any time” and autopilot requires the “continual and full attention” of the driver. Chief executive officer of the Tesla Elon Mask makes promises every year:

2014 year – “A Tesla car next year will probably be 90 percent capable of autopilot.” [81];

2015 year – “We are probably only a month away from having autonomous driving at least for highways.” [82];

2019 year – “I feel very confident predicting autonomous robotaxis for Tesla next year” [83];

2020 year – “I am extremely confident of achieving full autonomy and releasing it to the Tesla customer base next year.” [84];

2021 year – “My personal guess is that we'll achieve Full Self-Driving this year” [85];

2022 year – “I would be shocked if we do not achieve full-self-driving safer than a human this year.” [86].

But as a result, the provided autonomous drive is still not safer than a person. Advanced driver assistance systems, such as Autopilot, aren’t the same as self-driving cars - meaning the human driver is still responsible for paying attention to the road. Cadillac, Infiniti, Mercedes-Benz, Nissan, and Volvo offer systems like Autopilot, under various names. These systems can maintain a vehicle’s place in the flow of traffic and keep it within the lines of its lane well enough to full drivers into complacency. Only Cadillac’s Super Cruise has a driver-facing camera that will issue warnings if the driver stops looking at the road [5].

Review of the industrial research and results about the railway traffic safety

The situation is even more complicated on the railway, because of the huge weight of the vehicle, long braking distance and no possibility to change the movement trajectory. The railway industry uses the Grade of Automation (GoA) system to rank the level of automation and independence possessed by train control systems:

- GoA 1 denotes a train employing a human train engineer as a driver but with a computer controlling speed;
- GoA 2 trains still have a human driver, but acceleration and braking are automated;
- GoA 3 trains drive autonomously, but human drivers remain on board to manage unexpected situations;
- GoA 4 trains are fully autonomous and capable of driving safely without human crew members in all circumstances.

Rail operators are actively pursuing automation (GoA3/GoA4) for open rail networks, including two major European operators, SNCF (Société nationale des chemins de fer) and Deutsche Bundesbahn (DB).

SNCF the French national rail operator has established two consortia to develop two driverless train prototypes for an autonomous freight train, and autonomous express regional passenger train. Artificial intelligence elements for the autonomous passenger train include an automated driving module, capable of reacting to possible hazards on the guideway and for automatic train door closure on open platforms. SNCF’s aspiration is to deploy semi-autonomous trains by 2023 and fully autonomous trains by 2025 [88]. DB the German national operator is also pursuing driverless operations and has aspirations to achieve this by 2023 [89]. Other National operators such as Network Rail in the UK are focusing on the application of automatic train operation (ATO) at GoA2 (with a driver present) [87],[90],[91].

Review of the industrial research and results about the aerial traffic safety

Unmanned aerial vehicles (UAV) are already used in several rescue operations, inspection of dangerous objects, delivery of specific cargo, as well as for military purposes. Increasingly unmanned vehicles are used in rail transport, such as trams [54] and locomotives. News about working on projects to develop driverless vehicles appropriate for road traffic are being published more and more as well. For example, the UK has started testing unsupervised electrical vehicle (UEV) to transport of supermarket shoppers inside the mall and around the adjacent area in early 2020. Photographers all over the world offer to use modern technologies to gain a better result of taking photos of celebrations, such as weddings. In Australia quadcopters are equipped with artificial intelligence powered software that can distinguish sharks from dolphins, wales, boats, and other marine life in real-time with 90% accuracy [14]. Amazon legally delivered its first Prime order in the United Kingdom in December 2016 [15]. And the use of electrical aerial vehicles still continues to spread to various areas.

In most cases, man-operator, who sets the speed and the trajectory of motion by using the remote control, controls the aerial vehicle. However, it is not productive use of human time and force in the automatization and optimization century. Moreover, man, who is driving aerial vehicle, has ability to make mistakes, which can lead to different unpleasant consequences. That is why the use of intelligence systems in aerial vehicles is developing also. For example, the Patroller multi-sensor long endurance quadcopter was designed for armed forces, intelligence, operational support and maritime surveillance. Featuring a highly modular design, it can be fitted with a wide range of sensors and is easily deployed in foreign theaters of operation. Using its own automatic launch system, it can be deployed from an airport without requiring any change to ground facilities and offers 20-hour endurance with a payload up to 250 kg. The Patroller carries a new-generation very-high-performance imaging system, and the ground station is interoperable with NATO control systems and networks [16].

There are bunches of areas where UAV are used:

- Military purposes;
- Patrolling and searching tasks;
- Order delivery;
- Mapping of landslide, hurricane or other cataclysms affected area;
- Infested crop damage assessment;
- 3-Dimensioinal terrain model construction etc.

Such tasks as patrolling or searching require the team of UAVs working in the same area. In cases of solving tasks with several UAV, one more risk appears – risk of collision between UAV. Let’s suppose that three UAV are looking for the same object, and each of them is programmed to reach his goal. It means that each of them will try to reach approximately the same point coordinates.

Review of the scientific research and literature about traffic safety

Review of the scientific research results about data processing for dangerous situation recognition and collision probability evaluation and minimization

Data must be obtained before analyzing the situation. Julija Freimane et al. in 2017 proposed an article, devoted to the maneuver movements' safety increase using maneuver locomotive identification and distance control. In the article is proposed maneuver locomotives identification and distance control, using RFID technologies passive sensors [79]. In difference with that article, in this research vehicles are not only obtaining their own data by using sensors and cameras, but also transmitting own data and receiving data from other vehicles. This provides an opportunity to analyze the common situation on the polygon and to predict collisions with other moving objects.

Data must be analyzed, after all the data is received. It is necessary to recognize all the objects that were detected by the video camera. Image analysis is the process of extracting data or information from images using image processing techniques. Computers are neccessary to analyze large amounts of data, digital geometry, signal processing, which requires complex calculations to obtain quantitative information using image recognition methods.

Some programs which are based on image processing and parameterization also include object recognition, and image compression [31]. In 2005, Ioannou et al. [32] described an emotion recognition system based on neurophysiological systems. This system combined the psychological discovery of emotion representation with the analysis and evaluation of facial expressions. The author classified facial expression using a continuous 2D emotion area. Thanh et al. [33] in 2007 proposed the use of the Noise Generalized Fuzzy Inference System (GFIS) for noise image processing, which had a multi-layered neo-physiological structure that combined both the Mandani model and the Takaggi-Sugeno (T-S) fuzzy model to form a hybrid fuzzy system. In 2009, Chakrapani and Soundararajan [34] proposed ANFIS for fractal image compression (FIC). The authors used an ANFIS-based network to classify gray-level image domain cells based on statistical properties to perform fractal image compression. In scientific paper [35] authors advise to address the aforementioned challenges in traffic light detection, by identifying the traffic light region of interest in the image, using automatically generated saliency maps. The traffic light detection system is implemented and proposed for an advanced driver assistance system and autonomous vehicle using an on-board color camera and GPS sensor obtained information. The proposed algorithm consists of two steps: an off-line saliency map generation step and a real-time traffic light detection step.

Neural network is proposed for the traffic sign recognition process [36], for the road surface traffic sign detection [37], risk evaluation [38] etc. Simple fuzzy severity estimation model has a comparable performance to more complicated systems such as the CoTEC (CoOperative Traffic congestion detECtion) [39].

Convolution neural network (CNN) is less studied method than traditional artificial neural network (ANN). But existing articles show that method is useful in object recognition process. For example, Rongqiang Qian et al. in 2016 presented research about traffic sign recognition with convolutional neural network (CNN) based on max pooling positions [40]. In comparison with previous methods, which usually use CNN as feature extractor and multilayer perception as classifier, authors proposed max pooling positions as an effective discriminative feature to predict category labels. Weishan Zhang et al. in 2015 wrote research about an integrated approach for vehicle detection and type recognition [41]. The described result shows that good results can be achieved with the combination of traditional moving object detection method with deep CNNs. Rohan Ghosh et al. in 2014 presented a paper with a research description about Real-Time Object Recognition and Orientation Estimation Using an Event-Based

Camera and CNN [42]. CNNs are recognised as a powerful, bio-inspired tool for visual classification, providing high accuracy for tasks such as digit classification [43]. In 2016 Chen Liang et al. introduced an adaptive learning algorithm with an adaptive learning rate for human face recognition [44]. Amel Tuama et al. in the research about camera model identification with the use of deep CNNs [45] evaluated the efficiency of using CNNs for source camera model identification based on deep learning and convolutional neural networks. The contribution represents a big challenge since it is quite different from existing conventional techniques for camera identification. Author tried a small net by tuning the AlexNet model. Chang-Di Huang et al. in 2015 proposed a three stream CNN is for recognize human actions such as fall floor and baby craw [46]. Three stream CNN considerate spatial information, temporal information and movement of human for classify the human activities. Huang Guan et al. in 2016 proposed a Real-time lane-vehicle detection and tracking system in research [47]. A novel and effective method based on convolutional neural networks was used to achieve high-quality and real-time vehicle detection.

Ivars Namatevs et al. in 2019 proposed an efficient method for training a neural network to count moving objects in a video, while another neural network concurrently prepares a labeled dataset for the first one. The method shows that by using CNN, the computing power and speed-up time for training a Recurrent Neural Network with a Long Short-Term Memory cell for counting moving objects can be decreased. [70]. Roberts Kadikis in 2018 presented a doctoral thesis, that describes novel video-based moving object detectors that process only a line of pixels in a frame [71]. CNN is also applied in the system that is proposed in this doctoral thesis, but only as a supporting method for obtained data processing.

Tianhua Chen et al. in 2022 proposed the results of the application of CNN in e-sport for traffic intensity selection as a feature combined with deep learning classifiers for streaming videos classification with different resolutions and frame rates per second [67]. Elans Grabs et al. in 2022 presented an article, that contains results of training and testing machine learning models with captured network traffic data. The main goal was to perform classification of video traffic in computer networks. Multiple performance metrics have been evaluated for commonly used classic supervised machine learning algorithms, as well as more advanced convolutional neural network model [68]. In difference with those researches, this one represents the use of CNN for real time object recognition process for electric transport dangerous situation recognition and prevention task.

An algorithm of the electrical vehicle collision risk solving system has been proposed in this doctoral thesis, based on basic collision risk and vulnerabilities of accidents. Similar algorithm was proposed in Yancai Hu and Gyei-Kark Park scientific paper [64], but in difference with that paper, this research is devoted to the risk assessment and collision prevention task for different types of electrical vehicles.

Juris Kreicbergs et al. in 2021 mentioned in his research, that the current Latvian road traffic safety statistics is well below the average level of the EU member states and even the pandemics did not cause similar reduction as in most member states [72]. Other scientists in 2021 made a road traffic safety analysis of different junction types on the state roads [73]. Vilma Jasiuniene et al. in 2020 presented an analysis about the pedestrian risk group – older road users. The extensive use of appropriate road safety management solutions is required to improve traffic conditions for older pedestrians and thus to encourage their mobility, in addition to educational activities and control measures [75]. Davis Buss et al. in 2019 made an analysis and stated that the majority of road accidents demonstrates that the weakest part of human-machine system limiting its efficency and reliability is a human [78]. This proves that transportation process is still not completely safe and requires novel methods for collision probability minimization.

Before minimizing the collision probability is necessary to calculate the possible collision probability. One of the ways to do this is to detect the crossing point of two objects. Usually, the crossing point is calculated as one point of specific coordinates, but in real life vehicles are 3D models with specific sizes and length. So even if centers of two vehicles are not crossing, they still can go tangentially and collide. In [48] authors use a sensor-based recognition and avoidance strategies. Collision detection helps them to gain prior information around an obscure element in the workspace while collision avoidance re-courses to locate a substitute way around the unidentified element. In [49] conflict resolution was achieved with obstacle trajectory data taken from a simulated camera and range-finder in the presence of their respective measurement uncertainties. In difference with that research, arthor of this research propose a system, where sensors and cameras are not the main part of the system. Data is received by using GNSS or GPS and radio frequency models for transmitting the information between vehicles. In 2014 anti-collision system for navigation inside an UAV using fuzzy controllers and range sensors [50] was proposed. In that research authors were working to provide a system that will help to prevent UAV collisions with obstacles, but nothing about collisions with other UAVs was said.

Aleksandrs Bitins et al. in 2021 presented an article, that reviews using adverse event pyramids to assess probabilities in airline safety management [74]. In difference with that research, in this doctoral thesis algorithm of the electrical vehicle collision risk solving system has been proposed based on basic collision risk and vulnerabilities of accidents.

Vilma Jasiuniene et al. in 2020 proposed a paper, that describes both the empirical Bayes method for predicting road accidents and the application of one of the road safety indicators – the expected fatal accident density – to determine five road safety categories across the road network [76]. Situation analysis was also used in this doctoral thesis, to define safety level of the situation, but the main novel algorithm, proposed in this thesis is devoted to collision probability minimization.

Review of the scientific research and literature about electric transport control

Real-time traffic flow parameter estimation from unmanned aerial vehicle video based on ensemble classifier and optical flow was proposed in 2019 [30]. In difference with that paper, the system provided in this research, does not need additional infrastructure embedded devices or objects. In 2016. S. Roelofsen, A. Martinoli and D. Gillet proposed a collision avoidance algorithm for unmanned aerial vehicles (UAVs) with limited field of view constraints [29]. Authors presented a safe collision avoidance algorithm based on potential fields for fixed-wing UAVs with constrained field of view sensors such as cameras. In difference with mentioned research, this research deals with a collision avoidance of different types of electrical vehicles. Articles [27] and [26] also deal with motion control tasks of UAVs and both have reference on Lyapunov method. In article [27] authors are focused on theoretical justification of the applicability of Lyapunov method for UAVs motion control. The authors of article [26] describe usage of this method for neuro adaptive controller without visual situation monitoring and immune neuro-fuzzy application to calculate collision probability. Comparing motion control of the aerial vehicles with surface transport allows concluding that of surface autonomous vehicles have higher probability of collision, because there is no ability to change the altitude component. Hierarchical collision-avoidance strategy is proposed as theoretically approach for high-speed self-driving vehicle collision avoiding task solving [25]. Fuzzy logic and FMCW radar (Frequency-Modulated Continuous Wave) usability for motion control task solving are described in research [24], but this system also is not provided with surrounding environment visual monitoring elements or sensors. Real-time traffic flow parameter estimation from unmanned aerial vehicle video based on ensemble classifier and optical flow was proposed in 2019 [55]. In difference with that paper, the system provided in this research, does not need additional infrastructure embedded devices or objects. Andrejs Bondarenko in 2020 presented a doctoral thesis, that is devoted to knowledge extraction from trained artificial neural networks [69]. Research described in this doctoral thesis proposes a novel ANN, that does not require previous training, the proposed system is unsupervised and does not need a teacher. Nguyen Dinh Dung in 2019 presented a study, that proposes drone-following models for managing drones in the transportation management system in smart cities. These models are based on the initial idea that drones flight towards a leading drone in the traffic flow [77]. But in difference with this doctoral thesis, in [77] several important situations were not considered, like safe distance is measured not only in the drone directly in front but also in two drones beside, or increasing the number of drones. Mihails Gorobecs et al. in 2015 proposed an article about intelligent electric vehicle motion and crossroad control using genetic algorithms [80]. Donato Repole in 2021 presented doctoral thesis, that proposes an application case for the VHDL based “neuro-fuzzy controllers” researches, aiming the use of learning/training controller’s capability to off-load the mechanical design. Author describes a parallel computing neuro-fuzzy networks for unmanned vehicles [4]. In different with mentioned articles, this research is devoted to the use of a novel immune neural network (INN) algorithms.

In 2021 Raivo Sell et al. published a book, that was devoted to the scientific results: “Unmanned Electrical Vehicles and Autonomous System Simulation”. That book and its offshoots were prepared to provide a comprehensive introduction into the domain of the autonomous system [3]. In the book was also mentioned artificial intelligence for autonomouse drive, but no specific solutions, like artificial neural neural (ANN), were mentioned there.

Aleksandrs Kornejevs et al. in 2019 presented research “Adaptive Traction Drive Control Algorithm for Electrical Energy Consumption Minimisation of Autonomous Unmanned Aerial Vehicle” [56] and in 2020 presented a research “Neural Network Based UAV Optimal Control Algorithm for Energy Efficiency Maximization” [18]. In 2018 G. Strupka proposed research about algorithm for unmanned aerial vehicle to supervise applications for civil and power engineering tasks [57]. These authors advise to use artificial intelligence, such as ANN, for unmanned aerial vehicle control, but the main aim was to maximize energy efficiciency, but not to minimize collision probability.

The traditional ANN requires self-tuning for every single situation. The traditional ANN is using current weights for next re-training, but it takes a time because of the high diversity of situations [22]. The proposed in this doctoral thesis INN has the immune memory (IM) from previously successfully prevented collision situations, so finding the similar situation in this memory and using the stored weights as a base for re-training allow significantly reduce the self-tuning time to solve the current problem. Dong Hwa Kim proposed a similar solution in August 2002 [19]. The main idea of Dong Hwa Kim is a retraining of a static neural network, by using IM based algorithm. In [19] solution, creating initial population (set of chromosomes) randomly, i.e. a population of fuzzy weights of neural networks which are randomly specified, is used. In difference with that research, the main idea of this doctoral thesis is that input data are stored in the IM together with weights, that were used previously for solving this situation, that helps to reduce calculation time, that is very important for real time systems. Also, the proposed novel algorithm allows to retrain the neural network. In this doctoral thesis no initial population is needed. Training can start even with an empty IM. The found solutions will be those that will be written in the IM and further adjusted.

Rodrigo Pasti in 2006 presented research, where neural network was also strengthened by IM [23]. But in that research method was not based on a quality function to evaluate the solutions. The author was benchmarking his results against the best results in literature.

Topicality of the problem

It’s not just Tesla who is struggling to perfect self-driving technology. Honda is working on it too. As are Waymo owned by Google and Cruise owned by General Motors. All these companies predicted they’d have full self-driving cars by 2020. But as yet, none of them have. Creating full self-driving vehicle is much harder than vehicle makers initially thought. There is a wide range of possible risks and elements to consider.

This fact and completed reviews allow concluding that the problem of improving transport safety by the artificial intelligence systems is topical and requires scientific contribution for its solution.

The doctoral thesis is related to the safe motion of electric transport providing novel immune neural network-based algorithms for its control. The doctoral thesis contributes to the safety improvement of multiple collaborating unmanned electrical vehicles moving and performing own tasks in the same area by the research and development of immune neural network technology. The developed technology provides the ability for continuous unsupervised self-learning to avoid collisions by changing speed and trajectory maximizing the efficiency of task performance in real-time.

Goal and tasks

The goal of the dissertation is to develop immune neural network based technology of machine learning for unsupervised safe vehicle control.

The main hypothesis is that an immune neural network can make control decisions to prevent vehicle collisions with better performance than a traditional neural network in this task.

Following tasks were defined:

- To study the objects of electric transport traffic movement control and their interaction.
- To study the existing solutions for the recognition and prevention of dangerous situations in electric transport, which are based on the algorithms of artificial neural network.
- To compare centralized, decentralized and distributed system structures, to choose the most suitable one for the proposed task and to develop the novel system structure, which could help to make the proposed system cheaper, faster and easier to implement.
- To develop mathematical models and algorithms, that could help to solve different types of transport safety and collision prevention tasks, such as object recognition task, traffic light signal recognition task, possible crossing point detection task, collision probability evaluation task, collision prevention task.
- To develop novel immune neural network based algorithm for unmanned electrical vehicle dangerous situation recognition and prevention task.
- To develop electrical circuit diagram with an immune memory for unmanned electrical vehicle based on a single board computer.
- To make computer simulations and to prove the efficiency of the proposed algorithms.

Scientific novelty

The most significant scientific novelty of the doctoral thesis is the immune neural network technology inspired by two biological systems – immune system and neural networks and their artificial analogs. The developed novel mathematical models and algorithms for this technology allows skipping the preliminary supervised training step and adapted for real-time continuous unsupervised self-learning of unmanned electrical vehicle to recognize the dangerous situation and prevent the collision by making control decisions autonomously keeping the structures and weights of separate neural networks in the immune memory and retraining them to minimize the collision probability and maximizing the performance.

New mathematical models and algorithms for possible crossing point detection, for collision probability evaluation and for collision probability minimization by the neural network are developed in the thesis for this purpose.

Additional safety improvemt mathematical models and methods for object recognition and traffic light signal recognition are developed in the thesis and integrated in the proposed system.

Practical application

Algorithms developed in the dissertation can be implemented in intelligent electrical vehicle control systems to avoid crashes and minimize the risk of collisions. The results of the developed algorithms offer solutions to the tasks of data collection from video surveillance cameras, sensors, cloud databases and other objects of intelligent transport infrastructure, information processing, identification of potentially dangerous situations, risk assessment and decision-making on measures to avoid an accident.

Developed algorithms allow implementing the computer modeling and simulation of optimal control system for electric transport to recognize and prevent dangerous situations. Proposed algorithms are multifunctional and can be implemented into different types of vehicles without mandatory changes and improvement of the infrastructure objects. However, the intelligent infrastructure may provide additional inputs to the developed system.

Approbation

The results of the work were presented and discussed at 12 international scientific conferences:

1. International conference “2020 IEEE 61th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON)”, report “Unsupervised Transport Vehicle Control: Simulation Study and Performance Results”, 2020, Riga, Latvia.
2. International conference “2019 IEEE 60th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON)”, report “Modelling and Simulation of Transport Collision Probability Recognition Algorithm for Traffic Safety”, 2019, Riga, Latvia.
3. International conference “Applications of Intelligent Systems (APPIS 2019)”, report “Machine Learning Algorithm of Immune Neuro-Fuzzy Anti-collision Embedded System for Autonomous Unmanned Aerial Vehicles’ Team”. 2019, Las Palmas de Gran Canaria, Spain.
4. International conference “20th European Conference on Power Electronics and Applications, EPE’18 ECCE Europe”, report “Algorithm for Immune Neural Network in Transport Collision Prevention Control System of Unmanned Electrical Vehicle”. 2018, Riga, Latvia.
5. International conference “2018 IEEE 59th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON)”, report “Algorithm of Signal Recognition for Railway Embedded Control Devices”. 2018, Riga, Latvia.
6. International conference “22nd International Scientific Conference. Transport Means 2018”, report “Self-Organized Learning Algorithm for Immune Neuro-Fuzzy Anti-collision System of Autonomous Unmanned Aerial Vehicles’ Team”. 2018, Trakai, Lithuania.

7. International conference “12th International Conference Intelligent Technologies in Logistics and Mechatronics Systems”, report “Control Algorithm of Multiple Unmanned Electrical Aerial Vehicles for Their Collision Prevention”. 2018, Panevėžys, Lithuania.
8. International conference “2017 IEEE 58th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON)”, report “Convolutional Neural Network in Turn Recognition Tasks for Electric Transport Safety”. 2017, Riga, Latvia.
9. International symposium “25th International Symposium on Dynamics of Vehicles on Roads and Tracks (IAVSD 2017)”, report “Convolutional Neural Networks of Active Railway Safety System with Braking Dynamics Prediction. Dynamics of Vehicles on Roads and Tracks”. 2017, Rockhampton, Queensland, Australia.
10. International conference “31st European Conference on Modelling and Simulation”, report “Modeling and Simulation of Public Transport Safety and Scheduling Algorithm”. 2017, Budapest, Hungary.
11. International conference “Building up Efficient and Sustainable Transport Infrastructure (BESTInfra)”, report “Innovative neuro-fuzzy system of smart transport infrastructure for road traffic”. 2017, Prague, Czech Republic.
12. International conference “. 57th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON)”, report “Immune Algorithm and Intelligent Devices for Schedule Overlap Prevention in Electric Transport”. 2016, Riga, Latvia.

Publications

The main results of the work have been published in 16 papers in international conference proceedings and journals:

1. Beinarovica, A., Gorobetz, M., Ribickis, L. Immune Neuro-Fuzzy Network Based System for Collision Free Motion Control of Unmanned Electrical Vehicles. In: 25th European Conference on Power Electronics and Applications (EPE 2023): Proceedings, Aalborg, Denmark, 4.-8. September 2023.
2. Gorobetz, M., Ribickis, L., Beinarovica, A., Kornejevs A. Immune Neural Network Machine Learning of Autonomous Drones for Energy Efficiency and Collision Prevention. In: Open access chapter “Drones - Various Applications”, September 2023.
3. Beinarovica A., Gorobetz M., Alps I. Unsupervised Transport Vehicle Control: Simulation Study and Performance Results. In: 2020 IEEE 61st International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON 2020): Proceedings, Riga, Latvia, 5-7 November 2020, pp. 303-308.

4. Beinarovica A., Gorobetz M., Alps I. Modelling and Simulation of Transport Collision Probability Recognition Algorithm for Traffic Safety. In: 2019 IEEE 60th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON 2019): Proceedings, Riga, Latvia, 7-9 October 2019, pp. 1-6.
5. Beinarovica A., Gorobetz M., Ribickis L., Levcenkov A. Machine Learning Algorithm of Immune Neuro-Fuzzy Anticollision Embedded System for Autonomous Unmanned Aerial Vehicles Team. In: 2nd International Conference on Applications of Intelligent Systems (APPIS 2019): Proceedings. ACM International Conference Proceedings Series, Las Palmas de Gran Canaria, Spain, 7-9 January 2019, pp. 1-8.
6. Beinarovica A., Gorobetz M., Alps. I., Levcenkov A. Algorithm for Immune Neural Network in Transport Collision Prevention Control System of Unmanned Electrical Vehicle. In: 2018 20th European Conference on Power Electronics and Applications (EPE'18 ECCE Europe): Proceedings, Riga, Latvia, 17-21 September 2018, pp. 1-8.
7. Beinarovica A., Gorobetz M., Alps. I., Levcenkov A. Algorithm of Signal Recognition for Railway Embedded Control Devices. In: 2018 IEEE 59th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON 2018): Proceedings, Riga, Latvia, 12-14 November 2018, pp. 1-5.
8. Beinarovica A., Gorobetz M., Levcenkov A. Control Algorithm of Multiple Unmanned Electrical Aerial Vehicles for Their Collision Prevention. In: 12th International Conference Intelligent Technologies in Logistics and Mechatronics Systems: Proceedings, Panevėžys, Lithuania, 26-27 April 2018, pp. 37-43.
9. Beinarovica A., Gorobetz M., Levcenkov A. Self-Organized Learning Algorithm for Immune Neuro-Fuzzy Anti-collision System of Autonomous Unmanned Aerial Vehicles’ Team. In: Transport Means 2018: Proceedings of the 22nd International Scientific Conference. Part 3, Trakai, Lithuania, 3-5 October 2018, pp. 1334-1341.
10. Beinarovica A., Gorobetz M., Levcenkov A. Convolutional Neural Networks of Active Railway Safety System with Braking Dynamics Prediction. In: Dynamics of Vehicles on Roads and Tracks: Proceedings of the 25th International Symposium on Dynamics of Vehicles on Roads and Tracks (IAVSD 2017): Proceedings, Rockhampton, Australia, 14-18 August 2017, pp. 953-958.
11. Beinarovica A., Gorobetz M., Levcenkov A. Modeling and Simulation of Public Transport Safety and Scheduling Algorithm. In: 31st European Conference on Modelling and Simulation (ECMS 2017): Proceedings, Budapest, Hungary, 23-26 May 2017, pp. 215-221.
12. Beinarovica A., Gorobetz M., Levcenkov A. Convolutional Neural Network in Turn Recognition Tasks for Electric Transport Safety. In: 2017 IEEE 58th International Scientific Conference on Power and Electrical Engineering of Riga Technical

University (RTUCON 2017): Proceedings, Riga, Latvia, 12-13 October 2017, pp. 231-236.
13. Beinarovica A., Gorobetz M., Levcenkov A. Innovative nauro-fuzzy system of smart transport infrastructure for road traffic safety. In: Building up Efficient and Sustainable Transport Infrastructure (BESTInfra 2017): Proceedings, Prague, Czechia, 21-22 September 2017, pp. 1-8.
14. Alps I., Gorobetz M., Beinarovica A., Levcenkov A. Immune Algorithm and Intelligent Devices for Schedule Overlap Prevention in Electric Transport. In: 2016 57th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON 2016): Conference Proceedings, Riga, Latvia, 13-14 October 2016, pp. 191-197.
15. Beinarovica A., Gorobetz M., Levcenkov A. Algorithm of Energy Efficiency Improvement for Intelligent Devices in Railway Transport. In: Electrical Control and Communication Engineering (ECCE journal), Riga, Latvia, 2016, pp. 29-34.
16. Gorobetz M., Ribickis L., Alps I., Beinarovica A. Patent application "Bezpilota Transporta Līdzekļa Sadursmju Novēršanas Iekārta ar Pašapmācošo Imūno Atmiņu".

Summary of review

Review of the industrial and scientific research show that fully autonomous drive for the electric transport control is not developed yet. Exist a lot of limitations, because of which it is not possible to implement autonomous UEV in the infrastructure. Electrical car factories, which produce autonomous electrical vehicles, already exist nowadays, but produced vehicles do not provide completely autonomous drive. The driver still must attend in case of unforeseen situations. That is why the question about the workable algorithms for the autonomous unmanned electric drive control is still open.

The purpose of this research is to study methods of artificial intelligence for safety tasks and to develop the new algorithms for collision prevention of electrical vehicles. For this purpose existing system models are compared and system structure is developed and described in the next chapter of doctoral thesis.

1. PROBLEM FORMULATION OF ELECTRIC TRANSPORT SAFETY CONTROL TASK

1.1. Situation description

Existing systems do not provide a fully autonomous drive, that is why the task of developing the fully autonomous vehicle is still actual. First chapter of the doctoral thesis is devoted to the comparing centralized, decentralized and distributed system models and developing the novel system structure, which could help to make the proposed system cheaper, faster and easier to implement.

1.2. Developed system structure

1.2.1. General safety system structure for electric transport dangerous situation recognition and prevention task

Standard set of the Autopilot equipment consists of elements, which would help to increase the safety level: emergency braking system; wireless network equipment; navigation equipment; remote control cameras; computing unit (microcontroller); steering attachment; steering sensor; collision avoidance equipment (LIDAR, Ultrasonic sensors).

System models can be divided into centralized, decentralized and distributed.

1.1 table

Advantages and disadvantages of the centralized system [58]

Advantages	**Disadvantages**
It has a relatively high degree of security, because of the existence of only one kind of access to the system	The system performance for each user decreases when many users try to attach simultaneously
More cost efficient for small systems up to a certain limit – As the central systems take less funds to set up, they have an edge when small systems have to be built	Highly dependent on the network connectivity – System can fail if the nodes lose connectivity as there is only one central node
Quick updates are possible – Only one machine to update	No graceful degradation of system – abrupt failure of the entire system
Dedicated resources (memory, CPU cores, etc)	Less possibility of data backup. If the server node fails and there is no backup, you lose the data straight away
	Difficult server maintenance – There is only one server node and due to availability reasons, it is inefficient and unprofessional to take the server down for maintenance

In a centralized system, all computing power may be allocated to one user when no other users are attached to the system. Consequently, the execution time of all users' applications will be increased if the mainframe serves many users. [58] The main advantages and disadvantages of the centralized system are listed in table 1.1.

Decentralization is a system model structure in which the decision-making is made at various levels. Typically, decentralized systems are divided into smaller segments or groups in order to make it easier to measure the performance of the common system and the individuals within each of the sub-groups. [60] The main advantages and disadvantages of the decentralized system are listed in table 1.2.

1.2 table

Advantages and disadvantages of the decentralized system [60]

Advantages	**Disadvantages**
Quick decision and response times – it is important for decisions to be made and implemented in a timely manner	Difficult to achieve global big tasks – No chain of command to command others to perform certain tasks
Minimal problem of performance bottlenecks occurring – The entire load gets balanced on all the nodes; leading to minimal to no bottleneck situations	Difficult to know which node responded – When a request is served by a decentralised system, the request is actually served by one of the nodes in the system but it is actually difficult to find out which node indeed served the request
High availability – Some nodes (computers, mobiles, servers) are always available/online for work, leading to high availability	Difficult to know which node failed – Each node must be pinged for availability checking and partitioning of work has to be done to actually find out which node failed by checking the expected output with what the node generated
More autonomy and control over resources – As each node controls its own behavior, it has better autonomy leading to more control over resources	No regulatory oversight

A distributed operating system is an operating system that runs on several machines whose purpose is to provide a useful set of services, generally to make the collection of machines behave more like a single machine. Distributed operating systems typically run cooperatively on all machines whose resources they control. These machines might be capable of independent operation, or they might be usable merely as resources in the distributed system. [59] The main advantages and disadvantages are listed in table 1.3.

1.3 table

Advantages and disadvantages of the distributed system [59]

Advantages	**Disadvantages**
Distributed systems clearly have a price/performance advantages over more traditional systems	Conventional way of logging events by absolute time they occur is not possible here
Incremental growth. The more processers will be added, the more computing power wil got	Difficult to achieve consensus
Reliability. A few parts of the system can be down without disturbing people using the other parts	
Distributed systems have low latency because of high geographical spread, hence leading to less time to get a response	

Developed in this research novel safety system structure for electric transport dangerous situation recognition and prevention task is proposed into two versions: centralized and distributed.

Developed centralized system structure is shown in the figure 1.1.

Developed centralized system structure consists of following groups and elements:

1. Objects – other objects on the way of motion or in the visibility range.

1.1. Level crossing – different type of level crossings.

1.1.1. Signal receiver – component to receive the signal about the train approaching.

1.1.2. Radio frequency model – to communicate with other devices embedded into the electrical vehicles and control center.

1.1.3. Power supply – to supply electric power to an electrical load of the components of the level crossing device.

1.1.4. Microcontroller – to obtain the data about the circumstances in the level crossing, to make necessary calculations.

1.1.5. Barrier arrow – component to control the opening and closing the level crossing for different movement directions.

1.2. Traffic light – different types of the traffic light.

1.2.1. Radio frequency model – to communicate with other devices embedded into the electrical vehicles, and control center.Signal lamps – indications, that allow or disallow the movement.

1.2.2. Sensors – components, that obtain the information about electrical vehicles, which are nearby.

1.2.3. Power supply – to supply electric power to an electrical load of the components of the traffic light.

1.2.4. Microcontroller – to control the traffic light.

1.3. Road signs – signs, that limit the possible movement parameters on the concrete road section.

2. Electrical vehicle.

2.1. GNSS receiver – to receive coordinates of the electrical vehicle location.

2.2. Radio frequency model – to communicate with other devices embedded into other electrical vehicles and infrastructure objects.

2.3. Sensors.

2.3.1. Videocamera – to receive the images of the environment.

2.3.2. Proximity sensor – to receive the data about other electrical vehicles and other objects nearby.

2.3.3. Vibration sensor – to receive the data about the vibration level for road surface recognition.

2.3.4. Humidity sensor – to receive the data about the weather for road surface as well as visibility recognition.

2.3.5. Wheel rotation frequency sensor – to receive the data of the electrical vehicles speed.

2.3.6. Light level sensor – to receive the data about the weather and time of the day for visibility recognition.

2.3.7. Weather and visibility sensor – to receive the data about the weather and time of the day for visibility recognition.

2.3.8. Windshield pollution sensor – to receive the data of the windshield condition for visibility recognition.

2.3.9. Compass – to receive the data of the electrical vehicle movement direction.

2.3.10. Speed sensor – to receive the data of the electrical vehicle speed and acceleration.

2.4. Driver control system – to control actions of the driver. If the driver received the indications on the information display but is not reacting in time, then driver control system sends the signal to the electrical vehicle control system, and the proposed system intervenes in the electrical vehicle control.

2.5. Driver informing system – to inform the driver about necessary movement parameters change.

2.5.1. Display actuator – a component to control the information display.

2.5.2. Information display – to show the information for the driver.

2.6. Control system – to control the electric drive, braking and steering systems.

2.6.1. Brake actuator – a component to control the braking system.

2.6.2. Braking system – to control the stop of the electrical vehicle.

2.6.3. Drive actuator – a component to control the electric drive.

2.6.4. Electric drive – to control the acceleration and speed of the electrical vehicle.

2.6.5. Steering system actuator – a component to control the steering system.

2.6.6. Steering system – to control the trajectory of motion of the electrical vehicle.

2.7. Mechanical part – part of the electrical vehicle that is responsible for the electrical vehicles moving.

3. Control center – the main control center, that is obtaining the data from electrical vehicles and other objects, makes evaluation of the collision probability, calculates necessary movements parameters changes and sends the found solution to electrical vehicles.

3.1. Radio frequency model – to communicate with other devices embedded into the electrical vehicles, level crossings and traffic lights.

3.2. Database – common storage of the obtained data, received from the electrical vehicles, level crossings and traffic lights.

3.3. Computer – to calculate motion parameters by using the proposed algorithms, to evaluate the collision probability, to calculate motion parameters in case of minimizing the found collision probability, to send the information to the drivers information system and to control the electric drive.

3.3.1. Object recognition algorithm is using incoming video data for convolutional neural network (CNN) to recognize other electrical vehicles and vehicles that do not have the proposed embedded device as well as other objects in the visibility range.

3.3.1.1. Moving object recognition algorithm – to recognize moving objects such as human being, animals, other vehicles in the visibility range.

3.3.1.2. Static object recognition algorithm – to recognize static objects such as fallen tree or broken car on the way of motion.

3.3.1.3. Road recognition algorithm – to recognize the presence of turns and road profile.

3.3.1.4. Visibility level recognition algorithm – to determine the visibility range.

3.3.2. Danger lever recognition algorithm – to calculate the possible crossing point of two or more objects and to evaluate the collision probability by taking into account the input data.

3.3.3. Opportunity assessment algorithm based on ANN is responsible for generation of the optimal speed curves to avoid the crash and in the same way to perform the safest and the smoothest motion as possible.

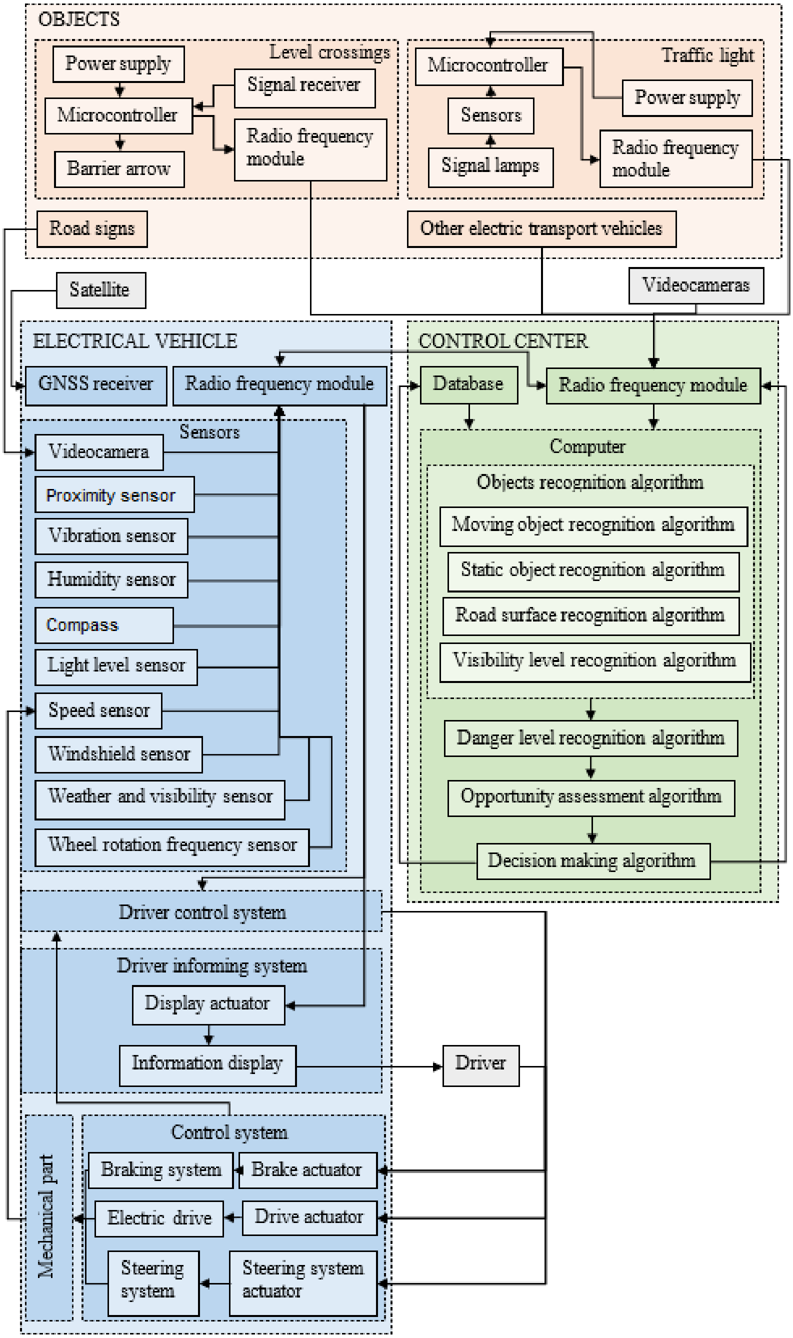
3.3.4. Decision making algorithm – makes decisions about possible actions to decrease risk and controls the motion process according the calculated curves to achieve the expected speed in target point on the route.

4. Videocamera - to receive the images of the environment.

5. Satellite – component to obtain the coordinates of the object position in real time.

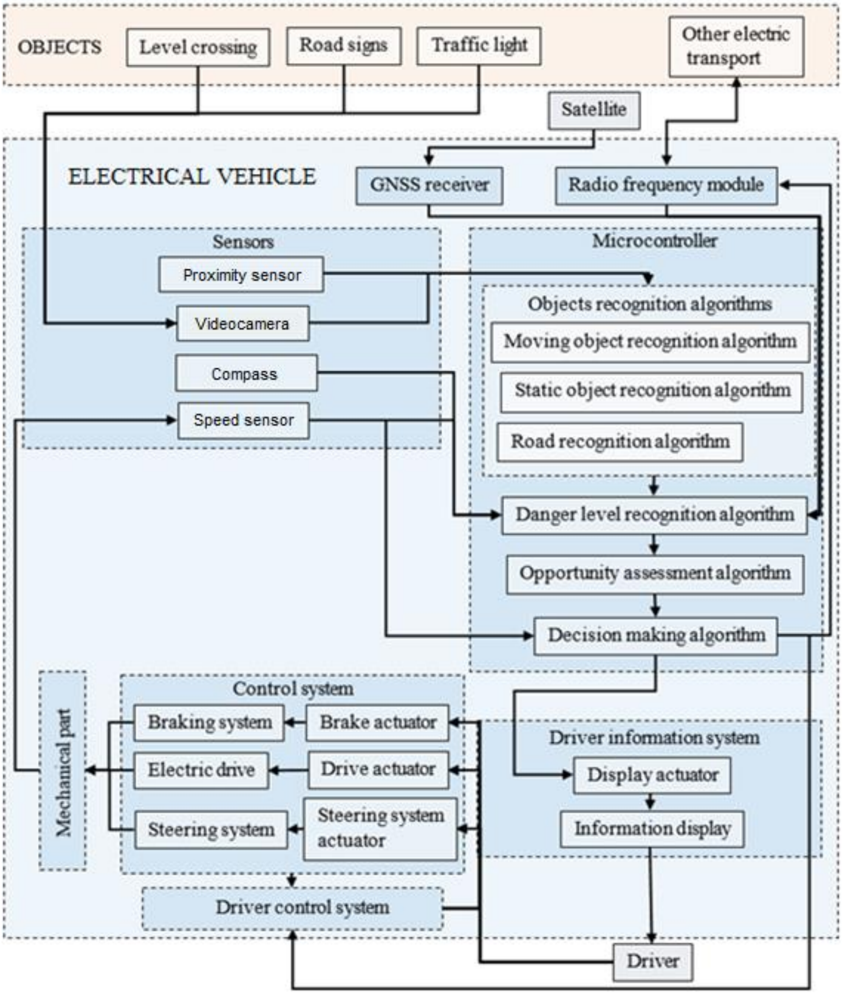
6. Driver – to control the electrical vehicle.

Distributed system structure, which was developed in this research, is shown in the figure 1.2. The number of sensors was reduced in distributed system in order to focus on the object recognition and novel algorithm of the collision probability minimization tasks also.



1.1 fig. Centralized system structure.

There is no need to use common control center as well as smart level crossings and traffic lights that could send signals to the control center in the proposed system structure. All these functions are performed by the microcontroller, embedded in each electrical vehicle, where the object recognition process and risk assessment are done, as well as opportunity assessment and decision making about necessary movement parameters change are done.



1.2 fig. Distributed system structure with redused number of components.

The control element of the electrical vehicle is not centralized. This method makes the proposed system easier to implement and cheaper, because there is no need to embed devices into the infrastructure. The proposed system can work anywhere.

The comparison of the centralized and distributed systems was done with a view to choose the best model. Where:

1 – the best solution;
0 – worst solution.

1.4 table

Comparison of the centralized and distributed systems

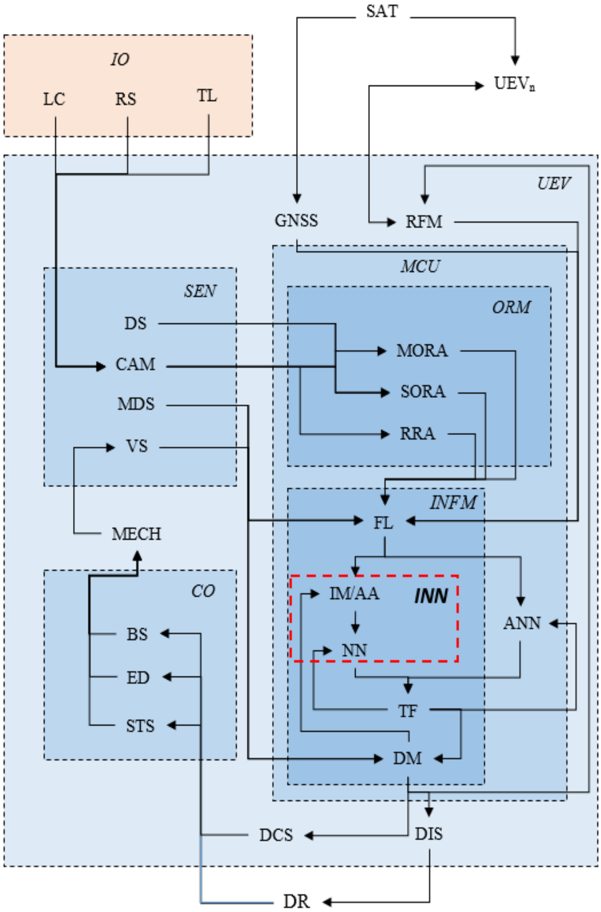
Nr.	**Parameter**	**Centralised**	**Distributed**
1	price for vehicle owner	1	0
2	price for infrastructure owner	0	1
3	place or area of application	0	1
4	possibility to use in different transport (road, rail, area, marine)	0	1
5	difficulty level to reach consensus	1	0
6	calculation and data receiving delay	0	1
7	general clock	1	0
8	implementation difficulty level	0	1
9	algorithm difficulty level	1	0
10	speed of the reaction	0	1
11	one unit influence on all the system	0	1
12	number of users influence	0	1
13	dependent on the network connectivity	0	1
14	reliability, system failure risk	0	1
15	data security	1	0
16	possibility of data backup	1	0
17	statistics and log files	1	0
	SUM:	7	10

The results of comparison show that distributed system is preferable than centralized one. Centralized system is much cheaper for vehicle owner and is good for receiving statistic data also, because all the data are stored in one database. Nevertheless, centralized system has a set of disadvantages, such as implementation difficulty and possible area of applying, big price for infrastructure owner, big amount of necessary components, increased time for reaction and increased risk at system failure. Distributed system is also the best solution for real time systems, because there is no need to transmit the data to the common center and backward, that is why distributed system was used for further research development.

Scheme of the distributed system structure is shown in the figure 1.3. Where:

SAT – satellite;
UEV – unmanned electrical vehicle;
UEVn – other unmanned electrical vehicles;
LC – level crossings;
RS – road signs;
TL – traffic lights;
GNSSR – GNSS receiver;
RFM – radio frequency module;
DS – distance sensor;
CAM – videocamera;

MDS – movement direction sensor;
VS – speed sensor;
MORA – moving object recognition algorithm;
SORA – static object recognition algorithm;
RRA – road recognition algorithm;
FL – fuzzy logic;
IM – immune memory;
AA – affinity algorithm;
ANN – traditional artificial neural network;
NN – neural network as a part of the INN;
INN – immune neural network;
TF – target function;
DM – decision making module;
DIS – driver informing system;
DCS – driver control system;
BS – electrical vehucke braking system;
ED – electrical vehicle electric drive;
STS – electrical vehicle steering system;
MECH – mechanical part;
IO – infrastructure objects;
CO – control objects;
SEN – sensors;
MIC – microcontroller;
ORM – object recognition module;
INFM – immune neuro-fuzzy module;
DR – electrical vehicle driver.



1.3 fig. Scheme of the distributed system structure with reduced number of components.

Different parts of the proposed system are used for different tasks:

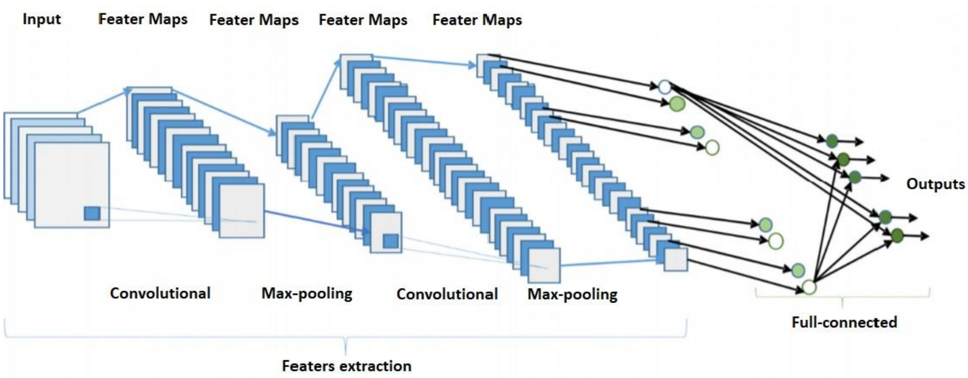
- ORM object recognition module, based on CNN convolutional neural network is used for the moving and static object recognition tasks as well as for road recognition task.
- INFM immune neuro fuzzy module, based on the novel INN immune neural network is used, for unsupervised collision probability evaluation and minimization task.
- ANN artificial neural network is used for evaluating and minimizing collision probability, along with FL fuzzy logic. In this research, ANN is included to compare its results with those of the proposed novel INN immune neural network. The objective is to draw conclusions on whether the novel network is better or worse than the traditional one.

Structure of the subsystems of the proposed methods are provided in 1.2.2., 1.2.3. and 1.2.4. chapters.

1.2.2. Subsystem structure for object recognition task

It is necessary to recognize the collision probability and to make a decision about its prevention to avoid the collision.

The proposed object recognition module ORM is based on the convolutional neural network CNN. System structure of the proposed CNN convolutional neural network is shown in the figure 1.4.



1.4 fig. Structure of the Convolutional neural network [61].

Figure 1.4 shows the overall architecture of CNNs consists of:

- feature extractors;
- classifier.

In the feature extraction layers, each layer of the network receives the output from its immediate previous layer as its input and passes its output as the input to the next layer. The CNN architecture consists of a combination of three types of layers:

- convolution;
- max-pooling;
- classification.

There are two types of layers in the low and middle-level of the network:

- Convolutional layers ($CONV_{CNN}$);
- max-pooling layers ($POOL_{CNN}$).

The even numbered layers are for convolutions and the odd numbered layers are for max-pooling operations. The output nodes of the $CONV_{CNN}$ and $POOL_{CNN}$ layers are grouped into a 2D plane called feature mapping. Each plane of a layer is usually derived from the combination of one or more planes of previous layers. The nodes of a plane are connected to a small region of each connected planes of the previous layer. Each node of the convolution layer extracts the features from the input images by convolution operations on the input nodes.

Higher-level features are derived from features propagated from lower level layers. The output of the last layer of the CNN is used as the input to a fully connected network which is called classification layer or Fully-connected layer (FC_{CNN}). Feed-forward neural networks have been used as the FC_{CNN} layer as they have better performance [62,63]. In the FC_{CNN} layer, the extracted features are taken as inputs with respect to the dimension of the weight matrix of the final neural network [61].

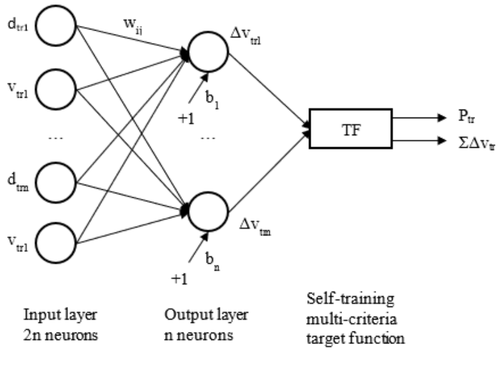
1.2.3. Subsystem structure for electrical vehicle collision probability evaluation and minimization task

In this research, neuro-fuzzy control proposed to be taken as intelligent control method. While fuzzy logic provides an inference mechanism under cognitive uncertainty in reactions to the transport situation danger level, computational neural networks offer exciting advantages, such as learning, adaptation, fault tolerance, parallelism and generalization.

The proposed ANN neural network structure is shown in the figure 1.5.

Each neural network input pair is made by two parameters:

- the distance of the vehicle till the trajectories crossing point d_{tr};
- average speed of the vehicle v_{tr}.



1.5 fig. Neural network structure.

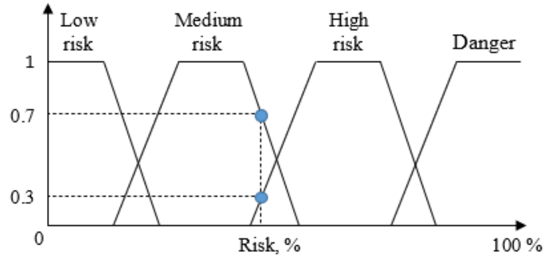
According to the number of vehicles, neural network has n outputs, and as a result the speed change for the vehicle is output.

Input neuron n of the output layer has wij weights and bj shifting.

Each j-th n of the ouput layer generates j-th vehicles speed change Δv_{tr}.

Values Δv_{tr1} ... Δv_{trn} are given to the self-training multi-criteria target function TF, that evaluate efficiency of the output according the collision probability P_{tr} and vehicles common speed change $\Sigma\Delta v_{tr}$.

General fuzzy logic structure, shown in the figure 1.6, consists of input as a risk level of recognized objects, membership functions for risk assessment, rule database for selection of actions and defuzzification functions for the level of the activity. In system identification, rule-based fuzzy models are usually applied. In these models, relations among variables are described by means of if–then rules with fuzzy predicates, such as “if the collision possibility is high, then the car speed must be changed” and “if the car speed was changed, but collision possibility still is high, then train speed must be changed”.

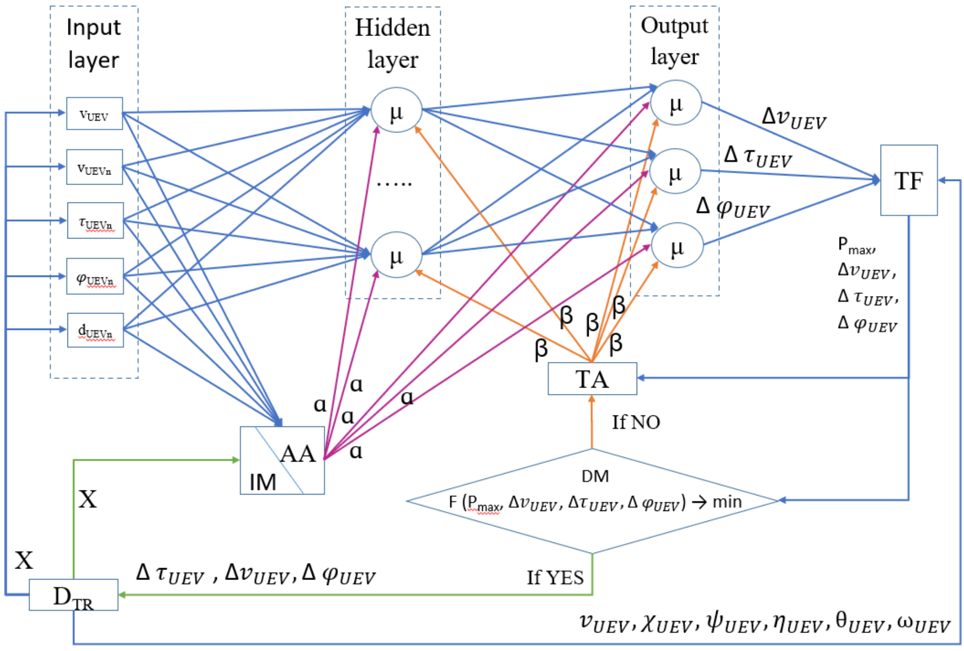


1.6 fig. Partitioning of the risk domain into four fuzzy sets.

Fuzzy sets are defined through their membership functions which map the elements of the considered universe to the unit interval [0, 1]. The extreme values 1 and 0 denote complete membership and non-membership, respectively, while a degree between 0 and 1 means partial membership in the fuzzy set. A particular domain element can simultaneously belong to several sets (with different degrees of membership). In figure 1.6, for instance, 45% of risk belongs to the set of high risk with membership 0.3 and to the set of medium risk with membership 0.7. We can suppose which action must be chosen for the crash prevention after defining the level of the risk in percentages. This gradual transition from membership to non-membership facilitates a smooth outcome of the reasoning (deduction) with fuzzy if–then rules, in fact a kind of interpolation.

1.2.4. Novel immune neural network based technology of machine learning for unsupervised safe vehicle control

Novel INN based technology can be used in distributed systems. It obtains data, makes calculations and provides necessary solutions how to avoid collision in the context of one particular UEV. It does not provide solutions for other participants.



1.7 fig. Immune neural network system structure of UEV.

Input data (X)

- Data:
  - v_{UEV} – speed of the own electrical vehicle;
  - v_{UEVn} – speed of other electrical vehicles;
  - τ_{UEVn} – horizontal movement direction of the other electric vehicle relative to one's own direction;
  - φ_{UEVn} – vertical movement direction of the other electric vehicle relative to one's own direction;
  - d_{UEVn} – distance till the possible crossing point with the other electrical vehicle.
- Number of parameters in the input data (X) depends on the situation - the number of other UEVs in the control area of own UEV. There is 1 input parameter for own UEV: speed. There are 4 input parameters for other UEVs: speed, horizontal movement direction in relation to the own UEV, vertical movement direction in relation to the own UEV and distance till possible crossing point.
- Data is received from the UEV embedded electronic device and is sent to the input layer of the immune neural network INN.

Input layer

- Input layer of the INN receives input data (X).
- Each UEV takes into account only those UEVs, which are in his control area, in case to minimize number of necessary calculations.

- Input data (X) is ordered for a more accurate recognition of the situation. The goal is to order multiple UEVs in relation to the own UEV to better understand their position and relative movement. Three parameters are used for ordering the UEVs: the horizontal movement direction τ_{UEVn}, the vertical movement direction φ_{UEVn}, and the distance till the crossing point d_{UEVn}. Ordering of other UEVs is done according to the slope τ_{UEVn} to these objects, starting from 0°, clockwise. If multiple UEVs have the same value of τ_{UEVn}, then ordering of these UEVs is done according to the slope φ_{UEVn} to these objects, starting from 0°, clockwise. If multiple UEVs have the same value of φ_{UEVn}, then ordering of these UEVs is done according to the distance to the crossing point with these UEV d_{UEVn}. This method helps to describe the situation accurately.
- Input data (X) are sent from the input layer to the affinity algorithm AA and hidden layer.

Affinity algorithm (AA)

- Affinity algorithm AA checks all the similar situations, stored in the immune memory IM and calculates the discrepancies $\mathcal{E}$. Situation with a smallest discrepancy $\mathcal{E}$ is chosen and its number α is sent to all μ neurons of the INN. If there is no similar situation stored in IM immune memory, then situation number $\alpha = 0$ is sent to the μ neurons.

Immune memory (IM)

- Database, that contains input data about previous situations that were solved. Each situation has its' number α.
- All the data in IM is stored in clusters for easier and faster match finding processes. For example, if three vehicles are participating in the possible collision situation, there is no need to find a similar situation in the group of situations with two participants. Therefore, the method of clustering is used for data storage in immune memory IM and faster affinity algorithm AA work.

Hidden layer

- The hidden layer consists of specialized μ neurons.
- Input data of each μ neuron of the hidden layer is:
    - input data (X);
    - situation number α received from the affinity algorithm AA;
    - signal β that indicates the need to recalculate the weights of μ neurons and is received from the training algorithm TA.
- In the μ neuron number of situation α is stored together with set of weights W_μ, that were used while solving the exact problem i.e. processing the similar input data. After number of the situation α is received, weights W_μ are chosen and training can be started. If there is no similar situation and $\alpha = 0$, then $W_\mu = 0$.

Output layer

- Output layer consists of specialized μ neurons.
- Input data of each μ neuron of the output layer is:
  - output data of the μ neurons of the hidden layer;
  - situation number α received from the affinity algorithm AA;
  - signal β that indicates the need to recalculate the weights of μ neurons and is received from the training algorithm TA.
- In the μ neuron number of situation α is stored together with set of weights W_{μ}, that were used while solving the exact problem, similar as in the μ neuron of the hidden layer. After number of the situation α is received, weights W_{μ} are chosen and training can be started. If there is no similar situation and α = 0, then $W_{\mu} = 0$.
- Output data of the output layer:
  - necessary horizontal movement direction change of the own UEV $\Delta\tau_{UEV}$;
  - necessary vertical movement direction change of the own UEV $\Delta\varphi_{UEV}$;
  - necessary speed change of the own UEV Δv_{UEV}.

Target function (TF)

- Input data of target function TF:
  - necessary horizontal movement direction change of the own UEV $\Delta\tau_{UEV}$;
  - necessary vertical movement direction change of the own UEV $\Delta\varphi_{UEV}$;
  - necessary speed change of the own UEV Δv_{UEV};
  - input data obtained directly from UEV embedded electronic device
    - Data:
      - v_{UEV} – actual speed of the UEV;
      - χ_{UEV} – latitude of the UEV actual position;
      - ψ_{UEV} – longitude of the UEV actual position;
      - η_{UEV} – altitude of the UEV actual position;
      - θ_{UEV} – actual horizontal movement direction of the UEV;
      - ω_{UEV} – actual vertical movement direction of the UEV.
    - In current research, the location of the crossing points is a variable value, that makes the solution more complicated, as the found solution $< \Delta v_{UEV},\ \Delta\tau_{UEV}, \Delta\varphi_{UEV} >$ has an influence on the distance to the crossing point. Thus, the evaluation of the target function TF requires the additional inputs $< v_{UEV}, \chi_{UEV}, \psi_{UEV}, \eta_{UEV}, \theta_{UEV}, \omega_{UEV} >$ obtained directly from the UEV to re-calculate the crossing point, distance and time to it.
- Target function TF calculates the collision probability P_{max}.
- Output data of target function TF:
  - collision probability P_{max};
  - necessary horizontal movement direction change $\Delta\tau_{UEV}$;

- necessary vertical movement direction change $\Delta\varphi_{UEV}$;
- necessary speed change Δv_{UEV}.

<u>Decision module (DM)</u>

- Input data of decision module DM:
  - collision probability P_{max}, received from the target function TF;
  - necessary horizontal movement direction change $\Delta\tau_{UEV}$, received from the target function TF;
  - necessary vertical movement direction change $\Delta\varphi_{UEV}$, received from the target function TF;
  - necessary speed change Δv_{UEV}, received from the target function TF.
- Decision module (DM) evaluates the found solution.
  - If collision probability P_{max} is greater than acceptable (safe) collision probability P_{safe} and
    - If number of training iterations t is less than maximal possible number of iterations T_{max}, that means the solution is not found yet and training must be repeated. Decision making module (DM) sends signal to the training algorithm (TA).
    - If number of training iterations t is bigger or equal than maximal possible number of iterations T_{max}, that means that situation can not be solved in the defined time, so speed reduction is done. Decision making module (DM) sends signal to the UEV embedded electronic device to stop the UEV $v_{UEV} = 0$.
  - If collision probability P_{max} is less or equal than acceptable (safe) collision probability P_{safe}, then:
    - found solution $< \Delta v_{UEV}, \Delta\tau_{UEV}, \Delta\varphi_{UEV} >$ is sent to the UEV embedded electronic device;
    - match error ε_a between the current situation and situation chosen from the immune memory IM in the beginning of training is calculated
      - if match error ε_a is bigger than maximal possible match error ε_{lim}, responsible for creation new record in the immune memory IM or replacing the existing, then:
        - IM saves the situation as a new record;
        - each μ neuron of the hidden and output layers saves set of weights W_μ that was used for solving this situation together with a number of this situation α.
      - if match error ε_a is less or equal than maximal possible match error ε_{lim}, then
        - the record of the situation α is updated in the immune memory IM;

- values of weights W_{μ} of μ neurons of the hidden and output layers are updated according to the last used.

<u>Training algorithm (TA)</u>

- Input data of the training algorithm (TA):
  - collision probability P_{max}, received from the target function TF;
  - necessary horizontal movement direction change $\Delta\tau_{UEV}$, received from the target function TF;
  - necessary vertical movement direction change $\Delta\varphi_{UEV}$, received from the target function TF;
  - necessary speed change Δv_{UEV}, received from the target function TF;
  - signal to repeat training, received from the decision making module DM.
- Training algorithm is used instead of traditional backpropagation algorithm. Backpropagation is typically used in supervised learning where the network is trained using labeled data, but the proposed novel INN is based on unsupervised learning.
- Training algorithm TA stores the last value of the P_{max}, which was received while solving this situation and compares this value to the new one. TA sends a signal β to all μ neurons, that means that training must be repeated. Signal β differs according to the result of the P_{max} comparison.
  - If it is the first training iteration, TA does not have information about previous P_{max}, so TA sends a signal β_1 to all μ neurons of the hidden and output layer. Signal β_1 means that the found solution does not solve the situation and training must continue.
  - The same happens if the result of the found solution is better or equal than previous $P_{max2} \leq P_{max1}$. TA sends a signal β_1 to all μ neurons of the hidden and output layer, that means the found solution is not worse than previous one and training must continue.
  - if the result of the found solution is worse than previous $P_{max2} > P_{max1}$, then TA sends a signal β_2 to all μ neurons of the hidden and output layer. Signal β_2 means that the found solution does not solve the situation and the result of last iteration is worse than the result of the previous one. The values of the weights must be returned to the previous before training continues.
  - <u>Training of μ neurons:</u>
    - When receiving β_1 new values of weights $W_{\mu j}$ are randomly chosen from the range $(W_{\mu j} - z \leq W_{\mu j+1} \leq W_{\mu j} + z)$, where z is predefined range parameter (may be adjustable).
    - When receiving β_2 $n$ew values of weights $W_{\mu j}$ rollback to the previous values $W_{\mu j-1}$ and then are randomly chosen from the range $(W_{\mu j-1} - z \leq W_{\mu j} \leq W_{\mu j-1} + z)$, where z is predefined range parameter (may be adjustable).

1.3. Chapter 1 summary

First chapter of doctoral thesis was devoted to the system model analysis and novel system structure developing:

1. Centralized, decentralized and distributed system models were compared in the first chapter of the doctoral thesis. The results of comparison show that distributed system is preferable because it is easier to implement and has less components, it is cheaper for infrastructure owner and it is not connected to the specific area, it has also decreased time for reaction and decreased risk at system failure. That is why a distributed system was chosen for this research.
2. Novel common system structure was developed. All the functions are performed by the microcontroller, embedded in each electrical vehicle, where the object recognition process and risk assessment are done, as well as opportunity assessment and decision making about necessary movement parameters change are done.
3. The proposed system structure is based on the:
   a. CNN based subsystem for object recognition task;
   b. NN based subsystem for electrical vehicle collision probability evaluation and minimization task;
   c. INN based technology of machine learning for unsupervised safe vehicle control.
4. CNN based subsystem structure for object recognition task was developed and described.
5. NN based subsystem structure for electrical vehicle collision probability evaluation and minimization task was developed and described.
6. Novel INN based technology of machine learning for unsupervised safe vehicle control was developed and described.

Mathematical models for the proposed system structure were developed and described in the second chapter of doctoral thesis.

2. DEVELOPED MATHEMATICAL MODELS

2.1. Situation description

Mathematical models were developed and proposed in the second chapter of the research, to prove the workability of the provided in the first chapter system structure and novel INN based technology of machine learning for unsupervised safe vehicle control.

Mathematical models are devided according to the performed task: objects and signals recognition, collision probability evaluation and location of the possible crossing point calculation, collision probability minimization.

2.2. Abbreviations

Abbreviations are added by the author.

2.1 table

List of abbreviations

Abbreviation	**Meaning**
AA	Affinity algorithm
a_{INN}	Number of the situation
B	Shifts, used in the neural network
B	Blue color pixels code of the RGB picture;
BS	Electrical vehicle braking system
C	Number of trajectories crossing point
CAM	Video camera for the video filming
CL_{CNN}	Number of output classes
CNN	Convolutional neural network
CO	Electrical vehicle control objects
$CONV_{CNN}$	Covolutional layer of the convolutional neural network
D	Common safety, i.t. collision prevention criterion
DATA	Information of each vehicle, which depends on the state of environment
D_{CNN}	Depth of the matrix before passing through the convolutional layer
DCS	Driver control system
DIS	Driver informing system
DM	Decision making module
DR	Driver of the electrical vehicle
DS	Distance sensor

Abbreviation	Meaning
D_{TR}	Devices for transport units obtain the position, calculate the motion parameters and communicate with other devices.
d_{tr}	Vehicles distance till the crossing point
E	Energy
ED	Electrical drive of the electrical vehicle
FC_{CNN}	Full-connected layer og the convolutional neural network
F_{CNN}	Convolutional neural network filter’s spatial extent
FL	Fuzzy logic
G	Green color pixels code of the RGB picture
GNSS	Global Navigation Satellite System
GNSSR	GNSS signal receiver
GPS	Global Positioning System for the location data obtaining
H_{CNN}	Height of the matrix before passing through the convolutional layer
HID_{CNN}	Hidden layer of the convolutional neural network
IM	Immune memory
INFM	Immune neuro-fuzzy module
INFN	Immune neuro-fuzzy network
INN	Immune neural network
IO	Infrastructure objects
ITR_{INN}	Maximal number of INN training iterations
K	Random number
K_{CNN}	Number of convolutional neural network filters
LC	Level crossing
M_{CNN}	Matrix, used in the convolutional neura; network
MDS	Movement direction sensor
MECH	Mechanical part of the electrical vehicle
MIC	Microcontroller for the data obtaining, calculations and decision making, according to the developed algorithm
MORA	Moving object recognition algorithm
N	Neuron
NN	Neural network
ORM	Object recognition module
P_{CNN}	The amount of zero padding
P_{col}	Point of the potential collision

Abbreviation	**Meaning**
$POOL_{CNN}$	Pooling layer of the convolutional neural network
P_{safe}	Maximal acceptable (safe) value of collision probability
P_{tr}	Collision probability of the electrical vehicle
PIC	Picture taken by video camera
PIC_H	Height of the picture
PIC_W	Weight of the picture
PIC_S	Sensitivity
R	Coefficient for speed reduction to avoid the collision if solution is not found
R	Red color pixels code of the RGB picture
RA	Risk assessment module
RFM	Radio frequency module
ROI	Region of interest
RRA	Road recognition algorithm
RS	Road signs
S	Safety distance
SAT	Satellite
S_{CNN}	Stride, used in convolutional neural network
SEN	Sensors
SORA	Static object recognition algorithm
STS	Electrical vehicle steering system
T	Number of training iterations
TA	Training algorithm
TF	Target function
TL	Traffic light
TP	Target point
TR	Training set
UEV	Unmanned electrical vehicle
VS	Speed sensor
v_{tr}	Speed of the electrical vehicle
W	Values of the weights, used in NN, CNN, INN or INFN
WCL_{CNN}	Number of weights with bias for the output layer of convolutional neural network
W_{CNN}	Weight of the matrix before passing through the convolutional layer

Abbreviation	Meaning
$WHID_{CNN}$	Number of weights with bias for the hidden layer of convolutional neural network
A	Closest match
β_{INN}	New values of the weights for the immune neural network
η_c	Altitude ηη of the crossing point
η_{tp}	Altitude ηη of the target point of electrical vehicle
η_{tr}	Altitude ηη of the elecrical vehicles position
θ_{tr}	Yaw angle of the flight of the electrical vehicle
μ_{INN}	Neurons with immune memory
τ_{tr}	Movement trajectory of the electrical vehicle
φ_{tr}	Pitch angle of the flight of the electrical vehicle
χ_c	Latitude χχ of the crossing point
χ_{tp}	Latitude χχ of the target point of electrical vehicle
χ_{tr}	Latitude χχ of the electrical vehicles position
ψ_c	Longitude ψψ of the crossing point
ψ_{tp}	Longitude ψψ of the target point of electrical vehicle
ψ_{tr}	Longitude ψψ of the electrical vehicles position
ε	Error
ε_{lim}	Maximal match error, responsible for creation new record in IM or replacing the existing

2.3. Mathematical sets of system objects

Electric transport control system is defined by following object classes:

$$UEVS = \{IO;\ SAT; UEV; DR\} \tag{2.1}$$

Where:

UEVS – unmanned electrical vehicle system structure;

IO – infrastructure objects;

SAT – satellite, component to obtain the coordinates of the electrical vehicle position in real time;

UEV – unmanned electrical vehicles;

DR – driver of the electrical vehicle.

$$IO = \{LC; RS; TL\} \tag{2.2}$$

Where:

LC – level crossings;

RS – road signs;
TL – traffic lights.

$$LC = \{LC_1; LC_2; \dots; LC_n\} \tag{2.3}$$
$$RS = \{RS_1; RS_2; \dots; RS_n\} \tag{2.4}$$
$$TL = \{TL_1; TL_2; \dots; TL_n\} \tag{2.5}$$
$$SAT = \{SAT_1; SAT_2; \dots; SAT_n\} \tag{2.6}$$
$$UEV = \{UEV_1; UEV_2; \dots; UEV_n\} \tag{2.7}$$
$$DR = \{DR_1; DR_2; \dots; DR_n\} \tag{2.8}$$
$$UEV = \{SEN; GNSSR; RFM; ORM; INFN; DIS; DCS; CO; MECH\} \tag{2.9}$$

Where:
SEN – sensors to obtain the input data;
GNSSR – *GNSS* signal receiver to obtain the $DATA_{GNSS}$;
RFM – radio frequency module to obtain the $DATA_{RFM}$;
ORM – object recognition module;
INFN – immune neuro-fuzzy module;
DIS – driver informing system;
DCS – driver control system;
CO – electrical vehicle control system;
MECH – mechanical part of the electrical vehicle.

$$SEN = \{DS; CAM; MDS; VS\} \tag{2.10}$$

Where,
DS – distance sensor to obtain the $DATA_{DS}$;
CAM – videocamera to obtain the $DATA_{CAM}$;
MDS – movement direction sensor to obtain the $DATA_{MDS}$;
VS – speed sensor to obtain the $DATA_{VS}$.

$$DATA_{GNSS} = \{\eta_{tr}; \chi_{tr}; \psi_{tr}\} \tag{2.11}$$

Where,
η_{tr} – altitude ηη of the electrical vehicles position;
χ_{tr} - latitude χχ of the electrical vehicles position;
ψ_{tr} – longitude ψψ of the electrical vehicles position.

$$DATA_{RFM} = \{\eta_{trn}; \chi_{trn}; \psi_{trn};\ \theta_{trn};\ \omega_{trn};\ v_{trn}\} \tag{2.12}$$

Where,
η_{trn} – altitude ηη of the other electrical vehicles position;
χ_{trn} – latitude χχ of the other electrical vehicles position;

ψ_{trn} – longitude ψψ of the other electrical vehicles position;
θ_{trn} – yaw angle of the flight of the other electrical vehicle;
ω_{trn} – pitch angle of the flight of the other electrical vehicle;
v_{trn} – other electrical vehicles speed.

$$DATA_{DS} = \{d_{tr}\} \tag{2.13}$$

Where,
d_{tr} – distance till the object.

$$DATA_{CAM} = \{RGB;\ XY\} \tag{2.14}$$

Where,
RGB – red, green, blue pixels code;
XY – position of the pixel.

$$DATA_{MDS} = \{\theta_{tr};\ \omega_{tr}\} \tag{2.15}$$

Where,
θ_{tr} – yaw angle of the flight of the electrical vehicle;
ω_{tr} – pitch angle of the flight of the electrical vehicle.

$$DATA_{VS} = \{v_{tr}\} \tag{2.16}$$

Where,
v_{tr} – electrical vehicles speed.

$$ORM = \{MORA; SORA; RRA\} \tag{2.17}$$

Where,
MORA – moving object recognition algorithm;
SORA – static object recognition algorithm;
RRA – road recognition algorithm.

$$MORA = \{DATA_{DS};\ DATA_{CAM}; CNN\} \tag{2.18}$$

Where,
$DATA_{DS}$ – data, taken by distance sensor;
$DATA_{CAM}$ – data, taken by video camera;
CNN – convolutional neural network.

$$SORA = \{DATA_{DS};\ DATA_{CAM}; CNN\} \tag{2.19}$$

Where,
$DATA_{DS}$ – data, taken by distance sensor;
$DATA_{CAM}$ – data, taken by video camera;

CNN – convolutional neural network.

$$RRA = \{DATA_{CAM}; CNN\} \quad (2.20)$$

Where,

$DATA_{CAM}$ – data, taken by video camera;

CNN – convolutional neural network.

$$CNN = \{CONV_{CNN};\ POOL_{CNN};\ FC_{CNN}\} \quad (2.21)$$

Where,

$CONV_{CNN}$ – convolutional layer of the convolutional neural network;

$POOL_{CNN}$ – pooling layer of the convolutional neural network;

FC_{CNN} – fully-connected layer of the convolutional neural network.

$$CONV_{CNN} = \{K_{CNN}; F_{CNN}; S_{CNN}; P_{CNN}\} \quad (2.22)$$

Where,

K_{CNN} – number of convolutional neural network filters;

F_{CNN} – convolutional neural network filter's spatial extent;

S_{CNN} – stride;

P_{CNN} – the amount of zero padding.

$$POOL_{CNN} = \{F_{CNN}; S_{CNN}\} \quad (2.23)$$

Where,

F_{CNN} – convolutional neural network filter's spatial extent;

S_{CNN} – stride.

$$FC_{CNN} = \{HID_{CNN}; CL_{CNN}\} \quad (2.24)$$

Where,

HID_{CNN} – hidden layer of the convolutional neural network;

CL_{CNN} – number of output classes.

$$INFN = \{FL; IM; AA; NN; TF; DM\} \quad (2.25)$$

Where,

FL – fuzzy logic;

IM – immune memory;

AA – affinity algorithm;

NN – neural network;

TF – target function;

DM – decision making algorithm.

$$CO = \{BS; ED; STS\} \quad (2.26)$$

Where,

BS – braking system;

ED – electrical drive;

STS – steering system.

The class interaction diagram is shown in figure 1.3. Videocamera CAM of the electrical vehicle receives the data from the level crossings LC, road signs and markings RS and traffic lights TL. Also, distance DS, movement direction MDS and speed VS sensors obtain the relevant data. The obtained from videocamera CAM and speed sensor VS data are sent to the object recognition module, which contains moving object recognition algorithm MORA, static object recognition algorithm SORA and road surface recognition algorithm RRA. These algorithms are based on convolutional neural network CNN and color recognition algorithms and help to recognize the type of objects, their size, speed and movement direction. GNSS receiver receives the data about own vehicles location. Data about other electrical vehicles v_{tr} speed, movement direction and location are obtained by using radio frequency module RFM. Data obtained by GNSS receiver, radio frequency module RFM, movement direction sensor MDS and speed sensor VS as well as output data of the object recognition module are sent to the fuzzy logic block FL of the immune neuro-fuzzy module. Fuzzy logic FL is used for risk assessment or danger level recognition. If the level of the danger is bigger or equal than specified, then the data is sent to the affinity algorithm AA. Affinity algorithm AA checks all the similar situations, stored in the database in the immune memory IM and calculates the discrepancies. The situation with the smallest discrepancy is chosen and its weight w are used in the neural network ANN. Neural network ANN calculates necessary movement parameters change that could help to reduce collision probability. After necessary changes of the movement parameters are calculated, they are evaluated by using the target function TF. If the collision probability is still bigger than specified, then new values of the weights are chosen and new values of movement paramaters changes are calculated by neural network ANN. Otherwise, if collision probability is less or equal than specified, then decision making module DM sends the necessary solution to the driver informing system DIS, to the immune memory IM of the immune neuro-fuzzy module to save the input parameters and weights values, used for the solution, and to the radio frequency module RFM with a view to inform other electrical vehicles UEVn about movement parameters change.

Driver receives the information about necessary speed and trajectory of motion change on the information display and controls the electrical vehicle mechanical part MECH by controlling the braking system BS, steering system STS and electric drive ED. Speed sensor VS obtains speed changes and sends this data to the decision making module DM. The data from movement direction sensor MDS also is sent to the decision making module DM. If decision making module DM doesn’t receive the information about speed or trajectory of motion change, then the signal to driver control system is sent and electrical vehicle mechanical part MECH is controlled without driver intervention. This may help to control the electrical vehicle when the driver is not reacting in time because he for example fell asleep or passed out. Also, this can help to exclude the driver of the proposed system, after the system will be trained and immune memory IM will be full enough of good examples. Excluding the driver will make the electrical vehicle autonomous.

2.4. Mathematical model for traffic light red signal recognition task

System structure is shown in the figure 1.3. The elements of the developed system interact as follows:

At the very begging camera connection is done. Then setting parameters of the PIC picture is done:

$$PIC = \{PIC_H, PIC_w, PIC_s\} \tag{2.27}$$

Where,

PIC_H – picture height;

PIC_w – picture weight;

PIC_s – sensitivity.

PIC picture “original” creating and PIC picture “original” output are done. Region of interests (ROI) detection in “original” PIC picture by detection the traffic light, by using the CNN Convolutional neural network, that was proposed in the previous chapter. ROI is copied into RGB red green blue colors picture:

$$ROI = \{R, G, B\} \tag{2.28}$$

Where,

R – red color pixels code of the RGB picture;

G – green color pixels code of the RGB picture;

B – blue color pixels code of the RGB picture.

Intervals for the red color pixels code were defined:

R1 <= R <= R2;

G1 <= G <= G2;

B1 <= R <= B2.

The color of RGB picture point is assumed to be red if point meets the requirements of red color pixels code.

Intervals for the green color pixels code were defined:

R3 <= R <= R4;

G3 <= G <= G4;

B3 <= R <= B4 .

The color of RGB picture point is assumed to be green if point meets the requirements of green color pixels code.

Points that meet the requirments of red and green color pixels code are saved into the variables:

$$r\_range = \{R_1, R_2, \dots, R_n\} \quad (2.29)$$

$$g\_range = \{G_1, G_2, \dots, G_n\} \quad (2.30)$$

$$b\_range = \{B_1, B_2, \dots, B_n\} \quad (2.31)$$

r_range, g_range, b_range channels alteration and saving into the rgb_and variable:

$$rgb\_and = \{r\_range, g\_range, b\_range\} \quad (2.32)$$

Counting the sum of points in the rgb_and picture: number of points / 255 and saving the result into massive parry.

Setting the condition:

If the number of frames is less than specified, then sum up all the sums of points of all frames, else sum up the sums of points of the last 10 frames.

If the sum of points > 10, then the traffic light shows red color, else if the sum of points > 20, then the traffic light shows green color.

Mathematical model for traffic light red signal recognition task was used:

- To develop algorithm for traffic light red signal recognition method for electric transport dangerous situation recognition and prevention task (Chapter 3.2).
- To develop computer model for testing the algorithm of traffic light red signal recognition method for electric transport dangerous situation recognition and prevention task (Chapter 4.2).
- To make experimental testing of the proposed algorithm of traffic light red signal recognition method for electric transport dangerous situation recognition and prevention task (Chapter 5.2).

2.5. Mathematical model for object recognition task

The proposed system structure works as follows: CAM videocamera receives the data about other objects (LC level crossings, RS road signs, TL traffic lights, UEVn other electrical vehicles) by taking pictures. After these data are sent to the ORM object recognition module, based on the CNN convolutional neural network, where depending on the object type, object is recognized bu MORA moving object regognition algorithm, SORA static object recognition algorithm or RRA road recognition algorithm. After object is recognized, a risk assessment and decision about necessary movement paramateres change is made by the INFM immune neuro-fuzzy module. The decision about necessary movement parameters change is sent to the DIS driver informing system by DM decision making algorithm. DIS driver information system consists of information display, so the the DR driver receives the information about necessary movement parameters change and controls the MECH mechanical part according to the received advice. If the DCS driver control system receives the information, that movement parameters were not changed and DR driver does not react in due time, then DCS driver control system intervenes in electrical vehicle control and activates BS braking system.

Convolutional layer $CONV_{CNN}$ computes the output of neurons n that are connected to local regions in the input. Each neuron n computing a dot product between their weights W and a small region they are connected to in the input volume.

The convolutional layer $CONV_{CNN}$ parameters consist of a set of learnable filters F_{CNN}. This layer requires four hyper-parameters:

K_{CNN} - number of filters;

F_{CNN} – filter's spatial extent;

S_{CNN} - the stride;

P_{CNN} - the amount of zero padding.

Input of this layer is a 3D matrix $M_{CNN}1$ with pixel values and $W_{CNN}1 \times H_{CNN}1 \times D_{CNN}1$ size.

$W_{CNN}1$ – weight of the matrix before passing through the CONV layer;

$H_{CNN}1$ – height of the matrix before passing through the CONV layer;

$D_{CNN}1$ – depth of the matrix before passing through the CONV layer.

For RGB picture depth of the matrix $D_{CNN}1=3$.

After matrix $M_{CNN}1$ passes through the convolutional layer $CONV_{CNN}$, the new matrix $M_{CNN}2$ with pixel values and $W_{CNN}2 \times H_{CNN}2 \times D_{CNN}2$ size is generated, where:

$$W_{CNN}2 = (W_{CNN}1 - F_{CNN} + 2P_{CNN})/S_{CNN} + 1 \qquad (2.33)$$

$$H_{CNN}2 = (H_{CNN}1 - F_{CNN} + 2P_{CNN})/S_{CNN} + 1 \qquad (2.34)$$

$$D_{CNN}2 = K_{CNN} \qquad (2.35)$$

With parameter sharing, it introduces $F_{CNN} \cdot F_{CNN} \cdot D_{CNN}1$ weights W per filter, for a total of $(F_{CNN} \cdot F_{CNN} \cdot D_{CNN}1) \cdot K_{CNN}$ weights W and K_{CNN} biases.

Functions of the pool layer $POOL_{CNN}$ are: to reduce progressively the spatial size of the representation; to reduce the amount of parameters and computation in the network; and hence to also control overfitting. This layer requires two hyper-parameters:

F_{CNN} – filter's spatial extent;

S_{CNN} - the stride.

Input of this layer is a 3D matrix $M_{CNN}1$ with pixel values. After matrix $M_{CNN}1$ passes through the $POOL_{CNN}$layer the new matrix $M_{CNN}2$ with pixel values and $W_{CNN}2 \times H_{CNN}2 \times D_{CNN}2$ size is generated, where:

$$W_{CNN}2 = (W_{CNN}1 - F_{CNN})/S_{CNN} + 1 \qquad (2.36)$$

$$H_{CNN}2 = (H_{CNN}1 - F_{CNN})/S_{CNN} + 1 \qquad (2.37)$$

$$D_{CNN}2 = D_{CNN}1 \qquad (2.38)$$

Fully connected layer FC_{CNN} computes the class scores. Each neuron n in this layer will be connected to all the numbers in the previous volume as in ordinary ANN. This layer requires two hyper-parameters:

HID_{CNN} - Number of hidden layer neurons;

CL_{CNN} - Number of output classes.

Input of this layer is a 3D matrix M_{CNN} with pixel values and $W_{CNN}\times H_{CNN}\times D_{CNN}$ size. Layer produces a matrix of size $1\times CL_{CNN}$ where:

$$CL_{CNN} = (c_{CNN1}, \ldots, c_{CNNCL}) \quad (2.39)$$

$$c_{CNN1} = \{0, 1\} \quad (2.40)$$

$$\sum_{i=1}^{CL} c_{CNNi} = 1 \quad (2.41)$$

Number of weights W with bias for the hidden layer HID_{CNN}:

$$WHID_{CNN} = (W_{CNN}\times H_{CNN}\times D_{CNN}+1)\times HID_{CNN} \quad (2.42)$$

Number of weights W with bias for the output layer:

$$WCL_{CNN}=(HID_{CNN}+1)\times CL_{CNN} \quad (2.43)$$

CNN consists of the sequence of Convolutional layer $CONV_{CNN}$ and Pooling layer $POOL_{CNN}$ with Full-connected subnetwork FC_{CNN} at the end. General structure of the CNN:

INPUT $[W_{CNN}0\times H_{CNN}0\times D_{CNN}0] \rightarrow$

$\rightarrow$ $CONV_{CNN}1[K_{CNN}1,F_{CNN}1,P_{CNN}1,S_{CNN}1]$ =

$= OUT_{CNN}[W_{CNN}11=(W_{CNN}0-F_{CNN}+2P_{CNN})/S_{CNN}+1\times H_{CNN}11=$

$= (H_{CNN}0-F_{CNN}+2P_{CNN})/S_{CNN}+ 1H_{CNN}11\times K_{CNN}1>D_{CNN}0, K_{CNN}1/D_{CNN}0 = int]$

$\rightarrow$

$\rightarrow$ $POOL_{CNN}1[F_{CNN}P_{CNN}1, S_{CNN}P_{CNN}1]$ =

=

$OUT_{CNN}[W_{CNN}12=W_{CNN}11/F_{CNN}P_{CNN}1\times H_{CNN}12=H_{CNN}11/F_{CNN}P_{CNN}1\times K_{CNN}1] \rightarrow$ (2.44)

$\rightarrow$ …

$\rightarrow$ $CONV_{CNN}n[K_{CNN}n, F_{CNN}n_{CCN}n, P_{CNN}n, S_{CNN}nC_{CNN}n]$ = $OUT_{CNN}[W_{CNN}n1\times H_{CNN}n1\times K_{CNN}n1] \rightarrow$

$\rightarrow$ $POOL_{CNN}n[F_{CNN}P_{CNN}n, S_{CNN}P_{CNN}n]$ =

=

$OUT_{CNN}[W_{CNN}n2=W_{CNN}2/F_{CNN}P_{CNN}n\times H_{CNN}n2=H_{CNN}2/F_{CNN}P_{CNN}n\times K_{CNN}n1] \rightarrow$

$\rightarrow$ $FC_{CNN}[HID_{CNN},CL_{CNN}] = OUT_{CNN}[1\times CL_{CNN}]$

The bigger number of layers CNN has, the more exact result can be gotten.

Mathematical model for object recognition task was used:

- To develop algorithm for convolutional neural network for object recognition for electric transport dangerous situation recognition and prevention task (Chapter 3.3).
- To develop computer model for testing the algorithm of convolutional neural network for object recognition for electric transport dangerous situation recognition and prevention task (Chapter 4.3).
- To make experimental testing of the proposed algorithm of convolutional neural network for object recognition for electric transport dangerous situation recognition and prevention task (Chapter 5.3).

2.6. Mathematical model for possible crossing point detection and collision probability evaluation task

As input data such parameters of each vehicle are used:

LFn - coordinates of the left front angle of the vehicle;

RFn - coordinates of the right front angle of the vehicle;

LRn - coordinates of the left rear angle of the vehicle;

RRn - coordinates of the right rear angle of the vehicle;

Vn - speed of the vehicle;

Tn - trajectory of the motion of the vehicle;

Ln – length of the vehicle.

Model finds out either the object is in the control area or not, after all the input data is selected. We propose that object 2 is in the control area of object 1 only if object 2 is in front of the object 1 or on the same level.

If object 2 is in the control area of object 1 then calculations are done, otherwise no calculations are needed. As coordinates of left and right angles of vehicles are known, formulas of the straight lines of the left and right sides are calculated and crossing points of these lines are detected.

The next step is to detect the minimal and maximal distances till the possible crossing point. The minimal distance till the crossing point is calculated as:

For object 1: Coordinates of the crossing point (R1; L2) minus coordinates of the right front angle of the vehicle 1 minus half of the length of the vehicle 1:

$$Dist1min = (R1; L2) - (RF1) - (L1/2) \tag{2.45}$$

For object 2: Coordinates of the crossing point (R1; L2) minus coordinates of the left front angle of the vehicle 2 minus half of the length of the vehicle 2:

$$Dist2min = (R1; L2) - (LF2) - (L2/2) \tag{2.46}$$

The maximal distance till the crossing point is calculated as:

For object 1: Coordinates of the crossing point (L1; R2) minus coordinates of the left front angle of the vehicle 1 plus half of the length of the vehicle 1:

$$Dist1max = (L1; R2) - (LF1) + (L1/2) \tag{2.47}$$

For object 2: Coordinates of the crossing point (L1; R2) minus coordinates of the right front angle of the vehicle 2 plus half of the length of the vehicle 2:

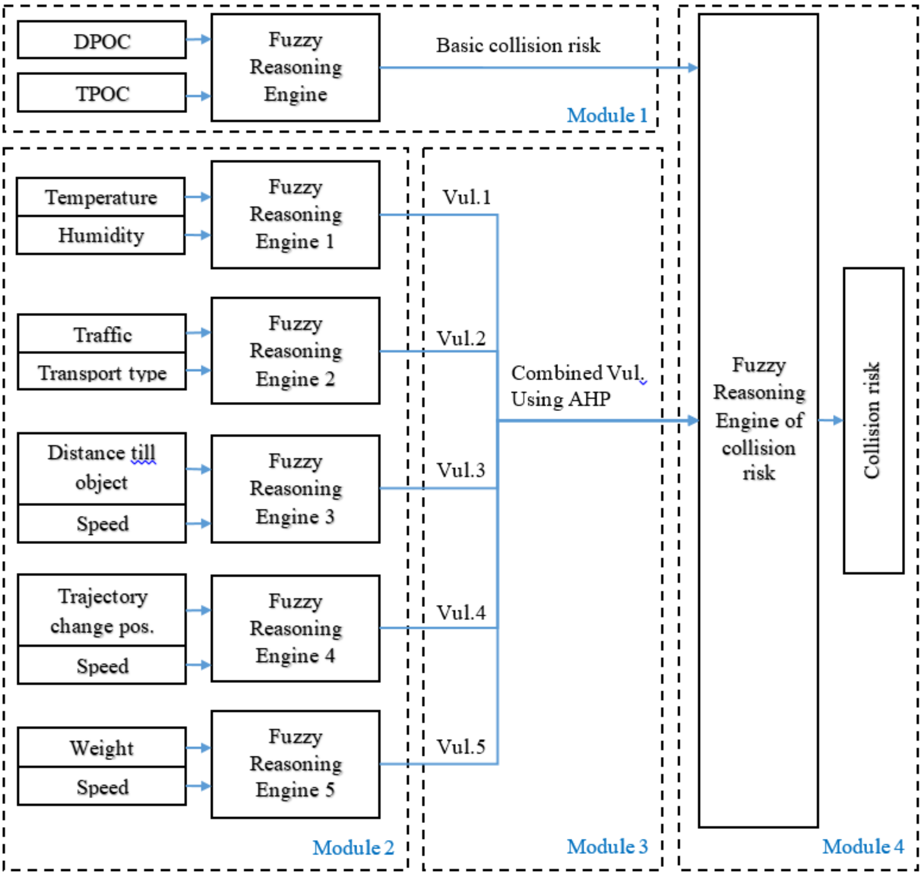
$$Dist2max = (L1; R2) - (RF2) + (L2/2) \tag{2.48}$$

After distances are calculated, minimal and maximal time till the crossing point is calculated for both objects:

$$Timenmin = Distnmin \,/\, Vn \tag{2.49}$$

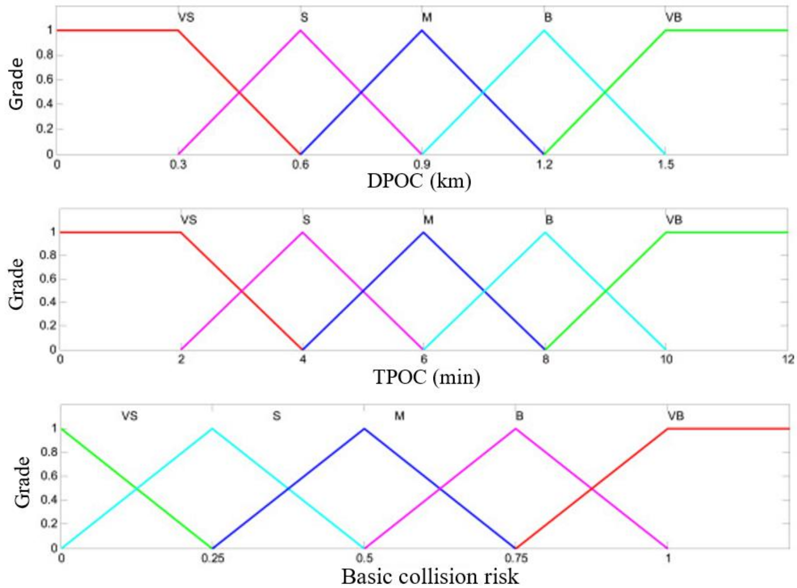
$$Timenmax = Distnmax \,/\, Vn \tag{2.50}$$

The algorithm of the electrical vehicle collision risk solving system has been proposed based on basic collision risk and vulnerabilities of accidents [64].



2.1 fig. Structure of the collision risk solving system.

Vulnerability can increase the possibility of collision accidents. The factors of vulnerabilities including distance and time till the crossing point with other object, bad weather conditions, that can influency visibility and possible speed or trajectory change, traffic congestion and type of vehicle are involved in the fuzzy reasoning engines to evaluate the navigational conditions and environment. Fuzzy logic is employed to reason basic collision risk using distance to the possible point of collision (DPOC) and time till the possible point of collision (TPOC). Analytical Hierarchy Process (AHP) method is used to obtain the integration of vulnerabilities [64]. In this research, vulnerability factors have been proposed to improve the collision risk assessment especially in situations when the connection between vehicles or GNSS signal are lost and vehicles need to rely on camera and sensors data.



2.2 fig. Membership functions for DPOC, TPOC and basic collision risk.

Structure of the collision risk solving system is shown in the figure 2.1. It consists of four modules:

a. **Module 1**

   In module 1, the basic collision risk is solved by using DPOC and TPOC, which are significant input variables that can determine the possibility of an electrical vehicle collision. Basic collision risk is obtained based on the membership functions and rules of DPOC and TPOC (2.2 fig.).

   (DPOC, TPOC) -> Basic collision risk

   A fuzzy reasoning model is used and the membership functions for DPOC, TPOC and basic collision risk are classified as five linguistic values. The calculation of basic collision risk contains two inputs and one output, which is determined by the reasoning rules.

2.2 table

Reasoning rules of DPOC, TPOC and basic collision risk [64]

DPOC	**TPOC**				
	very low	**Low**	**medium**	**high**	**very high**
very low	VH	VH	H	H	M
low	VH	H	H	M	M
medium	H	H	M	M	L
high	H	M	M	L	L
very high	M	M	L	L	VL

The reasoning rules are listed for the fuzzy logic reference engine in table 2.2, where:

VH - very high,
H - high,
M - medium,
L – low,
VL - very low [64].

The inputs of DPOC & TPOC and the output of basic collision risk have five linguistic variables. As a result, 25 fuzzy rules are developed to determine the basic collision risk. The bigger is DPOC and TPOC, the smaller is collision probability.

b. Module 2

In the module 1 distance and time till the possible point of collision are chosen as main variables for collision probability evaluation task, but other parameters also need to be analysed, to make the found result more accurate. To achive this goal, module 2 was developed.

Vulnerabilities proposed in the second module, can be changed, according to the type of the electrical vehicle. For example, there is no need to evaluate the possibility of trajectoty change or traffic jams for rail transport, but these parameters are very important for road electrical vehicles. General vulnerabilities were used in this research and as they are less important than DPOC and TPOC, only 3 linguistic variables are used, and 9 fuzzy rules are designed for each vulnerability.

- **Weather conditions**

  Weather conditions are the first vulnerability to be analysed (2.3 tab.).
  (HF, Temp) -> Vulnerability (1) of weather conditions.

2.3 table

Reasoning rules for the vulnerability of weather conditions for road vehicles

Humidity/ fallout	Temperature		
	Low	**Medium**	**High**
Low	M	L	VL
Medium	H	M	L
High	VH	H	M

The results differ according to the transport type. Reasoning rules for the vulnerability of weather conditions for road vehicles are shown in the 2.3 table. As humidity/fallout is getting bigger and temperature is getting smaller, the collision risk will get bigger.

Triangular membership functions of weather conditions are described:

- humidity/fallout (mm/12h):
  - "low"=(0 0 10),
  - "medium"=(0 10 50),
  - "high"=(10 50 50),
- Temperature(°C):
  - "low"=(-40 -40 0),
  - "medium"=(-40 0 40),
  - "high"=(0 40 40).

Consequence: (0 0 0.25), (0 0.25 0.5), (0.25 0.5 0.75), (0.5 0.75 1), (0.75 1 1). The values of 0/10/50 (mm/12h) and -40/0/40 (°C) can be taken to correspond the linguistic values of low, medium, high for humidity/fallout and temperature, respectively.

- **Traffic and restrictions**

Traffic congestion is the second vulnerability to be analysed (2.4 tab.).
(TC, Ttype) -> Vulnerability (2) of traffic and restrictions.

2.4 table

Reasoning rules for the vulnerability of traffic congestions

Traffic congestion level	Transport type		
	Road	**Aerial**	**Rail**
Low	M	M	VL
Medium	H	H	L
High	VH	VH	M

Three transport types were considered in traffic congestion analysis. As a result, 9 fuzzy rules are developed. The smaller the level of the traffic congestion is, the smaller is collision probability. The level of safety differs according to the transport type also. Traffic congestions are most dangerous for road vehicles, because of the number of participants, lack of knowledge of inexperienced drivers, as well as selfish and undisciplines behavior of some participants, and aerial vehicles, because of the speed and size of some aerial vehicles.

Triangular membership functions of traffic congestion level are described:

- Traffic congestion level (%):
  - “low”=(0 0 25),
  - “medium”=(0 25 100),
  - “high”=(25 100 100),
- Transport type:
  - “rail”=(0 0 1),
  - “aerial”=(0 1 2),
  - “road”=(1 2 2).

Consequence: (0 0 0.25), (0 0.25 0.5), (0.25 0.5 0.75), (0.5 0.75 1), (0.75 1 1). The values of 0/25/100 (%) and 0/1/2 (transport type) can be taken to correspond the linguistic values of low, medium, high for vulnerability of traffic congestions.

- **Other objects on the way**

The existence of objects on the way is the third vulnerability to be analysed (2.5 tab.). It can be static object, for example fallen tree, or moving objects, such as human beings, animal or other vehicle.

(Dist, V) -> Vulnerability (3) existence of objects on the way.

2.5 table

Reasoning rules for the vulnerability of other objects on the way

Speed	**Distance till other object**		
	Low	**Medium**	**High**
low	H	M	VL
medium	VH	M	L
high	VH	H	M

Collision probability depends on the existence of other objects on the way. The bigger is distance to the object and the smaller is speed of the electrical vehicle, the more safe is traffic.

Triangular membership functions of existence of objects on the way are described:

- speed (km/h):
  - “low”=(0 0 100),
  - “medium”=(0 100 500),
  - “high”=(100 500 500),
- Distance till other object (km):
  - “low”=(0 0 1),

- "medium"=(0 1 10),
- "high"=(1 10 10).

Consequence: (0 0 0.25), (0 0.25 0.5), (0.25 0.5 0.75), (0.5 0.75 1), (0.75 1 1). The values of 0/100/500 (km/h) and 0/1/10 (km) can be taken to correspond the linguistic values of low, medium, high for speed of the vehicle and distance till the object, respectively.

- **Maneuverability**

Maneuverability is the next vulnerability to be analysed (2.6 tab.).

The more possibility to maneuver has the electrical vehicle, the smaller is collision probability, because vehicle has more options.

(TrCh, V) -> Vulnerability (4) maneuverability.

The smaller is possibility of the trajectory change, the bigger is the risk of the collision. The high speed of vehicle increases the collsion probability also.

2.6 table

Reasoning rules for the vulnerability of maneuverability

Speed	**Trajectory change possibility**		
	Low	**Medium**	**High**
low	M	L	VL
medium	H	M	L
high	VH	H	H

Triangular membership functions of maneuverability are described:

- speed (km/h):
  - "low"=(0 0 100),
  - "medium"=(0 100 500),
  - "high"=(100 500 500),
- Trajectory change possibility (%):
  - "low"=(0 0 50),
  - "medium"=(0 50 100),
  - "high"=(50 100 100).

Consequence: (0 0 0.25), (0 0.25 0.5), (0.25 0.5 0.75), (0.5 0.75 1), (0.75 1 1). The values of 0/100/500 (km/h) and 0/50/100 (%) can be taken to correspond the linguistic values of low, medium, high for speed of the vehicle and possibility of trajectory change, respectively.

- **Possibility of the speed change**

The possibility of the speed change is next vulnerability to be analysed (2.7 tab). Speed and weight of electrical vehicle were considered, to analyse the possibility of the speed change.

(W, V) -> Vulnerability (5) possibility of the speed change.

2.7 table

Reasoning rules for the vulnerability of possibility of the speed change

Speed	**Weight**		
	Low	**Medium**	**High**
low	VL	L	M
medium	L	M	H
high	M	H	VH

The bigger is weight of the electrical vehicle, the more different is to stop the vehicle. For example, the braking distance of the train with 50 cargo wagons is up to 1 km long.

Triangular membership functions of possibility of the speed change are described:

- speed (km/h):
  - “low”=(0 0 100),
  - “medium”=(0 100 500),
  - “high”=(100 500 500),
- Speed change possibility (%):
  - “low”=(0 0 50),
  - “medium”=(0 50 100),
  - “high”=(50 100 100).

Consequence: (0 0 0.25), (0 0.25 0.5), (0.25 0.5 0.75), (0.5 0.75 1), (0.75 1 1). The values of 0/100/500 (km/h) and 0/50/100 (%) can be taken to correspond the linguistic values of low, medium, high for speed of the vehicle and possibility of speed change, respectively.

c. Module 3

The purpose of the module 3 is to assess the interacting vulnerabilities that significantly increase collision probability of the electrical vehicles. For this purpose, analytic hierarchy process (AHP) method was used. AHP method is a multipurpose decision-making method to solve decision making problems involving multiple goals [65,66]. AHP is applied to measure the weight of vulnerability factors under consideration. Preference of factors was analysed on the importance scale from 1 to 9 (2.8 tab.).

2.8 table

Saaty's Scale of Relative Importance [67]

Scale	**Numerical Rating**	**Reciprocal**
Extremely Preferred	9	1/9 ≈ 0.11
Very strongly to extremely	8	1/8 ≈ 0.13
Very strongly preferred	7	1/7 ≈ 0.14
Strongly to very strongly	6	1/6 ≈ 0.17

Scale	Numerical Rating	Reciprocal
Strongly preferred	5	1/5 = 0.2
Moderately to strongly	4	1/4 = 0.25
Moderately preferred	3	1/3 ≈ 0.33
Equally to moderately	2	½ = 0.5
Equally preferred	1	1/1 = 1

2.9 table

Importance matrix of vulnerability factors

	S1	S2	S3	S4	S5	Total scores	Weight	Rank
S1	1	7	0.2	0.5	0.11	8.81	0.16	4
S2	0.14	1	0.13	0.5	0.13	1.9	0.03	5
S3	5	8	1	0.5	0.5	15	0.27	2
S4	2	2	2	1	3	10	0.18	3
S5	9	8	2	0.33	1	20.33	0.36	1
					SUM	56.04	1	

Importance matrix of vulnerability factors is shown in the 2.9 table, where:
S1 – vulnerability "Weather condition";
S2 – vulnerability "Traffic congestion";
S3 – vulnerability "Other objects on the way";
S4 – vulnerability "Maneuverability";
S5 – vulnerability "Possibility of the speed change".

If row is more important then the colomn, then numerical rating is used, otherwise reciprocal is used.

d. Module 4

The risk will be calculated using two input variables as shown in 2.10 table in which the collision risk is considered as main factor, while vulnerability is of subordination. Fuzzy logic is also used to get the collision risk because it is not possible to obtain the fusion of collision risk and vulnerability in mathematical way.

2.10 table

Reasoning rules for collision risk

Collision risk	Combined vulnerability				
	Very low	**Low**	**Medium**	**High**	**Very high**
Very low	VL	VL	L	M	H
Low	L	L	M	H	VH
Medium	M	M	H	VH	VH
High	H	H	VH	VH	VH
Very high	VH	VH	VH	VH	VH

The membership functions of collision risk, combined vulnerability and consequence are designed as (0 0 0.25), (0 0.25 0.5), (0.25 0.5 0.75), (0.5 0.75 1), (0.75 1 1) and the rules of fuzzy reasoning in the table 2.10 are designed in the way of on author's thinking that the low value of combined vulnerability affects lightly the high value of basic collision risk.

Mathematical model for possible crossing point detection and collision probability evaluation task was used:

- To develop algorithm for electrical transport collision probability evaluation task (Chapter 3.4).
- To develop computer model for testing the algorithm of electrical transport collision probability evaluation task (Chapter 4.4).
- To make experimental testing of the proposed algorithm of electrical transport collision probability evaluation task (Chapter 5.4).

2.7. Mathematical model for the neural network

The mathematical model is represented with following sets:

$U \subset (U_1, \dots, U_n)$ - set of transport units as a subsets of different types, where for different transport safety task it could be:

$U^1 = (U_1^1, \dots, U_{n1}^1)$ - subset of railway transport units;

$U^2 = (U_1^2, \dots, U_{n2}^2)$- subset of road vehicles;

$U^3 = (U_1^3, \dots, U_{n3}^3)$- subset of aerial vehicles etc.

$P = (p_1, p_2, \dots, p_c)$ - set of infrastructure objects, where the collision of vehicles, e.g. for railway transport it could be level-crossings, switches, etc. For this research, the crossing section is assumed as a short straight segment of the route or trajectory.

The geographical coordinates of all crossing of possible routes or trajectories of transport units are defined by these sets:

$$\chi_b^p = \{\chi_b^{p_1}, \chi_b^{p_2}, \dots, \chi_b^{p_c}\}, \psi_b^p = \{\psi_b^{p_1}, \psi_b^{p_2}, \dots, \psi_b^{p_c}\}, \quad (2.51)$$

$$\chi_e^p = \{\chi_e^{p_1}, \chi_e^{p_2}, \dots, \chi_e^{p_c}\}, \psi_e^p = \{\psi_e^{p_1}, \psi_e^{p_2}, \dots, \psi_e^{p_c}\}, \quad (2.52)$$

Where

$\chi_b^{p_i}$- latitude of the beginning point of the crossing sector;

$\psi_b^{p_i}$- longitude of the beginning point of the crossing sector (crossing);

$\chi_e^{p_i}$- latitude of the ending point of crossing sector;

$\psi_e^{p_i}$- longitude of the ending point of the crossing sector;

c – number of trajectories crossing point;

t^p_{vest} – safe closing time of each trajectories crossing point $p \in P^2$.

There is no information either the output value is correct or not, that is why there is no possibility to use error back propagation algorithm.

Random sequential delta law self-training algorithm and target function was developed for the neural network training.

Function of the optimization is defined by two criteria:

- Collision possibility P with the aim of minimizing;
- Changes of the vehicles' speed $\Sigma\Delta vi$ with the aim of minimizing.

First criterion is connected with safety. The situation considered to be dangerous if trajectories of two transport vehicles has a common crossing point, and exists a probability, that transport vehicles will arrive at the crossing point of their trajectories at the same time. This situation is actual for each transport type.

Second criterion is connected with the transport traffic specifics. For example, train traffic precision is very important for railway transportation operations, because train delays causes obstacles for other trains. That is why timetable is very important for such transport vehicles as trains and trams. Also, time of departure and arrival is important for the public transport. That is why it is necessary to make minimal speed changes of such type of vehicles.

Based on the individual weighted criteria, the target function was developed:

$$F(\Delta v) = \begin{cases} P = \max{(P_{IJ})} \rightarrow min \\ \sum \Delta v_i \rightarrow min \end{cases} \tag{2.53}$$

Where,

Δv – changes of the speed of vehicle, solution of the task;

P – maximal collision probability;

P_{IJ} – each i-th vehicle collision possibility with each j-th vehicle;

Δv_i – change of speed of i-th vehicle.

Mathematical model for neural network was used:

- To develop algorithm for neural network for collision probability evaluation and minimization for electric transport dangerous situation recognition and prevention task (Chapter 3.5).
- To develop computer model for testing the algorithm of neural network for collision probability evaluation and minimization for electric transport dangerous situation recognition and prevention task (Chapter 4.5).
- To make experimental testing of the proposed algorithm of neural network for collision probability evaluation and minimization for electric transport dangerous situation recognition and prevention task (Chapter 5.5).

2.8. Mathematical model for the immune neural network

INFM immune neuro-fuzzy module calculates the necessary movement parameters change of the UEV electrical vehicle for the collision probability minimization task, by using INFN immune neuro fuzzy network, after collision probability is recognized by using the subsystem structure described in the chapter 1.2.3.

Each UEV analyses the situation for himself. UEVs in the control area are detected. These UEVs are ordered for a more precise definition of the situation. For ordering following parameters are used:

a. The first UEV in the input data (X) always is UEV itself;
b. Three parameters are used for ordering other UEVs:
    a. the horizontal movement direction τ_{UEVn},
    b. the vertical movement direction φ_{UEVn},
    c. the distance till the crossing point d_{UEVn}.

Ordering of other UEVs is done according to the slope τ_{UEVn} to these objects, starting from 0°, clockwise. If multiple UEVs have the same value of τ_{UEVn}, then ordering of these UEVs is done according to the slope φ_{UEVn} to these objects, starting from 0°, clockwise. If multiple UEVs have the same value of φ_{UEVn}, then ordering of these UEVs is done according to the distance to the crossing point with these UEV d_{UEVn}.

Training process of the proposed INN depends on the current situation s_j that is solved. Current situation is different for each situation participant, because each UEV sets different order of UEVs according to the own position. Set of situations stored in the immune memory IM is represented as follows:

$$\mathrm{S} = \{\mathrm{s}_1, \mathrm{s}_2, \dots, \mathrm{s}_\mathrm{m}\} \tag{2.54}$$

There is no identical situations stored in the IM, because of additional verification, which defines either this is the same situation or it is a new situation. If the situation is the same, data of this situation can be updated. If it is a new situation, new record of the situation occurs in the IM.

Each situation s_j contains input data X_j and number of participants n_j. All the data in IM is stored in clusters for easier and faster match finding process. For example, if three vehicles are participating in the possible collision situation, there is no need to find a similar situation in the group of situations with two participants. Therefore, method of clustering is used for data storage in IM and faster AA work.

$$s_j = < X_j > \tag{2.55}$$

$$\mathrm{S}_k \subseteq S, s_j \in S_k, |s_j| = |X| \tag{2.56}$$

Where,

X_j – Input data;

S_k – is a subset of all situations stored in IM and contains only those situations, where the dimensions of the situation s_j data X_j is the same as the dimensions of the given situation X.

The proposed INN can be made of one or multiple layers, depending on the task that is solved. INN that is proposed in chapter 1.2.4 consists of input layer, one hidden layer and output layer.

Input layer consists of input data X that describes the situation. The situation for n-th UEV in general may be described by the set of the following subsets:

$$X = (v_0, v_1, \tau_1, \varphi_1, d_1 \dots, v_n, \tau_n, \varphi_n, d_n) \tag{2.57}$$
$$= (x_0, x_1, x_2, x_3, x_4, \dots, x_{4n-3}, x_{4n-2}, x_{4n-1}, x_{4n})$$

Where,

n – number of other vehicles. $n = 0$ – own vehicle, $n>0$ – all others vehicles;

v_n, x_{4n-3} – speed of n-th electrical vehicle;

τ_n, x_{4n-2} – horizontal movement direction of the n-th electrical vehicle. Direction of movement of the another electric vehicle ($n>0$) is relative to one's own ($n = 0$) direction, but $\tau_0 = 0$;

φ_n, x_{4n-1}– vertical movement direction of the n-th electrical vehicle. Direction of movement of the another electric vehicle ($n>0$) is relative to one's own ($n = 0$) direction, but $\varphi_0 = 0$;

d_n, x_{4n} – distance till the possible crossing point of own vehicle with another vehicle's $n>0$ trajectory. Thus $d_0 = 0$.

Discrepancies between input data and data of the situation that is stored in the immune memory IM is represented as follows:

$$\mathcal{E} = \{\varepsilon_1, \dots, \varepsilon_k\} \tag{2.58}$$

Where,

$$\varepsilon_j = \left|X - X_j\right| = \sum |x_i - x_{ij}|, X_j \in S_k \tag{2.59}$$

Situation with a smallest discrepancy – closest match is represented as follows:

$$\varepsilon_\alpha = \min(\varepsilon) \tag{2.60}$$

Hidden layer is represented as follows:

$$\mu_{HID} = \{\mu_1, \dots, \mu_c\} \tag{2.61}$$

Where,

μ – specialized μ_h neuron.

Specialized μ_h neuron of the hidden layer is represented by following subsets:

$$\mu_h = \left\{I_{\mu h}, W_{\mu h}, AF_{\mu h}, O_{\mu h}\right\} \tag{2.62}$$

Where:

$I_{\mu h}$ – input of the μ_h neuron;

$W_{\mu h}$ – weights of the μ_h neuron;

$AF_{\mu h}$ – activation function of the μ_h neuron;

$O_{\mu h}$ – output of the μ_h neuron.

Input data of μ_h neuron of the hidden layer is represented by following subsets:

$$I_{\mu h} = \{X, \alpha, \beta\} \tag{2.63}$$

Where,

α – number of the situation with a smallest discrepancy ε_α;

β – signal, received from training algorithm TA.

Each μ_h neuron of the hidden layer stores weights for all situations stored in IM. Number of the weights of the hidden layer depends on the amount of participants n in the situation plus additional weight b_i, which is also related to the situation. Set of weights of the μ_h neuron of the hidden layer is represented as follows:

$$W_{\mu h} = \{< \alpha_1, W_1 >, \dots, < \alpha_m, W_m >\} \tag{2.64}$$

Where,

$$W_i = (w_{0i}, w_{1i}, w_{2i}, \dots, w_{4ni}, b_i) \tag{2.65}$$

Where, i – index of μ_h neuron.

A random number z is generated and the weight coefficient is shifted during the training process to receive new values of weights W_{ih}:

$$w_{ji}^{t-1} - z \leq w_{ji}^{t} \leq w_{ji}^{t-1} + z \tag{2.66}$$

Activation function of the μ_h neuron of the hidden layer is represented as a pure linear function:

$$O_{\mu h} = AF_{\mu h}(\mathrm{X}, W_i) = \sum_{j=0}^{n_i*4} x_j w_{ji} + b_i \tag{2.67}$$

Output layer is represented as follows:

$$\mu_{OUT} = \{\mu_1, \dots, \mu_d\} \tag{2.68}$$

Where,

μ – specialized μ_p neuron.

Number of specialized μ_p neurons of the output layer depend on the number of unknowns in the solved task. There are three unknowns in the task proposed in this research ($\Delta v_{UEV}, \Delta\tau_{UEV}, \Delta\varphi_{UEV}$), so formula of the output layer for the collision prevention task of the thesis is represented as:

$$\mu_{OUT} = \{\mu_1, \mu_2, \mu_3\} \tag{2.69}$$

Specialized μ_p neuron of the output layer is represented by following subsets:

$$\mu_p = \{I_{\mu p}, W_{\mu p}, AF_{\mu p}, O_{\mu p}\} \tag{2.70}$$

Where:

$I_{\mu p}$ – input of the μ_p neuron;

$W_{\mu p}$ – weights of the μ_p neuron;

$AF_{\mu p}$ – activation function of the μ_p neuron;

$O_{\mu p}$ – output of the μ_p neuron.

Number of the inputs of the output layer depends on the number of μ_h neurons in the hidden layer. Input data of μ_p neuron is represented by following subsets:

$$I_{\mu p} = \{O_{\mu h}, \alpha, \beta\} \tag{2.71}$$

Where,

$O_{\mu h}$ – Output data of the μ_h neurons of the hidden layer;

α – number of the situation with a smallest discrepancy ε_α;

β – signal, received from training algorithm TA.

Number of the weights of the output layer depends on the number of μ_h neurons of the hidden layer plus additional weight b_i, which is also related to the situation. Set of weights of the μ_p neuron of the output layer is represented as follows:

$$W_{\mu p} = \{< \alpha_1, W_1 >, \dots, < \alpha_m, W_m >\} \tag{2.72}$$

Where,

$$W_i = (w_{0i}, w_{1i}, w_{2i}, \dots, w_{4ni}, b_i) \tag{2.73}$$

Type of the activation function of μ_p neuron of the output layer depends on the solved task. Different types of functions can be used according to the desired result.

$$O_{\mu p} = AF_{\mu p}(O_{\mu h}, W_i) = f(r) \tag{2.74}$$

Where,

$$r = \sum_{j=0}^{h} y_j w_{ji} + b_i \tag{2.75}$$

Where,

n – speed Δv_i, horizontal movement direction $\Delta\tau_i$ or vertical movement direction $\Delta\varphi_i$ change limit.

Logarithmic sigmoid function is used as activation function of μ_p neuron of the output layer in this research. This type of function is used, because it is possible to set limits for speed and direction changes. It is important, because different types of vehicles have different parameters and limits for speed and trajectory change. However, limits of the speed and distance changes are checked by the proposed target function also.

$$O_{\mu p} = AF_{\mu p}(O_{\mu h}, W_i) = \log(\frac{1}{1 + e^{-r}}) \tag{2.76}$$

Outputs of the μ_p neurons also depend on the solved task. For example, rail transport cannot change direction in any moment of time. Only speed change can be done. As a result, there will be only one μ_p neuron and only one input $O_{\mu p}$. Three outputs of the output layer are proposed in this research:

$$O_{\mu p1} = \Delta v_{UEV} \tag{2.77}$$

$$O_{\mu p2} = \Delta \tau_{UEV} \tag{2.78}$$

$$O_{\mu p3} = \Delta \varphi_{UEV} \tag{2.79}$$

Where,
Δv_{UEV} – necessary speed change of the own UEV;
$\Delta \tau_{UEV}$ – necessary horizontal movement direction change of the own UEV;
$\Delta \varphi_{UEV}$ – necessary vertical movement direction change of the own UEV.

Target function was proposed to define the interest of the UEV. The ultimate goal of the target function is to minimize the collision probability of UEV by minimal changes of the speed and direction in the given state of environment. Target function evaluates the state of the environment and then assesses how the actions of the UEV will impact the situation.

$$TF = w(x, a_1, a_2, \dots, a_n) \rightarrow opt \tag{2.80}$$

Where:
TF – target function – the objective of UEV;
x – state of the environment;
a_i – action of the i-th UEV.

The information of each UEV depends on the state of environment:

$$y_i = y_i(x) \tag{2.81}$$

Where:
y_i – information of i-th UEV.

The decision rule of i-th UEV results an action of i-th electrical vehicle and depends on the information:

$$a_i = \rho_i(y_i) \tag{2.82}$$

Where:
ρ_i – decision rule of i-th UEV.

Interaction between i-th and j-th UEV:

$$q_{ij} = \partial \mathrm{w} / \partial \mathrm{a_i} \, \partial \mathrm{a_j} \tag{2.83}$$

A set of decision rules is optimal if
$E(w(x,(\rho_1(y_1), \dots, \rho_n(y_n))) \rightarrow$ max for a given probability distribution on x.
For anti-collision test the set of possible points of potential collisions is defined:

$$P = (p_1, p_2, \dots, p_c) \tag{2.84}$$

The location L^{UEV} of UEV is represented by three subsets $< \chi_c^{UEVS}, \psi_c^{UEVS}, \eta_c^{UEVS} >$, that are latitutde χ, longitude ψ and altitude η:

$$\chi_c^{UEV} = \{\chi_c^{UEV_1}, \chi_c^{UEV_2}, \dots, \chi_c^{UEV_n}\}, \tag{2.85}$$

$$\psi_c^{UEV} = \{\psi_c^{UEV_1}, \psi_c^{UEV_2}, \dots, \psi_c^{UEV_n}\}, \tag{2.86}$$

$$\eta_c^{UEV} = \{\eta_c^{UEV_1}, \eta_c^{UEV_2}, \dots, \eta_c^{UEV_n}\} \tag{2.87}$$

Where:

χ_c^{UEV} – latitude of the current electrical vehicle point;

ψ_c^{UEV} – longitude of the current electrical vehicle point;

η_c^{UEV} –altitude of the current electrical vehicle point.

Horizontal movement direction of the other electrical vehicle is used as an input data. The horizontal movement direction θ^{UEV} of UEVs is represented as follows:

$$\theta^{UEV} = \{\theta^{UEV1}, \theta^{UEV2}, \dots, \theta^{UEVn}\}, \tag{2.88}$$

Vertical movement direction of the other electrical vehicle is used as an input data. The vertical movement direction ω^{UEV} of UEVs is represented as follows:

$$\omega^{UEV} = \{\omega^{UEV1}, \omega^{UEV2}, \dots, \omega^{UEVn}\}, \tag{2.89}$$

The safety criterion is following:

$$D = |\mathrm{UEV_i UEV_j}| = \sqrt{(\chi_c^j - \chi_c^i)^2 + (\psi_c^j - \psi_c^i)^2 + (\eta_c^j - \eta_c^i)^2} > D_{safe} \tag{2.90}$$

Where: D_{safe} is safety distance limit for each pair of $< \mathrm{UEV_i}, \mathrm{UEV_j} >$, $\mathrm{i} = 1..\mathrm{n}$, $\mathrm{j} = 1..\mathrm{n}$, $\mathrm{i} \neq \mathrm{j}$

Permissible changes of direction depend on the UEV specifications and other circumstances. Restrictions for the own horizontal movement direction change were also defined:

$$\tau_1^{UEV_i} < \tau^{UEV_i} < \tau_2^{UEV_i} \tag{2.91}$$

Restrictions for the own movement direction (in vertical plane) change were also defined:

$$\varphi_1^{UEV_i} < \varphi^{UEV_i} < \varphi_2^{UEV_i} \tag{2.92}$$

Restrictions for the speed change were also defined:

$$v_1^{UEV_i} < v^{UEV_i} < v_2^{UEV_i} \tag{2.93}$$

So, the general target function with anti-collision criteria is following:

$$\begin{cases} P_{max}(\chi_c^{UEV}, \psi_c^{UEV}, \eta_c^{UEV}, \Delta\tau, \Delta\varphi, \Delta v) = max(P_{IJ}) \rightarrow min \\ \Delta\tau_{\Sigma}(\Delta\tau) = \sum_{i=1}^{n} \Delta\tau_i \rightarrow min \\ \Delta\varphi_{\Sigma}(\Delta\varphi) = \sum_{i=1}^{n} \Delta\varphi_i \rightarrow min \\ \Delta v_{\Sigma}(\Delta v) = \sum_{i=1}^{n} \Delta v_i \rightarrow min \\ D = \left|UEV_i UEV_j\right| > S \\ \Delta\tau_1 < \Delta\tau_i < \Delta\tau_2 \\ \Delta\varphi_1 < \Delta\varphi_i < \Delta\varphi_2 \\ \Delta v_1 < \Delta v_i < \Delta v_2 \\ i = 1..n, j = 1..n, i \neq j \end{cases} \quad (2.94)$$

Where:

P_{max} - maximal collision probability from the set of probabilities of collision for all pairs of UEVs;

$\Delta\tau = (\Delta\tau_1, \ldots, \Delta\tau_n)$ - set direction changes in horizontal plane of all UEVs;

$\Delta\varphi = (\Delta\varphi_1, \ldots, \Delta\varphi_n)$ - set direction changes in vertical plane of all UEVs;

$\Delta v = (\Delta v_1, \ldots, \Delta v_n)$ - set of speed changes of all UEVs;

$P_{IJ} = (P(\langle UEV_1, UEV_2\rangle),\ldots,P(\langle UEV_i, UEV_j\rangle),\ldots,P(\langle UEV_{n-1}, UEV_n\rangle))$ - set of probabilities of collision for all pairs of UEVs $\langle UEV_i, UEV_j\rangle$, $i \neq j$, $i,j = 1..n$.

Each i-th UEV is looking for its own direction and/or speed change solution $< \Delta\tau_i, \Delta\varphi_i, \Delta v_i >$, according to the task. Thus, the target function for a single UEV can be expressed as:

$$\begin{cases} P_{max}(\chi_c^{UEV}, \psi_c^{UEV}, \eta_c^{UEV}, \Delta\tau_0, \Delta\varphi_0, \Delta v) = max\,(P_{0j}) \rightarrow min \\ \Delta\tau_0 \rightarrow min \\ \Delta\varphi_0 \rightarrow min \\ \Delta v_0 \rightarrow min \\ D = \left|UEV_0 UEV_j\right| > S \\ \Delta\tau_1 < \Delta\tau_0 < \Delta\tau_2 \\ \Delta\varphi_1 < \Delta\varphi_0 < \Delta\varphi_2 \\ \Delta v_1 < \Delta v_0 < \Delta v_2 \\ j = 1..n \end{cases} \quad (2.95)$$

Where:

P_{max} - represents the highest probability of collision between the own UEV_0 and all other UEVs within the control area;

$\Delta\tau_0$ - direction change in horizontal plane of the own UEV_0;

$\Delta\varphi_0$ - direction change in vertical plane of the own UEV_0;

Δv_0 - speed change of the own UEV_0;

$P_{0j} = (P(\langle UEV_0, UEV_1\rangle),\ldots,P(\langle UEV_0, UEV_j\rangle),\ldots,P(\langle UEV_0, UEV_n\rangle))$ - set of probabilities of collision between the own UEV_0 and all other UEVs within the control area, $j = 1..n$.

Function of the decision making module F_{DM} is represented as follows:

$$F_{DM} = TF(P_{max}, \Delta v_{UEV}, \Delta\tau_{UEV}, \Delta\varphi_{UEV}) \rightarrow min \quad (2.96)$$

Thus, we can evaluate the result of training the INN without a teacher with the help of the proposed target function and make a decision about accepting the solution or continue training.

Mathematical model for the novel immune neural network was used:

- To develop novel algorithm for immune neural network for unsupervised collision probability evaluation and minimization for electrical vehicle dangerous situation recognition and prevention task (Chapter 3.6).
- To develop a computer model for testing the novel algorithm of immune neural network for unsupervised collision probability evaluation and minimization for electrical vehicle dangerous situation recognition and prevention task (Chapter 4.6).
- To make experimental testing of the proposed novel algorithm of the immune neural network for unsupervised collision probability evaluation and minimization for electrical vehicle dangerous situation recognition and prevention task (Chapter 5.6).

2.9. Chapter 2 summary

Second chapter of doctoral thesis was devoted to the developed mathematical models. For this task not only already known methods, such as ANN and CNN were used, but developed novel INN based technology of machine learning for unsupervised safe vehicle control was used also:

1. Mathematical sets of system objects were defined.
2. Mathematical models for objects and signal recognition task were developed:
    a. Mathematical model for traffic light red signal recognition task;
    b. Mathematical model for object recognition task.
3. Mathematical model for collision probability evaluation and location of the possible crossing point calculation was developed:
    a. Mathematical model for possible crossing point detection and collision probability evaluation task. The algorithm of the electrical vehicle collision risk solving system has been proposed based on basic collision risk and vulnerabilities of accidents.
4. Mathematical models for collision probability minimization task were developed:
    a. Mathematical model for the neural network;
    b. Novel mathematical model for the immune neural network.

Algorithms for electric vehicle dangerous situation recognition and prevention task were developed and described in the third chapter of the doctoral thesis.

3. DEVELOPED ALGORITHMS FOR ELECTRIC VEHICLE DANGEROUS SITUATION RECOGNITION AND PREVENTION TASK

3.1. Situation description

Several tasks are performed in this research, according to the proposed system structure and mathematical model. Developed algorithms for different tasks of the research are provided in the third chapter of the doctoral thesis: algorithm for traffic light red signal recognition method for electric transport dangerous situation recognition and prevention task; algorithm for CNN for object recognition for electric transport dangerous situation recognition and prevention task; algorithm for electrical transport collision probability evaluation task; algorithm for neural network for collision probability evaluation and minimization for electric transport dangerous situation recognition and prevention task; novel algorithm for novel INN for unsupervised collision probability evaluation and minimization for electrical vehicle dangerous situation recognition and prevention task.

3.2. Developed algorithm for traffic light red signal recognition method for electric transport dangerous situation recognition and prevention task

Following attributes were used to develop this algorithm:

- Mathematical model for traffic light red signal recognition task (Chapter 2.4).

Algorithm for the traffic light red signal recognition was developed:

STEP 0. Camera connection.

STEP 1. Setting the parameters:

- Picture size variables: height (pixels); width (pixels);
- Sensitivity;
- Limiting variables: point limit for red color (redlim); point limit for green color (greenlim).

STEP 2. Picture “original” creating.

STEP 3. Picture “original” output.

STEP 4. Region of interests (ROI) detection in “original” picture by detection the traffic light, by using the CNN. ROI area marking on the picture.

STEP 5. ROI picture output.

STEP 6. Copy ROI into RGB colors picture.

STEP 7. Dividing RGB picture into three different channels:

- red (saving as r_plane variable),
- green (saving as g_plane variable),
- blue (saving as b_plane variable).

STEP 8. Setting the channels intervals (from 0 to 255) for red and green colors.

STEP 8.1. Channels were set for the red color:

R1 <= R <= R2;

G1 <= G <= G2;

B1 <= R <= B2.

STEP 8.2. Channels were set for the green color:

R3 <= R <= R4;

G3 <= G <= G4;

B3 <= R <= B4.

STEP 9. Finding all points in r_plane, g_plane and b_plane channels, which meet the condition of specified in STEP 8 channels intervals. Saving the result into r_range, g_range and b_range variables respectively.

STEP 10. r_range, g_range, b_range channels alteration and saving into the rgb_and variable.

STEP 11. Counting the sum of points in the rgb_and picture: number of points / 255 and saving the result into massive parry.

STEP 12. If the number of frames is less than specified, then sum up all the sums of points of all frames, else sum up the sums of points of the last 10 frames.

STEP 13. Setting the condition: if the sum of points > redlim, then the traffic light shows red color, else if the sum of points > greenlim, then the traffic light shows green color.

Developed algorithm for traffic light red signal recognition method for electric transport dangerous situation recognition and prevention task was used:

- To develop computer model for testing the algorithm of traffic light red signal recognition method for electric transport dangerous situation recognition and prevention task (Chapter 4.2).
- To make experimental testing of the proposed algorithm of traffic light red signal recognition method for electric transport dangerous situation recognition and prevention task (Chapter 5.2).

3.3. Developed algorithm for convolutional neural network for object recognition for electric transport dangerous situation recognition and prevention task

Following attributes were used to develop this algorithm:

- Mathematical model for object recognition task (Chapter 2.5).

CNN is good in object recognition process. A lot of examples were mentioned in Introduction chapter. An algorithm for CNN was also developed during this research:

STEP 0. Initialization:

- Initialize training set $TR = (<X_1,T_1> ,\ldots,<X_n,T_n>)$;
- Set parameters for each $CONV_{CNN}$ layer n K_{CNN}, $F_{CNN}c_i$, $S_{CNN}c_i$,P_{CNN};

- Set parameters for each $POOL_{CNN}$ layer n $F_{CNN}p_j$, $S_{CNN}p_j$;
- Set parameters for FC_{CNN} subnetwork HID_{CNN}, CL_{CNN}, ρ;
- Set K weight patterns for $CONV_{CNN1}$ layer $F_{CNN}xF_{CNN}$ ($F_{CNN}=F_{CNNc1}$), w_0 bias: PAT = (<$w^1{}_0$, $w^1{}_{11}$, …, $w^1{}_{FF}$>,…, < $w^1{}_0$,$w^k{}_{11}$, …, $w^k{}_{FF}$>);
- Generate initial random weights and biases for other $CONV_{CNN2}$ … $CONV_{CNNN}$ layers;
- Generate initial random weights and biases for FC_{CNN} network;
- Set index of selected image for training tr = 1;
- Set initial number training set and output matching ok = 0;
- Set logical value of made weight corrections cor = false.

STEP 1. Selection of the image:

- Select image TR_{tr} for training of CNN;
- Image spatial parameters are W_{CNN0}×H_{CNN0}×D_{CNN0};
- Set index i for next $CONV_{CNN}$-$POOL_{CNN}$ layers i = 1;
- Set initial number output classes value matching ok2 = 0.

STEP 2. Convolution of the image.

Using each j-th n of $CONV_{CNNi}$ layer convolute each j-th image region F_{CNNci} × F_{CNNci} using K_{CNN} patterns (weights):

$$u_j = \left(\sum_{a=1}^{FCNNc_i} \sum_{b=1}^{FcCNN_i} x_{ab} w^j_{a*FCNNc_i+b} \right) + w^j_0 \tag{3.1}$$

Where,

a, *b* – pixel 2D spatial (x,y) coordinates in the region F_{CNNci} × $_{FCNNci}$.

STEP 3. Pooling and feature maps.

Using each j-th n of $POOL_{CNN}$ layer down sampling each feature map k = 1.. K_{CNN} in region F_{CNNpi} × F_{CNNpi}:

$$u^k_j = \max\left(x^k_{ab}\right) \tag{3.2}$$

Where, a = 1..F_{CNNpi}, b = 1..F_{CNNpi} 2D spatial (x, y) coordinates F_{CNNpi} × F_{CNNp}.

STEP 4. Repeat convolutions and pooling by next layers.

$$i = i + 1 \tag{3.3}$$

If i<=N then go to STEP 2 else go to STEP 5

STEP 5. Feed-forward to FC_{CNN} subnetwork.

The results of N-th pooling layer $POOL_{CNN}$ with K_{CNN} feature maps are feed forwarded to FC_{CNN} subnetwork:

$$X_{FC} = \left(u^1_1, ..., u^1_{FCNNc^2_N/FCNNp^2_N}, \dots, u^k_j, \dots, u_1^{K_N}, \dots,\ u^{K_N}_{FCNNc^2_N/FCNNp^2_N}\right) \tag{3.4}$$

Where,

$$k = 1..K_N \tag{3.5}$$

$$j = 1..\ FCNNc_N^2 / FCNNp_N^2 \tag{3.6}$$

STEP 6. Feed-forward to the output via hidden layer of FC_{CNN}

Calculate and activate each h-th n of hidden layer:

$$u_h = tansig\left(\sum_{j=1}^{||X_{FCCNN}||+1} x_j \cdot w_j^h + w_0^h\right), h = 1..HIDCNN \tag{3.7}$$

STEP 7. Feed-forward the output to the competitive layer.

Calculate and activate the output of each c-th n of the output layer:

$$u_c = tansig\left(\sum_{j=1}^{||X_{HIDCNN}||+1} x_j \cdot w_j^c + w_0^c\right), c = 1..CLCNN \tag{3.8}$$

STEP 8. Getting the final classification result.

Calculate and activate the competitive layer:

$$u_{max} = \max(u_1, \dots, u_{CLCNN}) \tag{3.9}$$

$$OUT_{CNN} = \begin{cases} 1,\ u_c = u_{max} \\ 0, u_c < u_{max} \end{cases} \tag{3.10}$$

STEP 9. Calculating error for the output layer.

The error is calculated for each c-th n of output layer:

$$\delta_c = (t_c^{tr} - u_c) \cdot u_c \cdot (1 - u_c), c = 1..CLCNN \tag{3.11}$$

$$ok2 = \sum_{c=1}^{CLCNN} (\,t_c^{tr} \& OUT_{CNN}) \quad ok = ok + ok2 \tag{3.12}$$

STEP 10. Backpropagation error to the hidden layer.

The error is calculated for each h-th n of hidden layer, using error of each c-th n of the output layer:

$$\delta_h = \sum_{c=1}^{CLCNN} w_h^c \cdot \delta_c \cdot u_h \cdot (1 - u_h),\ \ h = 1..HIDCNN \tag{3.13}$$

STEP 11. Checking necessity for weights corrections.

If ok2=0 then go to STEP 12, else go to STEP 13.

STEP 12. Correcting all weights and biases of FC_{CNN}

$$w_h^c = w_h^c + \sum_{j=1}^{CLCNN+1} \rho \cdot \delta_j \cdot u_h\,,\ \ h = 1..HIDCNN, c = 1..CLCNN \tag{3.14}$$

$$w_x^h = w_x^h + \sum_{j=1}^{HIDCNN+1} \rho \cdot \delta_j \cdot u_x\,,\ \ x = 1..K_{CNN}, h = 1..HIDCNN \tag{3.15}$$

$$\text{cor} = \text{true} \tag{3.16}$$

STEP 13. Continue with next image from training set.

Developed algorithm for convolutional neural network for object recognition for electric transport dangerous situation recognition and prevention task was used:

- To develop computer model for testing the algorithm of convolutional neural network for object recognition for electric transport dangerous situation recognition and prevention task (Chapter 4.3).
- To make experimental testing of the proposed algorithm of convolutional neural network for object recognition for electric transport dangerous situation recognition and prevention task (Chapter 5.3).

3.4. Developed algorithm for electrical transport collision probability evaluation task

Following attributes were used to develop this algorithm:

- Mathematical model for possible crossing point detection and collision evaluation task (Chapter 2.6).

The algorithm of collision probability recognition and minimization for the traffic safety is following:

STEP 1. Data obtaining. Vehicle receives the information about own parameters: speed and coordinates of the center point by using GNSS and other vehicles speed and coordinates of the four vehicle's angles (left front angle, right front angle, left rear angle and right rear angle) by using RF.

STEP 2. The necessity of future calculations is detected, after the information is received.

STEP 2.1. Calculations of the vehicles front and rear angles coordinates are done.

$$RF = (\chi_{UEV} + \frac{w_{UEV}}{2};\ \psi_{UEV} + \frac{h_{UEV}}{2}) \tag{3.17}$$

$$\mathrm{LF} = (\chi_{UEV} - \frac{w_{UEV}}{2};\ \psi_{UEV} + \frac{h_{UEV}}{2}) \tag{3.18}$$

$$RR = (\chi_{UEV} + \frac{w_{UEV}}{2};\ \psi_{UEV} - \frac{h_{UEV}}{2}) \tag{3.19}$$

$$LR = (\chi_{UEV} - \frac{w_{UEV}}{2};\ \psi_{UEV} - \frac{h_{UEV}}{2}) \tag{3.20}$$

Where,

RF - coordinates of the right front angle of the vehicle;

LF - coordinates of the left front angle of the vehicle;

RR - coordinates of the right rear angle of the vehicle;

LR - coordinates of the left rear angle of the vehicle;

χ_{UEV} - latitude of the center point of the vehicle;

ψ_{UEV} - longitude of the central point of the vehicle;
w_{UEV} - width of the vehicle;
h_{UEV} - length of the vehicle.

STEP 2.2. Movement direction calculation. Front and rear angles are compared for this purpose. The example is shown in table 3.1.

Where: N – North; NE – North-East; E – East; SE – South-East; S – South; SW – South-West; W – West; NW – North-West.

3.1 table

Movement direction recognition

Situation Nr.	**Left rare angle coordinates in comparison with the left front angle coordinates**		**Movement direction**
	Latitude	**Longitude**	
1	=	<	N
2	<	<	NE
3	<	=	E
4	<	>	SE
5	=	>	S
6	>	>	SW
7	>	=	W
8	>	<	NW

STEP 2.3. Detecting objects in the control area - in front of the vehicle or on the same level. For this purpose, coordinates of the vehicle's angles are compared, similar as it was done in the STEP 2.2. If object is not in the control area, then no further calculations are needed, otherwise STEP 3 is done.

STEP 3. Coordinates of the crossing point calculation. As coordinates of left and right angles of vehicles are known, formulas of the straight lines of the left and right sides are calculated and crossing points of these lines are detected. In the result, 4 coordinates of the crossing point are calculated. Only χ_{UEV} coordinates are considered for calculating crossing point, because according to the algorithm, own ψ_{UEV} coordinate of each vehicle is equal zero and only χ_{UEV} coordinate can be changed.

$$yR_1 = k_1 * xR_1 + bR_1 \tag{3.21}$$

$$yL_1 = k_1 * xL_1 + bL_1 \tag{3.22}$$

$$yR_2 = k_2 * xR_2 + bR_2 \tag{3.23}$$

$$yL_2 = k_2 * xL_2 + bL_2 \tag{3.24}$$

$$yyR1R2 = cross(yR_1, yR_2) \tag{3.25}$$

$$yyR1L2 = cross(yR_1, yL_2) \tag{3.26}$$

$$yyL1R2 = cross(yL_1, yR_2) \tag{3.27}$$

$$yyL1L2 = cross(yL_1, yL_2) \tag{3.28}$$

Where,
yR_1 – line of the right side of UEV1;
yL_1 – line of the left side of UEV1;
yR_2 – line of the right side of UEV2;
yL_2 – line of the left side of UEV2;
k_1, k_2 – angle coeficients;
$yyR1R2$ – coordinates of right rear angle of the crossing area;
$yyR1L2$ – coordinates of left rear angle of the crossing area.

STEP 4. Distance till the crossing point calculation is done

$$ddR1R2 = \sqrt{(yyR_1R_2 - yRR_1)^2} \tag{3.29}$$

$$ddL1R2 = \sqrt{(yyL_1R_2 - yLR_1)^2} \tag{3.30}$$

$$ddR1L2 = \sqrt{(yyR_1L_2 - yRF_1)^2} \tag{3.31}$$

$$ddL1L2 = \sqrt{(yyL_1L_2 - yLF_1)^2} \tag{3.32}$$

STEP 4.1. Minimal distance till the crossing point calculation. It is necessary to calculate the distance for the vehicles left and right sides, to detect the minimal distance. The smallest value will be the minimal distance.

$$d_1^{min} = (ddR1L2, ddL1L2) \tag{3.33}$$

STEP 4.2. Maximal distance till the crossing point calculation. It is necessary to calculate the distance for the vehicles left and right sides, to detect the maximal distance. The biggest value will be the maximal distance.

$$d_1^{max} = (ddR1R2, ddL1R2) \tag{3.34}$$

STEP 5. Time till the crossing point calculation.
STEP 5.1. Minimal time till the crossing point calculation.

$$t_n^{min} = d_1^{min} \,/\, V_n \tag{3.35}$$

Where,
V_n - speed of the vehicle.

STEP 5.2. Maximal time till the crossing point calculation.

$$t_n^{max} = d_1^{max} / V_n \tag{3.36}$$

STEP 6. Collision probability evaluation is done.
IF $t_2^{min} > t_1^{min}$ AND $t_1^{max} > t_2^{max}$ AND $t_2^{max} > t_1^{min}$, then:

$$P1 = \frac{t_1^{max} - t_2^{min}}{t_1^{max} - t_1^{min}} \tag{3.37}$$

$$P2 = \frac{t_1^{max} - t_2^{min}}{t_2^{max} - t_2^{min}} \tag{3.38}$$

$$P = P1 * P2 \tag{3.39}$$

IF $t_2^{min} < t_1^{min}$ AND $t_2^{max} < t_1^{max}$ AND $t_1^{min} < t_2^{max}$, then:

$$P1 = \frac{t_2^{max} - t_1^{min}}{t_2^{max} - t_2^{min}} \tag{3.40}$$

$$P2 = \frac{t_2^{max} - t_1^{min}}{t_1^{max} - t_1^{min}} \tag{3.41}$$

$$P = P1 * P2 \tag{3.42}$$

IF $t_1^{min} < t_2^{min}$ AND $t_1^{max} < t_2^{min}$ OR $t_2^{min} < t_1^{min}$ AND $t_2^{max} < t_1^{min}$, then: P = 0.
IF $t_1^{min} <= t_2^{min}$ AND $t_1^{max} >= t_2^{max}$, then:

$$P = \frac{t_2^{max} - t_2^{min}}{t_1^{max} - t_1^{min}} \tag{3.43}$$

IF $t_2^{min} <= t_1^{min}$ AND $t_2^{max} >= t_1^{max}$, then:

$$P = \frac{t_1^{max} - t_1^{min}}{t_2^{max} - t_2^{min}} \tag{3.44}$$

IF P = 0 THEN „SAFE”;
IF 0 < P <= 0,05 THEN ”VERY SMALL collision probability”;
IF 0,05 < P <= 0,2 THEN ”SMALL collision probability”;
IF 0,2 < P <= 0,5 THEN ”MEDIUM collision probability”;
IF 0,5 < P <= 0,8 THEN ”HIGH collision probability”;
IF 0,8 < P < 1 THEN ” VERY HIGH collision probability”;
IF P = 1 THEN ”COLLISION”.

Developed algorithm for electrical transport collision probability evaluation task was used:

- To develop computer model for testing the algorithm of electrical transport collision probability evaluation task (Chapter 4.4).

- To make experimental testing of the proposed algorithm of electrical transport collision probability evaluation task (Chapter 5.4).

3.5. Developed algorithm for neural network for collision probability evaluation and minimization for electric transport dangerous situation recognition and prevention task

Following attributes were used to develop this algorithm:

- Mathematical model for the neural network (Chapter 2.7).

Developed algorithm consists of such steps:

Initialization:

- Index of training group element e = 1;
- Chosen for the correction n sn = 1;
- Chosen for the correction weight sw = 1;
- Retraining = false.

STEP 1. Take the element $e = \{d_{tr}^e, v_{tr}^e, d_{tr}^e, v_{tr}^e\}$ from the training set.

STEP 2. $x = (e_1, e_2, \ldots, e_n)$

STEP 3. Read the x_{min} and x_{max} parametres, which limit n network output.

STEP 4. Calculate output n adder values:

$$\sum\nolimits_j = \left(\sum\nolimits_{i=1}^{2n} x_i * w_{ij}\right) + b_j \qquad j = \overline{1..n} \tag{3.45}$$

Generate output layer n output value by positively and negatively saturated linear activation function:

$$\Delta v_{tr} = \begin{cases} x_{min}, & \sum_j \leq x_{min} \\ \sum\nolimits_j, & x_{min} < \sum_j < x_{max} \\ x_{max}, & \sum_j \geq x_{max} \end{cases} \tag{3.46}$$

STEP 5. Save the previous valuation, if it exists P_{tr0}, $\Sigma\Delta v_{tr0}$.

STEP 6. Evaluate the solution, that was found, using the target function $[P_{tr}, \Sigma\Delta v_{tr}] = TF(\Delta v_{tr})$.

STEP 7. If $P_{tr} > P_{safe}$ or $\Sigma\Delta v_{tr} > \Sigma\Delta v_{trlim}$, then go to STEP 8.

STEP 8. If the last element of the training set is not reached $e \neq e_{max}$,

then e=e+1 and go to STEP 1,

else if there is no need to retrain the network then FINISH,

else e = 1 and go to STEP 1.

STEP 9. Weight correction occurs sequentially:

If ($sn \neq 1$ and $sw \neq 1$) or ($P_{tr0} < P_{tr}$ and $\Sigma\Delta v_{tr0} < \Sigma\Delta v_{tr}$), that means if the element isn't first and the result is worse than it was before, then weight correction is done

$$w_{sw,\,sn} = w_{sw,\,sn} - k \tag{3.47}$$

Where, *k* is a random number.

$$k = \text{random (xmin, xmax)}/10\,000 \tag{3.48}$$

If sw > 2n, then sn = sn + 1, else sw = sw + 1.

If sn > n, then sn = 1, sw = 1.

STEP 10. If weight correction was done, then neural network must be retrained. Retrain = true.

STEP 11. Go to STEP 3.

Developed algorithm for neural network for collision probability evaluation and minimization for electric transport dangerous situation recognition and prevention task was used:

- To develop computer model for testing the algorithm of neural network for collision probability evaluation and minimization for electric transport dangerous situation recognition and prevention task (Chapter 4.5).
- To make experimental testing of the proposed algorithm of neural network for collision probability evaluation and minimization for electric transport dangerous situation recognition and prevention task (Chapter 5.5).

3.6. Novel developed algorithm for immune neural network for unsupervised collision probability evaluation and minimization for electrical vehicle dangerous situation recognition and prevention task

Following attributes were used to develop this algorithm:

- Mathematical model for the immune neural network (Chapter 2.8).

The algorithm of INN consists of the following steps:

STEP 1. Receive input data DAT from n UEVs located in the area of visibility. These data are locations $<\chi_c^{\mathrm{UEV}}, \psi_c^{\mathrm{UEV}}, \eta_c^{\mathrm{UEV}}>$, speed v^{UEV}, horizontal movement direction θ^{UEV} and vertical movement direction ω^{UEV} of UEV:

$$\mathrm{DAT} = (\chi_c^{\mathrm{UEV}}, \psi_c^{\mathrm{UEV}}, \eta_c^{\mathrm{UEV}}, \theta^{\mathrm{UEV}}, \omega^{\mathrm{UEV}}, v^{\mathrm{UEV}}) \tag{3.49}$$

Where,

χ_c^{UEV} – latitude of the UEV actual position;

ψ_c^{UEV} – longitude of the UEV actual position;

η_c^{UEV} – altitude of the UEV actual position;

θ^{UEV} – actual horizontal movement direction of the UEV;
ω^{UEV} – actual vertical movement direction of the UEV;
v^{UEV} – actual speed of the UEV.

STEP 2. The proposed INN requires data about other vehicles' location in relation to the own UEV location. Data DAT needs to be proceeded before it will enter the input layer of the proposed INN.

STEP 2.1. Input data DAT contains coordinates of other vehicles position $<\chi_c^{\mathrm{UEV}}, \psi_c^{\mathrm{UEV}}, \eta_c^{\mathrm{UEV}}>$, own position is known also. Distances to the possible crossing points with other vehicles d_{UEV} are calculated. The algorithm is described in the chapter 3.4. Two formulas are provided in chapter 3.4: formula for the minimal d_n^{min} (formula 3.33) and maximal d_n^{max} (formula 3.34) distance till the crossing point. Only minimal distance d_n^{min} is calculated in this step.

STEP 2.2. The next step is to organize the UEVs for a more precise definition of the situation. The first UEV is always UEV itself. The other UEVs are ordered according to their horizontal movement direction τ_n^{UEV} in relation to the own UEV, starting from 0 degrees and proceeding clockwise. If multiple UEVs have the same value of τ_n^{UEV}, they are then ordered according to their vertical movement direction φ_n^{UEV} in relation to the own UEV, starting from 0 degrees and proceeding clockwise. If multiple UEVs have the same value of φ_n^{UEV}, they are then ordered according to the distance to the crossing point with the own UEV d_n^{UEV}.

Horizontal movement directions of other UEVs τ_n in relation to the own UEV direction are calculated as follows:

$$\tau_n = \tan^{-1}((\tan\theta_n^{UEV} - \tan\tau_1)/(1 + \tan\tau_1 \tan\theta_n^{UEV})) \tag{3.50}$$

Where,
τ_1 – direction of the own UEV in horizontal plane;
θ_n^{UEV}– direction of other UEV in horizontal plane.

Vertical movement directions of other UEVs φ_n in relation to the own UEV direction are calculated as follows:

$$\varphi_n = \tan^{-1}((\eta_n^{\mathrm{UEV}} - \eta_1)/d_n) \tag{3.51}$$

Where,
η_1 – altitude of the own UEV;
η_n^{UEV} – altitude of other UEV;
d_n – horizontal distance between own UEV and the n-th UEV.

$$d_n = \sqrt{(\Delta\eta^2 + \Delta d^2)} \tag{3.52}$$

Where,
$\Delta\eta$ – is the difference in altitude between the two UEV;
Δd – is the horizontal distance between the two UEV.

Actions, provided in STEP 2, transform input data DAT to the input data X, that can be processed by the INN.

$$X = (v_0, v_1, \tau_1, \varphi_1, d_1 \dots, v_n, \tau_n, \varphi_n, d_n) \quad (3.53)$$
$$= (x_0, x_1, x_2, x_3, x_4, \dots, x_{4n-3}, x_{4n-2}, x_{4n-1}, x_{4n})$$

Where,

n – number of other vehicles, *n = 0* – own vehicle, *n>0* – all others vehicles;

v_n, x_{4n-3} – speed of n-th electrical vehicle;

$\tau_n,\ x_{4n-2}$ – horizontal movement direction of the n-th electrical vehicle. Direction of movement of the another electric vehicle (*n>0*) is relative to one's own (*n = 0*) direction, but $\tau_0 = 0$;

$\varphi_n,\ x_{4n-1}$– vertical movement direction of the n-th electrical vehicle. Direction of movement of the another electric vehicle (*n>0*) is relative to one's own (*n = 0*) direction, but $\varphi_0 = 0$;

$d_n,\ x_{4n}$ – distance till the possible crossing point of own vehicle with another vehicle's *n>0* trajectory. Thus $d_0 = 0$.

STEP 3. The calculation of collision probability P is intended to determine whether it is necessary to minimize the risk of collision. If no, end of the algorithm. If yes, then go to the next step. Collision probability algorithm is proposed in chapter 3.4.

STEP 4. After input data X enters the input layer of INN, data X is sent to the specialized μ neurons and affinity algorithm (AA). The AA (X, S) checks all situations S stored in the IM $\mathrm{S} = \{\mathrm{s}_1, \mathrm{s}_2, \dots, \mathrm{s}_\mathrm{m}\}$, calculates the set of discrepancies $\mathcal{E} = (\varepsilon_1, \dots, \varepsilon_\mathrm{k})$, where:

$$\varepsilon_\mathrm{j} = \sum_{\mathrm{i}=0}^{\mathrm{n}} \sum_{\mathrm{k}=1}^{2} \left(\frac{X_{\mathrm{ik}} - X_{\mathrm{ik}}^{\mathrm{j}}}{X_{\mathrm{ik}}} \right)^2 \quad (3.54)$$

and finds the closest match ε_α, where:

$$\varepsilon_\alpha = \min\ (\varepsilon) \quad (3.55)$$

STEP 5. When μ neuron receives input data X, it activates and increases iteration counter $t = t + 1$.

- When situation number α is received, set of weights W_μ are selected from the memory of the μ neuron.
- If there is no similar situation in the IM and $\alpha = 0$, then $W_\mu = 0$.

STEP 6. Input data X, situation number α, received from the affinity algorithm AA, and signal β, that indicates the need to recalculate the weights of μ neurons, are the input data of each μ neuron of the hidden layer μ_{HID}. Feed forward input through the NN is done. Formulas are provided in chapter 2.8. Outputs for own vertical movement direction change $O_{\mu p3} =$

$\Delta\varphi_{UEV}$, own horizontal movement direction change $O_{\mu p2} = \Delta\tau_{UEV}$ and own speed change $O_{\mu p1} = \Delta v_{UEV}$ are generated as a result.

STEP 7. The TF calculates the collision probability P_{max}, that is maximal collision probability from the set of probabilities of collision for all pairs of UEVs P_{IJ}. TF uses updated data, received directly from the UEV embedded device D_{TR}.

STEP 7.1. The TF function defines the directions τ^{UEV} and φ^{UEV} of each UEV in relation to the own UEV.

STEP 7.2. Next step is to detect the crossing point (χ_p, ψ_p, η_p) in 3D space.

STEP 7.3. If the crossing point (χ_p, ψ_p, η_p) is found and is located on the way of motion, then go to STEP 7.4. else go to STEP 7.6.

STEP 7.4. The distance between altitudes of i-th and own UEV is calculated for the (χ_p, ψ_p, η_p) point $\Delta\eta = \eta_p^i - \eta_p^{own}$.

STEP 7.5. If $\Delta\eta \leq D_{safe}$ then it is assumed that potentially dangerous point exists, and the probability of collision P is calculated. Collision probability algorithm is proposed in chapter 3.4.

STEP 7.6. If the crossing point (χ_p, ψ_p, η_p) is not found, then the trajectories are parallel and D_{safe} should be checked for safe passing.

STEP 8. If $P_{max} > P_{safe}$, where P_{safe} is maximal acceptable (safe) collision probability, then is checked either the solution is better than previous or worse

- If $t = 1$, then signal β is sent to all μ neurons and repeat from STEP 6.
- If $1 < t < T_{max}$ and $P_{max2} > P_{max1}$, then signal β is sent to all μ neurons. μ neurons return the previous values of W_μ and repeat from STEP 6.
- If $1 < t < T_{max}$ and $P_{max2} \leq P_{max1}$, then signal β is sent to all μ neurons and repeat from STEP 6.
- If $t \geq T_{max}$, then situation cannot be solved in the defined time, so the safe solution is necessary. In this research such solution is speed reduction $\Delta v_i = v$ and END algorithm else go to STEP 9.

STEP 9. If $P_{max} \leq P_{safe}$ then:

- The calculated speed Δv_{UEV}, horizontal and vertical movement directions $\Delta\tau_{UEV}$ and $\Delta\varphi_{UEV}$ changes are accepted as the solution and sent to the embedded electronic device for UEV control.
- Match error ε_a is compared with a maximal possible match error ε_{lim}, responsible for creation new record in the immune memory IM or replacing the existing:
    - if $\varepsilon_\alpha > \varepsilon_{lim}$, then each μ neuron saves new set of weights W_{m+1} that was used for solving this situation and IM saves the situation X as $S_{m+1} = X$ and $m = m + 1$;

- else if $\varepsilon_\alpha \leq \varepsilon_{\lim}$, then each μ neuron updates set of weights W_α and the record α in the IM is updated $s_\alpha = X$.

STEP 10. END of the algorithm.

Developed algorithm for novel immune neural network for unsupervised collision probability evaluation and minimization for electrical vehicle dangerous situation recognition and prevention task was used:

- To develop a computer model for testing the novel algorithm of immune neural network for unsupervised collision probability evaluation and minimization for electrical vehicle dangerous situation recognition and prevention task (Chapter 4.6).
- To make experimental testing of the proposed novel algorithm of the immune neural network for unsupervised collision probability evaluation and minimization for electrical vehicle dangerous situation recognition and prevention task (Chapter 5.6).

3.7. Chapter 3 summary

In the doctoral research algoritms for different task performing were developed:

1. All the proposed algorithms in the third chapter of the doctoral thesis are multifunctional and can be implemented into control systems of different electrical vehicles type.
2. The proposed algorithm for INN for unsupervised collision probability evaluation and minimization for electric transport dangerous situation recognition and prevention task is a novel algorithm and is foreseen to use for electrical vehicle unsupervised control.
3. All the other algorithms are used as helper methods in case to provide the autonouse safety drive of the electrical vehicles:
   a. Algorithm for traffic light red signal recognition method for electric transport dangerous situation recognition and prevention task;
   b. Algorithm for convolutional neural network for object recognition for electric transport dangerous situation recognition and prevention task;
   c. Algorithm for electrical transport collision probability evaluation task;
   d. Algorithm for neural network for collision probability evaluation and minimization for electric transport dangerous situation recognition and prevention task.

Computer simmulations were made to prove the efficiency of the proposed algorithms. Developed prototypes and computer models are described in the next chapters of the doctoral thesis.

4. DEVELOPED PROTOTYPE AND COMPUTER MODEL FOR TESTING PROPOSED ALGORITHMS

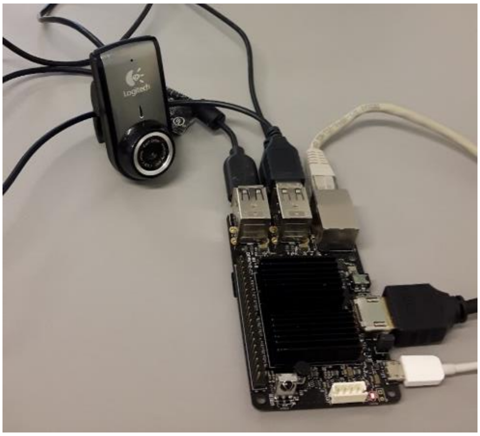
4.1. Situation description

Computer simulation is the process of mathematical modelling, performed on a computer, which is designed to predict the behaviour of the physical system. Computer simulation allows us to check the reliability of chosen mathematical models and algorithms faster, cheaper and safer than real experiments. Computer simulations have become a useful tool for the mathematical modeling of many systems. Wherefore, computer models and prototypes were developed to prove the effieciency of the algorithm, described in the third chapter of the doctoral thesis.

4.2. Computer model for testing the algorithm of traffic light red signal recognition method for electric transport dangerous situation recognition and prevention task

Following attributes were used to develop this computer model:

- Mathematical model for traffic light red signal recognition task (Chapter 2.4);
- Developed algorithm for traffic light red signal recognition method for electric transport dangerous situation recognition and prevention task (Chapter 3.2).



4.1. fig. Hardware of the traffic light indication recognition algorithm testing.

The computer model was developed to prove the efficiency of the proposed algorithm of traffic light red signal recognition method for electric transport dangerous situation recognition and prevention task. Following devices were used for testing:

- multicoreARM CortexV8 64 bit microcontroller, for all calculating making;
- webcamera, for video of the traffic light taking;

- keyboard and mouse for the microcontroller and webcamera control;
- prototype of the traffic light.

[*] cam_rg_auto.cpp

```
int RGBmax = 256;

bool g=0;

int main(int argc, char* argv[])
{

        CvCapture* capture = cvCreateCameraCapture(CV_CAP_ANY); //cvCaptureFromCAM( 0 );
        assert( capture );

        cvSetCaptureProperty(capture, CV_CAP_PROP_FRAME_WIDTH, 1260);//1280);
        cvSetCaptureProperty(capture, CV_CAP_PROP_FRAME_HEIGHT, 960);//960);

        double width = cvGetCaptureProperty(capture, CV_CAP_PROP_FRAME_WIDTH);
        double height = cvGetCaptureProperty(capture, CV_CAP_PROP_FRAME_HEIGHT);
        printf("[i] %.0f x %.0f\n", width, height );


        image = cvQueryFrame( capture );

        cvNamedWindow("original",CV_WINDOW_NORMAL);
        cvNamedWindow("rgb and red",CV_WINDOW_NORMAL);
        cvNamedWindow("rgb and green",CV_WINDOW_NORMAL);
        cvNamedWindow("ROI",CV_WINDOW_NORMAL);

        int roix=553, roiy=298, roix2=573, roiy2=336;
        int roiw=roix2-roix, roih=roiy2-roiy;
```

4.2 fig. Software of the traffic light indication recognition algorithm testing.

Computer model for testing the algorithm of traffic light red signal recognition method for electric transport dangerous situation recognition and prevention task was used:

- To make experimental testing of the proposed algorithm of traffic light red signal recognition method for electric transport dangerous situation recognition and prevention task (Chapter 5.2).

4.3. Computer model for testing the algorithm of convolutional neural network for object recognition for electric transport dangerous situation recognition and prevention task

Following attributes were used to develop this computer model:

- Mathematical model for object recognition task (Chapter 2.5);
- Developed algorithm for convolutional neural network for object recognition for electric transport dangerous situation recognition and prevention task (Chapter 3.3).

Object-oriented programming is effective in implementing ANN models. It was used for the computer model to test the proposed CNN algorithm.

The class of Convolutional neuron consists of:

Parameters:

- Filter size: integer;
- 2D array of weights: real;
- 2D array of inputs: real;
- Bias: real;

- Output value : real.

Methods:

- Constructor method;
- Setting weights;
- Evaluate output;
- Draw the weight pattern.

The class of Pooling neuron consists of:

Parameters:

- Filter size: integer;
- 2D array of inputs: real;
- Output value: real.

Methods:

- Evaluate output.

The class of Full-connected neuron consists of:

Parameters:

- Dynamic 2D array of inputs: real;
- Dynamic 2D array of weights: real;
- Number of inputs: integer;
- Output value: real;
- Error: real.

Methods:

- Constructor method;
- Set input value;
- Evaluate output;
- Activation method;
- Destructor method.

Following initial parameters were used in the computer model, with opportunity to change them according to the task:

4.1 table

Initial parameters of the computer model for testing the algorithm of CNN for object recognition for electric transport dangerous situation recognition and prevention task

Element	**Meaning**
#define SIZE	Spatial extent size
#define K	Feature maps of 1st layer
#define K2	Feature maps of 2nd layer for each 1st layer’s map
#define CL	Output classes
#define RO	Learning rate
#define TR	Training set size

Element	Meaning
#define TEST	Test set size
#define EPOCH_MAX	Maximum training iterations
#define HID 100	Size of hidden layer of FC

After initial parameters are set, image file names are listed, and training set expected clases are provided. The size of the training set depends on the number of expected clases.

Weight patterns are set in the next step. Image convolution and pooling is made further. Convolution and pooling layers are changing alternately according to the proposed algorithm.

Computer model for testing the algorithm of convolutional neural network for object recognition for electric transport dangerous situation recognition and prevention task was used:

- To make experimental testing of the proposed algorithm of convolutional neural network for object recognition for electric transport dangerous situation recognition and prevention task (Chapter 5.3).

4.4. Computer model for testing the algorithm of electrical transport collision probability evaluation task

Following attributes were used to develop this computer model:

- Mathematical model for possible crossing point detection and collision evaluation task (Chapter 2.6);
- Developed algorithm for electrical transport collision probability evaluation task (Chapter 3.4).

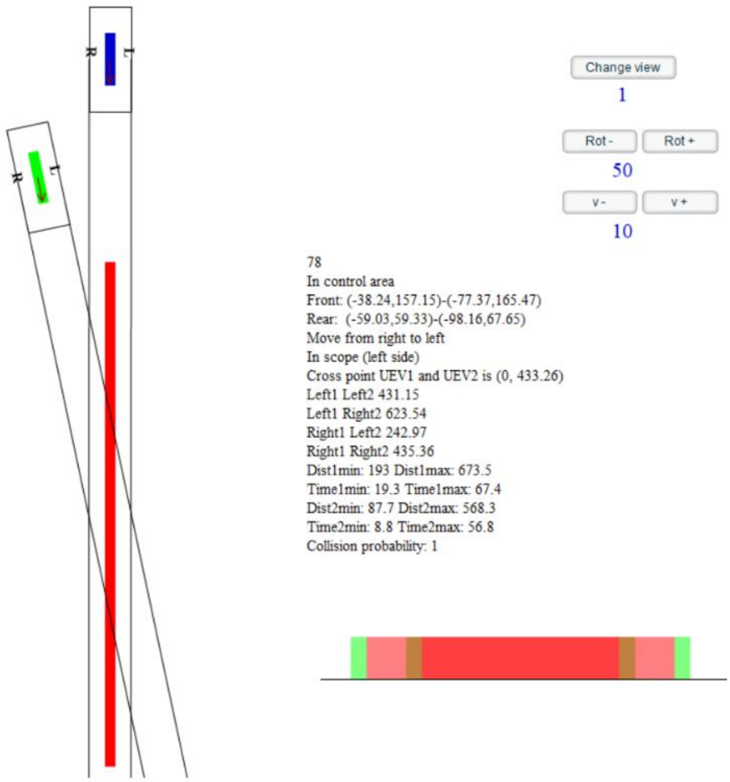
Computer model (4.3 fig.) was developed to prove the efficiency and workability of the proposed algorithm.

Computer model provides an opportunity:

- to choose a point of view by pushing the “Change view” button;
- to change an angle of the movement direction of the vehicle;
- to change speed of the vehicle.

Computer model also provides the data:

- is the object in control area or not;
- coordinates of the objects four angles (right-front, right- rear, left-front, left-rear);
- possible crossing point (as a point);
- possible crossing point as an area (four points);
- minimal and maximal distances till the crossing area;
- minimal and maximal time till the crossing area;
- collision probability.



4.3 fig. Computer model for testing the collision probability calculation algorithm.

Each object was calculating all the parameters and collision probability according to its own location and parameters. That means, that his own coordinates are (0;0), and other vehicles coordinates and crossing point coordinates are calculating comparatively to own coordinates.

The same speed of two objects was used for the computer simulation. The developed computer model also provides the usage of different speeds of the objects.

Parameters for both objects must be defined at the beginning of the experiment. The computer model makes calculations, using formulas that are proposed in developed algorithm.

Computer model for testing the algorithm of electrical transport collision probability evaluation task was used:

- To make experimental testing of the proposed algorithm of electrical transport collision probability evaluation task (Chapter 5.4).

4.5. Computer model for testing the algorithm of neural network for collision probability evaluation and minimization for electric transport dangerous situation recognition and prevention task

Following attributes were used to develop this computer model:

- Mathematical model for the neural network (Chapter 2.7);
- Developed algorithm for neural network for collision probability evaluation and minimization for electric transport dangerous situation recognition and prevention task (Chapter 3.5).

The computer model was developed to test the provided algorithm of multiple UEVs for their collision prevention task. The proposed algorithm can be implemented in different electrical vehicles types, such as train, tram, electro car, quadcopters and others. For this computer model quadcopters were chosen as an object.

Initial parameters are set for all vehicles in the beginning.

4.2 table

Initial parameters of the computer model testing the algorithm of multiple unmanned electrical vehicles for their collision prevention task

Element	Meaning
rotP	Pitch Euler angle
rotR	Roll Euler angle
rotY	Yaw Euler angle
w1, w2, w3, w4	Quadcopters engines speed
F	Traction force
Fz	Force vector on z axis
Fxy	Force vector on xy axis
E	Total energy consumed by the quadcopter's motors
U	Voltage of the quadcopters motor
I	Current of the quadcopters motor
t	Time of flight

All quadcopters have different parameters (longitude, altitude, latitude) of the start point. But all of them have the sampe parameters (longitude, altitude, latitude) of the target point.

Euler angles are calculated for all the objects:

$$rotP = w3 * w3 - w1 * w1 \quad (4.1)$$

$$rotR = w4 * w4 - w2 * w2 \tag{4.2}$$

$$rotY = w1 * w1 + w3 * w3 - w2 * w2 - w4 * w4 \tag{4.3}$$

Traction force of the electrical vehicles is calculated:

$$F = b * (w1 * w1 + w2 * w2 + w3 * w3 + w4 * w4) \tag{4.4}$$

Force vector on z axis is calculated:

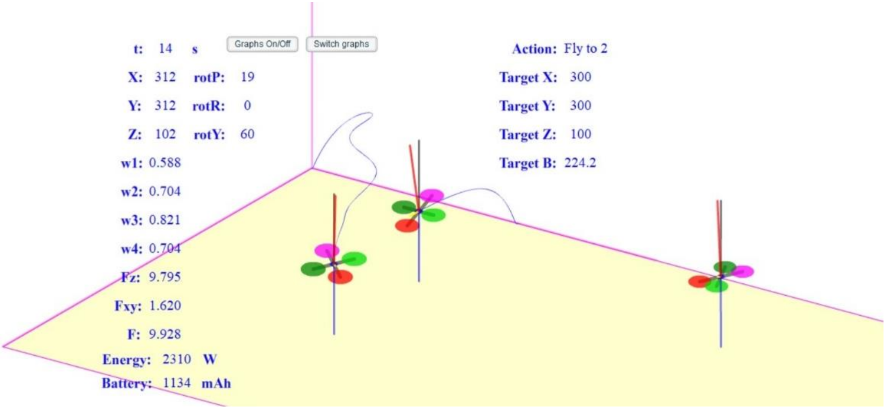
$$Fz = F * \frac{\cos(rotP) + \cos(rotR)}{2} \tag{4.5}$$

Force vector un xy axis is calculated:

$$Fxy = \left| F * \frac{\sin(rotR) + \sin(rotP)}{2} \right| \tag{4.6}$$

Energy consumption is calculated:

$$E = \int_{t_0}^{t_f} \sum_{i=1}^{4} U(t)I(t)dt \tag{4.7}$$



4.4 fig. Computer model for testing the algorithm of multiple unmanned electrical vehicles for their collision prevention task.

Target point needs to be set in the provided computer model. Additionally, model shows parameters for each UEV:

- time (t) of UEV's engine work;
- current coordinates (X,Y,Z);
- Euler angles of motion (rotP – pitch, rotR – roll, rotY – yaw);
- engines speeds (w1, w2, w3, w4);
- force vector on z axis (Fz);

- force vector on xy axis (Fxy);
- traction force (F);
- energy;
- battery capacity.

Computer model for testing the algorithm of neural network for collision probability evaluation and minimization for electric transport dangerous situation recognition and prevention task was used:

- To make experimental testing of the proposed algorithm of neural network for collision probability evaluation and minimization for electric transport dangerous situation recognition and prevention task (Chapter 5.5).

4.6. Computer model for testing the novel algorithm of immune neural network for unsupervised collision probability evaluation and minimization for electrical vehicle dangerous situation recognition and prevention task

Following attributes were used to develop this computer model:

- Mathematical model for the immune neural network (Chapter 2.8);
- Novel developed algorithm for immune neural network for unsupervised collision probability evaluation and minimization for electrical vehicle dangerous situation recognition and prevention task (Chapter 3.6).

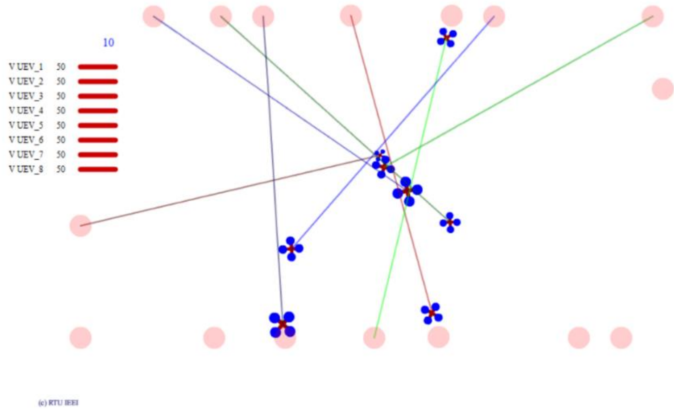
A computer model was developed, to prove the efficiency of the proposed unsupervised vehicle control system. It is implemented in 2D plane, assuming that all UEVs are moving at the same altitude.

In the model, the team of 8 autonomous UEVs of different random size is carrying out the mission in one area and their mission is to patrol the area. Each of UEVs is flying continuously between two target points - from the first point to the second one and back again. The number of UEVs does not change during the simulations.

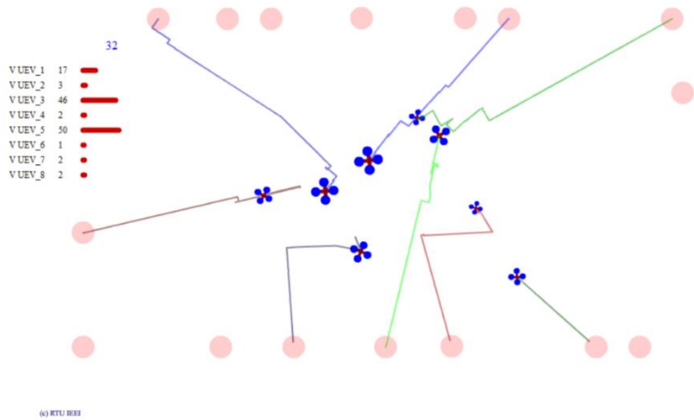
UEVs speed, trajectory, as well as distance till the possible collision point with others UEVs are used as input data. Each UEV calculates other vehicles' movement direction in relation to his own and finds the point of possible collision.

Each UEV calculates the collision probability only with those UEVs, which are in his control area – in front on the right size, in case to minimize number of necessary calculations.

Screenshots of the computer model working are shown in the figure 4.5 and figure 4.6. In figure 4.5 no motion control was used, so UEV are moving by using the shortest way from one point to another, without making a risk assessment and collision prevention. In figure 4.6 INN immune neural network is used for motion control. The proposed unsupervised vehicle control system calculates the collision probability and minimizes it by changing movement parameters change, that is why paths of UEVs are not straight in the figure 4.6.



4.5 fig. Computer model during the simulation process without any motion control.



4.6 fig. Computer model during the simulation process with proposed novel INN.

On the one hand it helps to minimize number of collisions but on the other hand it reduces the number of trips, increases the distance of the trip as well as duration of the trip. That is why the next challenge and goal of the research was to steady the developed model and to find the best parameters for different input data.

There are several parameters, which can be changed in order to get better simulation results:

AP – collision probability weight according to the movement parameters change. The bigger is collision probability weight the smaller is movement parameters change weight;

TrLim – maximal number of iterations for the one decision making process;

PSens – minimal collision probability to make calculations for the collision probability minimization;

AngLim – maximal possible trajectory (angle) change;

SafeDist – maximal distance till other UEV to start crash prevention and to calculate collision probability;

Vnom – nominal speed of the UEVs. All UEVs have the same nominal speed during one simulation.

Database is performing multiple tasks:

- Saving common information about trips: number of trips, duration of trips, distance of trips (4.7 fig).

uev	t	fnr	dist	dur
UEV_5	55.185	2	893.741	32.367
UEV_6	58.317	3	615.349	16.179
UEV_1	76.424	4	751.085	26.44
UEV_4	83.139	3	1055.853	27.689
UEV_8	169.773	13	427.958	8.56
UEV_2	171.989	5	1983.126	69.448

4.7 fig. Database: common results of the computer simulation.

Where:

uev – number of the UEV;

t – time of the trip;

fnr – number of the trip;

dist – distance of the trip;

dur – duration of the trip.

- IM for saving parameters for solving similar situations – situations with small discrepancies (4.8 fig).

uev	sit	xin	wgtv	wgtr	t
UEV_1	0	31	0	0	0.161
UEV_3	0	31	0	0	0.161
UEV_5	0	31	0	0	0.161
UEV_7	0	31	0	0	0.161
UEV_1	1	31	0	0	0.294
UEV_2	0	31	0	0	0.294

4.8 fig. Database: immune memory of the computer simulation.

Where:

uev – number of the UEV;
sit – number of the situation;
xin – input vector;
wgtv – necessary speed changes;
wgtr – necessary trajectory changes;
t – time of simmulation.

- Saving the data about found solutions (4.9 fig.).

uev	sit	fit	err	f	prob	dv	dr	tr	t	best	aver
UEV_1	0	-1	-1	0	0	0	0	1	0.161	1	1
UEV_2	-1	-1	-1	0	-1	1	90	100	0.161	100	100
UEV_3	0	-1	-1	0	-1	0	0	1	0.161	1	1
UEV_4	-1	-1	-1	0	-1	1	90	100	0.161	100	100
UEV_5	0	-1	-1	0	0	0	0	4	0.161	4	4
UEV_6	-1	-1	-1	0	-1	1	90	100	0.161	100	100

4.9 fig. Database: found solutions of the computer simulation.

Where:
uev – number of the UEV;
sit – number of the situation;
err – error;
f – function value;
prob – collision probability;
dv – necessary speed change;
dr – necessary trajectoru (angle) change;
tr – number of iterations;
t – time of the simmulation;
best – best iteration;
aver – average best iteration.

- Saving data about collisions (4.10 fig.).

Nos	T	X	Y	V	D
UEV_1	23.618	433.3	231.6	50	74.72
UEV_7	23.618	442.4	235.9	50	-145.56
UEV_2	24.022	472.8	175.4	50	166.78
UEV_3	24.022	464.05	182.5	50	-76.45
UEV_6	25.979	492.5	84.75	50	131.08
UEV_3	26.023	483.7	88.4	50	-74.64

4.10 fig. Database: detected collisions during the computer simulation.

Where:

uev – number of the UEV;
T – time of the simmulation;
X – x coordinate of the detected collision;
Y – y coordinate of the detected collision;
V – speed of the UEV at the moment of the detected collision;
D – anticollision criteria.

The proposed computer model not only is uded for simmulations, but also helps to receive analytical data for the future collsion prevention decisions, model improvement and weak sides detection.

Computer model for testing the algorithm of novel immune neural network for unsupervised collision probability evaluation and minimization for electrical vehicle dangerous situation recognition and prevention task was used:

- To make experimental testing of the proposed novel algorithm of the immune neural network for unsupervised collision probability evaluation and minimization for electrical vehicle dangerous situation recognition and prevention task (Chapter 5.6).

4.7. Electric scheme with an unsupervised immune memory for unmanned electrical vehicle based on a single board computer

An electrical circuit diagram of the collision prevention device with an unsupervised IM for UEV based on a single board computer was developed. The proposed electrical circuit was developed for the electrical vehicle – quadcopter, but it can be applied for other electyrical vehicles type also, because the developed collision prevention device is multifunctional and can be used with different electrical vehicles. To apply the proposed electrical circuit to other types of electrical vehicles, small changes are needed, for example to change set if sensors and signals, that are sent to the vehicle controller.

Collision prevention device is connected to the existing quadcopter electrical circuit, which in most cases is based on the following depicted basic components:

1. Fly Controller – the diagram refers to the Fly Controller on the quadcopter, its power scheme, sensors, etc.
2. Battery – quadcopter power supply, the output voltage parameters of which depend on the internal circuit of the batteries.
3. ESC1 - ESC4 – electronic speed controller, which controls motors of the quadcopter.
4. M1 - M4 – collectorless three-phase motors.

The Fly Controller is controlled using the following 6 control signals, which it receives from the collision prevention device master controller:

1. Sig1 – signal for the yaw angle of the flight change.
2. Sig2 – signal fot the pitch angle of the flight change.
3. Sig3 – signal for the roll angle of the flight change.
4. Sig4. – signal for the traction force.

5. Sig5 – signal for GPS attitude.
6. Sig6 – signal for emergency signal.

The main controller of the collision prevention device is the selected Odroid C2 single-board computer. It (as well as other collision prevention device components) is powered by Vdd 5V DC voltage, which is obtained from the DC-DC step-down pulse converter circuit based on the LM2576T chip. The circuit is powered by a quadcopter battery.

The choice of the Odroid C2 single-board computer as the main collision prevention device controller is basically determined by the fact that the system is intended to use a Full HD Camera, which is connected to USB1, which is used to take pictures for object and traffic light indication recognition process.

The Ultrasonic sensor group of 6 sensors is connected to the Odroid C2 GIPIO contacts, which are used for objects detection nearby.

An X-bee PRO S5 radio module adapter is connected to the UART1 terminals TXD1 and RXD1, which is equipped with a separate built-in Logic Level Shifter (LLS) and power supply (PS) to 3.3V, as well as a data transfer indicator for LEDs HL1 and HL2. With the help of the radio module, the collision prevention device communicates with the other collision prevention devices, thus receives and transmits data about vehicles speed, coordinates and trajectory of motion.

In addition, the GY 89 Gyro Accelerometer Sensor Module is connected to the Odroid C2, which contains:

a) Gyroscope 3D - chip L3GD20;

b) Accelerometer 3D + magnetometer 3D - chip LSM303D;

c) Barometer 1D - chip BMP180.

Gyroscope is used for measuring or maintaining orientation and angular velocity of the quadcopter.

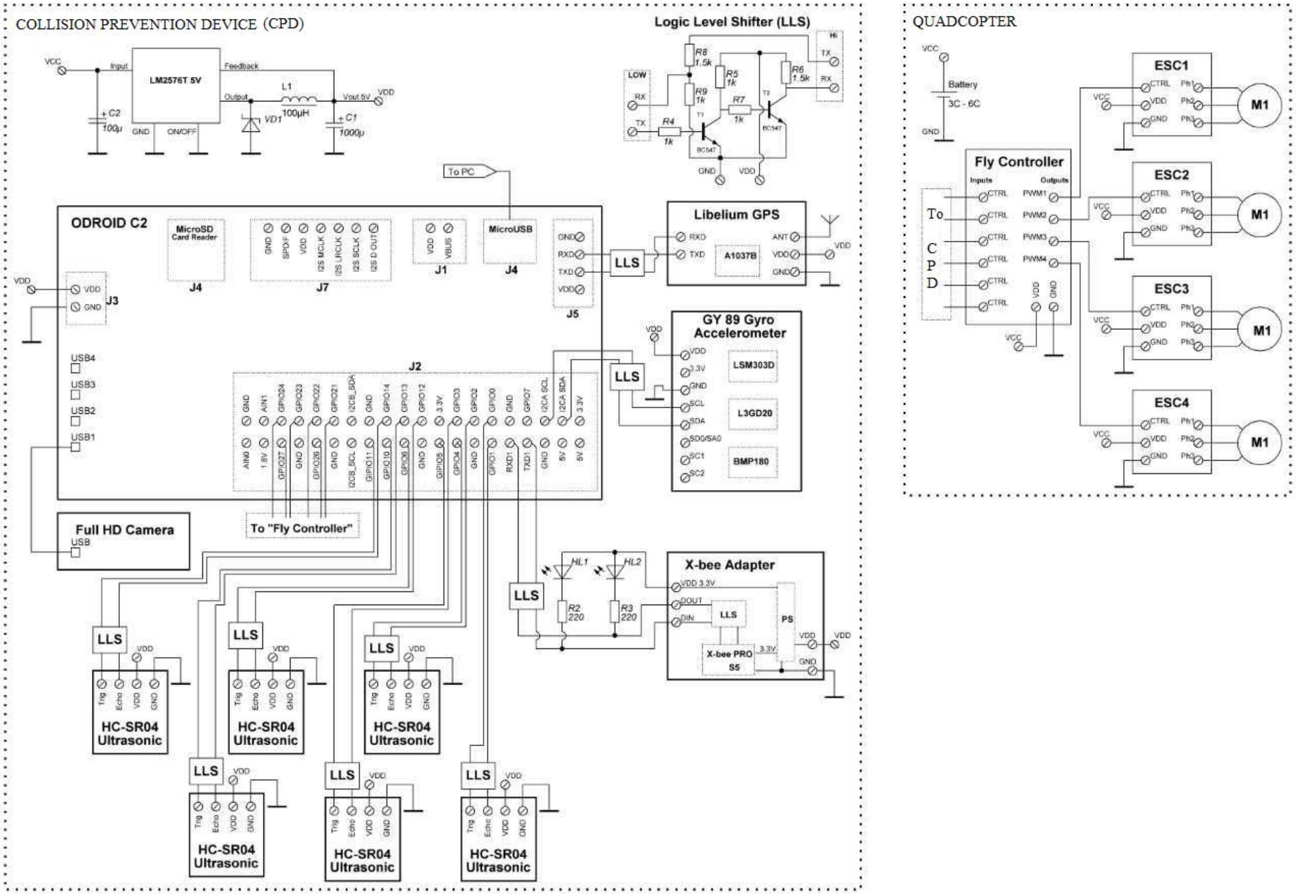
Accelerometer is used for measuring acceleration of the quadcopter.

Barometer is used for for automatic altitude control and automatic landing.

The last of the modules to be connected to the Odroid C2 is the Libelium GPS module with a built-in A1037B chip, which is used to obtain data from the GPS system about quadcopters location.

All external modules are powered by the above-mentioned power supply output voltage VDD 5V DC voltage, as well as for communication with Odroid C2 a Logic Level Shifter circuit is used to adjust the signal voltage levels.

Algorithms, developed and described in this research, can be implemented in the Odroid C2 single-board computer. As a result, Odroid C2 receives the data from all the connected sensors, calculates the collision probability and makes decision about the collision probability minimization by changing the movement parameters of the quadcopter. When the decision is done, 6 signals (signal for the yaw angle of the flight change; signal fot the pitch angle of the flight change; signal for the roll angle of the flight change; signal for the traction force; signal for GPS attitude; signal for emergency signal) are sent to the quadcopters fly control. Fly Controller controls the quadcopter’s 4 motors, according to the received signals.



4.11 fig. Electric scheme.

4.8. Chapter 4 summary

Several computer models and prototypes were developed and described in the fourth chapter of the doctoral thesis, to prove the workability of the developed algorithms:

1. Computer model for testing the algorithm of traffic light red signal recognition method for electric transport dangerous situation recognition and prevention task;
2. Computer model for testing the algorithm of convolutional neural network for object recognition for electric transport dangerous situation recognition and prevention task;
3. Computer model for testing the algorithm of electrical transport collision probability evaluation task;
4. Computer model for testing the algorithm of neural network for collision probability evaluation and minimization for electric transport dangerous situation recognition and prevention task;
5. Computer model for testing the novel algorithm of immune neural network for unsupervised collision probability evaluation and minimization for electrical vehicle dangerous situation recognition and prevention task.

Object-oriented programming was used for developing computer models. Database was developed additionally for saving the results of the computer simulations.

Electric scheme with an unsupervised immune memory for unmanned electrical vehicle based on a single board computer was also developed and described in fourth chapter of the doctoral thesis. The proposed electrical circuit was developed for the electrical vehicle – quadcopter, but it can be applied for other electrical vehicles type also, because the developed collision prevention device is multifunctional and can be used with different types of electrical vehicles.

Results of the computer simulations are proposed in the next part of the doctoral thesis.

5. EXPERIMENTAL TESTING OF THE PROPOSED ALGORITHMS

5.1. Situation description

Experimental testing of the developed algorithms, based on the computer models, is described in this part of the doctoral thesis. Previously developed computer models and protypes are used to prove the workability of the proposed algorithms.

5.2. Experimental testing of the proposed algorithm of traffic light red signal recognition method for electric transport dangerous situation recognition and prevention task

Following attributes were used for this experiment:

- Mathematical model for traffic light red signal recognition task (Chapter 2.4);
- Developed algorithm for traffic light red signal recognition method for electric transport dangerous situation recognition and prevention task (Chapter 3.2);
- Computer model for testing the algorithm of traffic light red signal recognition method for electric transport dangerous situation recognition and prevention task (Chapter 4.2).

5.1 fig. Traffic light signal recognition experiment.

Real time recognition experiment was made by using the traffic light prototype (5.1 fig.). The proposed system is trained to distinguish the red signal from other color signals without any mistakes.

5.3. Experimental testing of the proposed algorithm of convolutional neural network for object recognition for electric transport dangerous situation recognition and prevention task

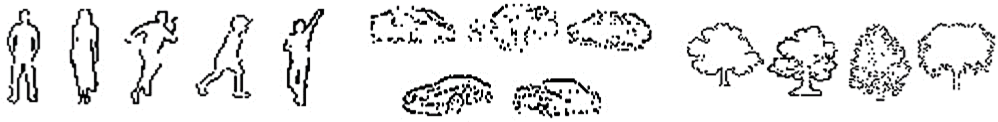
Following attributes were used for this experiment:

- Mathematical model for object recognition task (Chapter 2.5);

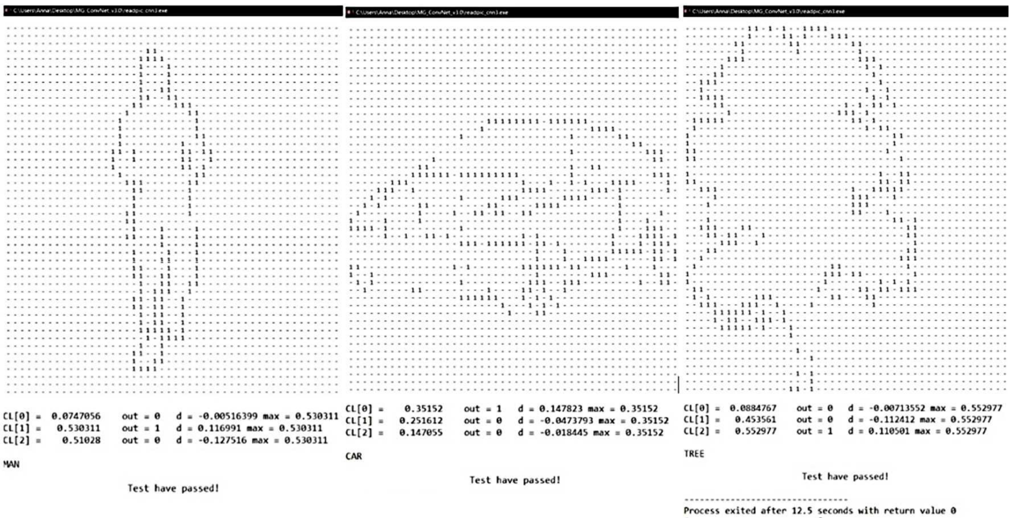
- Developed algorithm for convolutional neural network for object recognition for electric transport dangerous situation recognition and prevention task (Chapter 3.3);
- Computer model for testing the algorithm of convolutional neural network for object recognition for electric transport dangerous situation recognition and prevention task (Chapter 4.3).

5.3.1. Experiment Nr.1

The first experiment was made to verify usefulness of CNN in object recognition tasks. Before the object can be recognised, the picture needs to be processed using filter applications and picture resizing. In this example, images are processed manually. Then CNN was trained to recognize objects, such as humans, cars and trees.



5.2 fig. CNN object recognition training set.



5.3 fig. Result of CNN object recognition experiment.

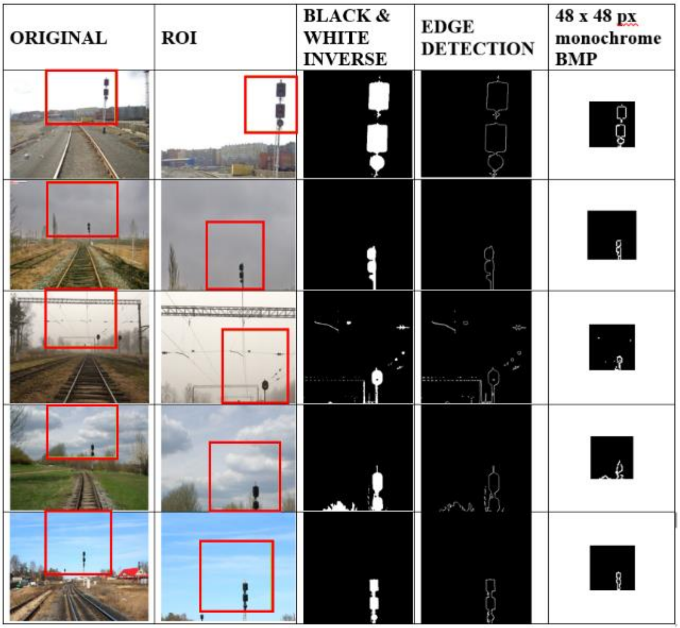
Training sets were used: set of 5 different silhouettes of humans; set of 5 silhouettes of cars and set of 4 silhouettes of trees (5.2 fig.) to train neural network to recognise objects.

The biggest number of railway accidents is happening because of collision with humans or cars. That is why for the experiment pictures of them were taken. The third type of objects – “trees” was used in recognizing process for making sure that CNN works correctly and is able to distinguish different types of objects.

It was asked to recognise 3 pictures, that differ from the pictures given in the training set, after CNN was trained. CNN training took 12.5 seconds (5.3 fig.).

5.3.2. Experiment Nr.2

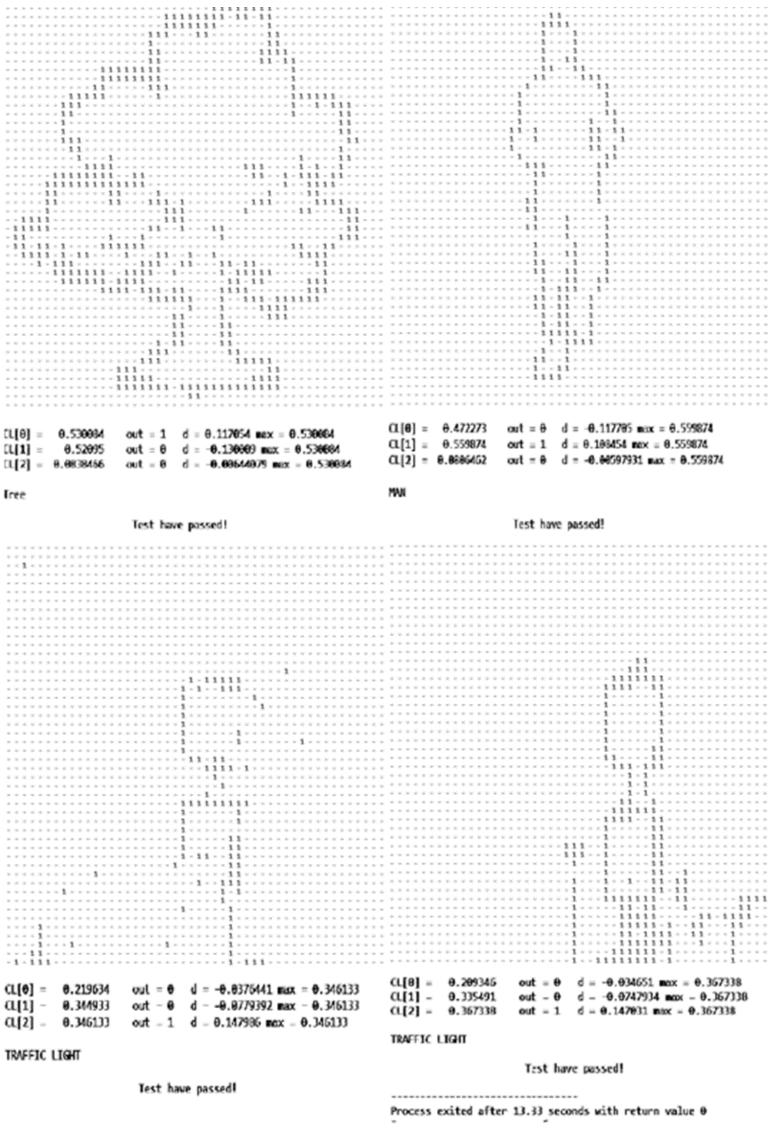
The training set and test set needs to be defined before making a computer model and checking the proposed CNN algorithm in action. ROI region of interests was processed and size reduced till 48x48 pix (5.4 fig.) for this purpose.



5.4 fig. Traffic light recognition training set.

A test set of pictures of traffic light, men and trees was given to the input, after CNN was trained to recognize traffic lights.

As a result, CNN recognized all the traffic lights correctly (5.5 fig.). The pictures of other objects, like trees and human beings also were used in computer experiment, to find out either the CNN could distinguish traffic lights from other objects or not.



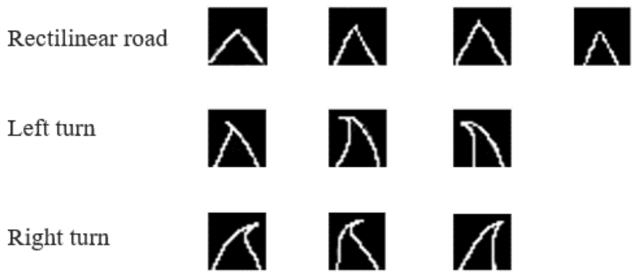
5.5 fig. Traffic light recognition experiment.

Untrained CNN recognize the traffic lights in 13.83 seconds. The training process took the biggest amount of time. After training is done, recognition process take less than a second. The positive result of the experiment proves the efficiency of the algorithm.

5.3.3. Experiment Nr.3

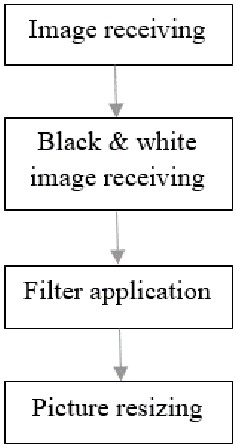
System is developed for the electric vehicles and, for the experiments, pictures of the tram or train rails were taken. It is enough to recognize the direction of rails or lines on the sides of the road for recognizing road direction. A picture of the road or rails for the proposed system can be taken by camera on the vehicle. Developed system can recognize a different colour object on the black background, that is why it is necessary to use filters and to lead the received picture to the appropriate form.

Three training samples with 12x12 pixels size were taken for the left and right turns and four training samples were taken for the rectilinear road (5.6 fig.).



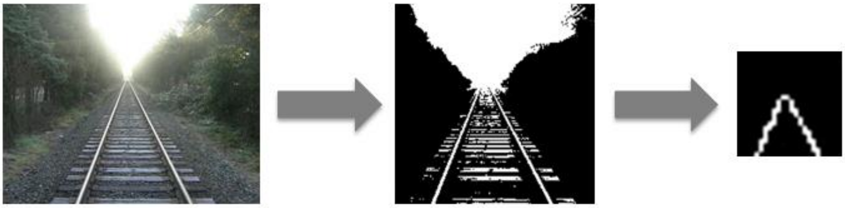
5.6 fig. Set of the training sample.

The picture must be processed as it is shown in figure 5.7 before the road direction can be recognized on the received picture. Received picture must be reformatted because the proposed CNN is processing BMP pictures only. The size of the picture must be resized for faster operation process.



5.7 fig. Image preparation.

Example of the image preparation is shown in the figure 5.8. Filters and formatting process were made manually for the experiment. In the next research authors plan to add a block, that could automatize a filter application and picture resizing processes, into the developed algorithm.



5.8 fig. Image preparation example.

The road direction recognition process was started after image preparation and CNN training process was done.

Three images with different road directions were taken for the experiment. All the figures differ from the images, which were taken for the training set.

The result shows that CNN was trained and recognized all the road turns correctly in 5.697 seconds (5.9 fig.). CNN must be trained before recognition process for making this process faster.

```
C:\Users\Anna\Desktop\MG_ConvNet_v1.0\readpic_cnn3.exe

Training successfully finished!

Epoch = 325 Sample = 10

tr = 10
 - - - - - - - - - - - -
 - - - - - - - - - - - -
 - - - - - - - - - - - -
 - - - - - 1 - - - - - -
 - - - - 1 1 - - - - - -
 - - - - 1 - 1 - - - - -
 - - - 1 - - 1 - - - - -
 - - - 1 - - - 1 - - - -
 - - 1 1 - - - 1 - - - -
 - - 1 - - - - - 1 - - -
 - - 1 - - - - - - 1 - -
 - 1 - - - - - - - 1 - -

CL[0] =   0.468437    out = 1    d = 0.132361 max = 0.468437
CL[1] =   0.134125    out = 0    d = -0.0155766 max = 0.468437
CL[2] =   0.410597    out = 0    d = -0.0993675 max = 0.468437

RECTILINEAR road

                       Test have passed!

 Epoch = 325 Sample = 11

 tr = 11
  - - - - - - - - - - - -
  - - - - - - - - - - - -
  - - - 1 1 - - - - - - -
  - - - - 1 - - - - - - -
  - - - - 1 1 - - - - - -
  - - - - 1 - 1 - - - - -
  - - - 1 - - - 1 - - - -
  - - - 1 - - - - 1 - - -
  - - 1 - - - - - 1 - - -
  - - 1 - - - - - - 1 - -
  - - 1 - - - - - - 1 - -
  - 1 - - - - - - - - 1 -

 CL[0] =   0.209766    out = 0    d = -0.0347718 max = 0.427567
 CL[1] =   0.427567    out = 1    d = 0.140105 max = 0.427567
 CL[2] =    0.36662    out = 0    d = -0.0851328 max = 0.427567

 LEFT turn

                       Test have passed!
```

```
Epoch = 325 Sample = 12

tr = 12
 - - - - - - - - - - - -
 - - - - - - - - - - - -
 - - - - - - - - 1 1 1 -
 - - - - - - 1 1 1 1 - -
 - - - - - 1 1 1 1 - - -
 - - - - 1 1 - 1 - - - -
 - - - 1 1 - - - 1 - - -
 - - - 1 - - - - 1 - - -
 - - 1 - - - - - 1 1 - -
 - - 1 - - - - - - 1 - -
 - 1 - - - - - - - - 1 -
 1 1 - - - - - - - - 1 -

CL[0] =   0.576532    out = 0   d = -0.140756 max = 0.592726
CL[1] =  0.0407994    out = 0   d = -0.00159668 max = 0.592726
CL[2] =   0.592726    out = 1   d = 0.0983167 max = 0.592726

RIGHT turn

                       Test have passed!

--------------------------------
Process exited after 5.697 seconds with return value 0
```

5.9 fig. Result of the experiment.

5.3.4. Experiment Nr.4

Following system work principle was taken for the experiment:

STEP 0. UAV unmanned aerial vehicle is located on the locomotive. The UAV charger is also located there.

STEP 1. UAV starts moving above the train roof and finds the last train wagon by using CNN.

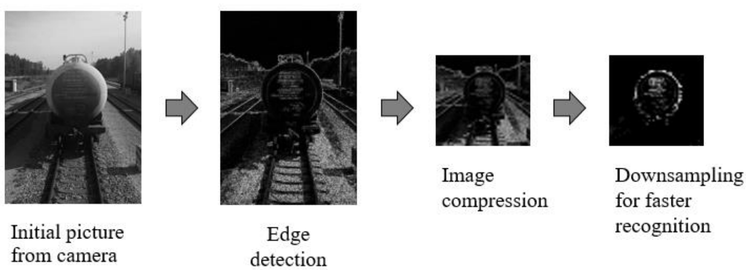
STEP 2. UAV finds objects on the rails by using CNN, and finds the distance till the object by using distance meter.

STEP 3. UAV sends the information to the locomotives cabin by using control components.

STEP 4. All necessary calculations are made by locomotives DTR device. If the recognized object is wagon, then the necessary speed is calculated according to the distance till the wagon. If the object is something else, like human-being or a car, then train is stopped.

STEP 5. UAV components control the existence of objects on the rails and distance till them during all the process convergence of wagons.

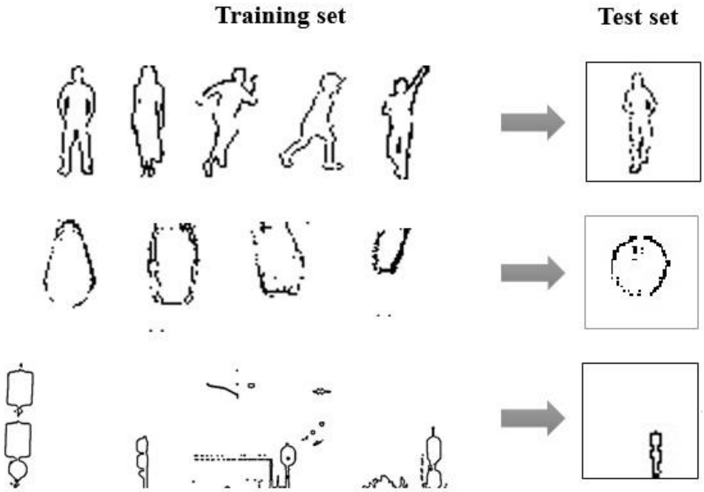
STEP 6. After the process is done, the UAV goes back to its location.



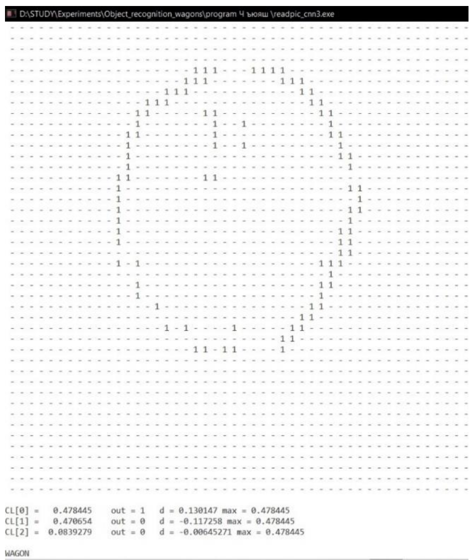
5.10 fig. Picture processing before object recognition.

Pictures of wagons were taken for the experiment, because the system is developed for the wagon recognition task. The system also needs to distinguish wagons from other objects, sets of traffic lights pictures and also human beings were taken for this purpose.

A picture of wagons or other objects can be taken for the proposed system by UAV camera. The developed system can recognize a different colour object on the black background, that is why it is necessary to use filters and to lead the received picture to the appropriate form (5.10 fig.).



5.11 fig. Training and test sets of the experiment.



5.12 fig. Wagon recognition experiment.

Three training sets and three test sets with 48x48 pixels size samples (5.11 fig.) were taken for the computer experiment in case to prove the efficiency of the proposed algorithm.

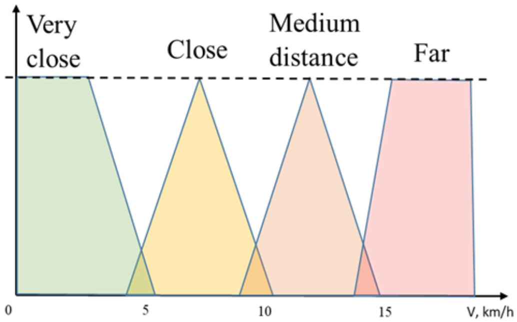
One of the goals of the experiment was to recognise wagon between other objects, such as traffic light or human being. At the result all objects of the test set were recognised correctly by the proposed CNN algorithm, wagon also (5.12 fig.).

It is necessary to find out the distance till the wagon and to calculate necessary speed, after the wagon was recognized. Following algorithm was proposed:

STEP 1. Data obtaining: class of the object O, distance S till the object.

STEP 2. If object O is not a wagon, then speed is reduced to zero.

STEP 3. If object O is a wagon, then speed is calculated according the distance till the object by using Fuzzy Logic (5.13 fig.).



5.13 fig. Fuzzy logic in distance recognition task.

Also, the algorithm for the choice of the necessary speed, according to the distance till the wagon or group of wagons, was proposed.

5.1 table

Train speed accordance to the distance till the wagon group

Speed	**Distance, m**	**Distance, wagons**
>15 km/h	28 m	2
15-10 km/h	14 m	1
10-5 km/h	7 m	0.5
5-3 km/h	3.5 m	0.25

As the result, wagons were recognized by CNN based algorithm. Then risk assessment was made by using fuzzy logic. The distance till the wagon or group of wagons can be determinated by using sensors and as the result necessary speed is found out according to the found distance.

5.4. Experimental testing of the proposed algorithm of electrical transport collision probability evaluation task

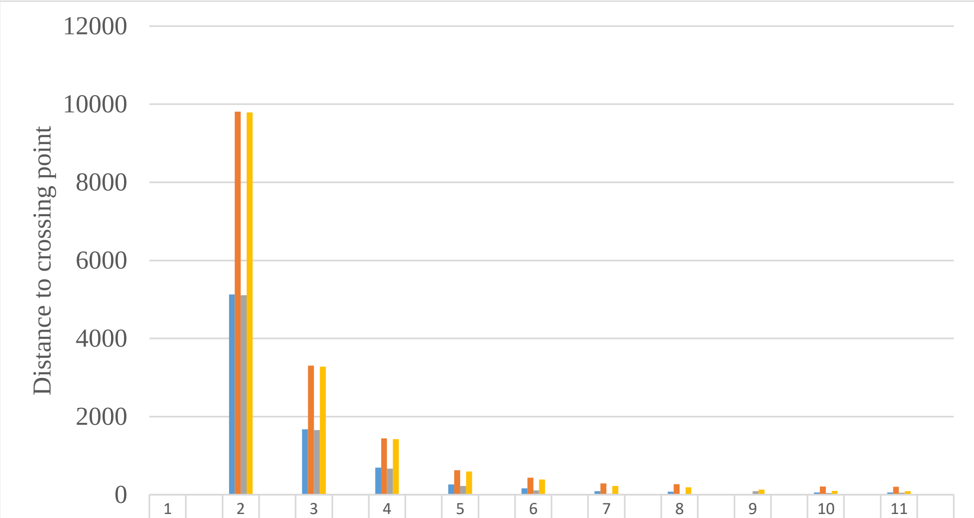
Following attributes were used for this experiment:

- Mathematical model for possible crossing point detection and collision evaluation task (Chapter 2.6);
- Developed algorithm for electrical transport collision probability evaluation task (Chapter 3.4);
- Computer model for testing the algorithm of electrical transport collision probability evaluation task (Chapter 4.4).

Results of the computer simulation are shown in the table 5.2. Each object was calculating all the parameters and collision probability according to own location and parameters. That means, that his own coordinates are (0;0), and other vehicles coordinates and crossing point coordinates are calculating comparatively to own coordinates.

The same speed of two objects was used, for the computer simulation. The developed computer model also provides the usage of different speeds of the objects.

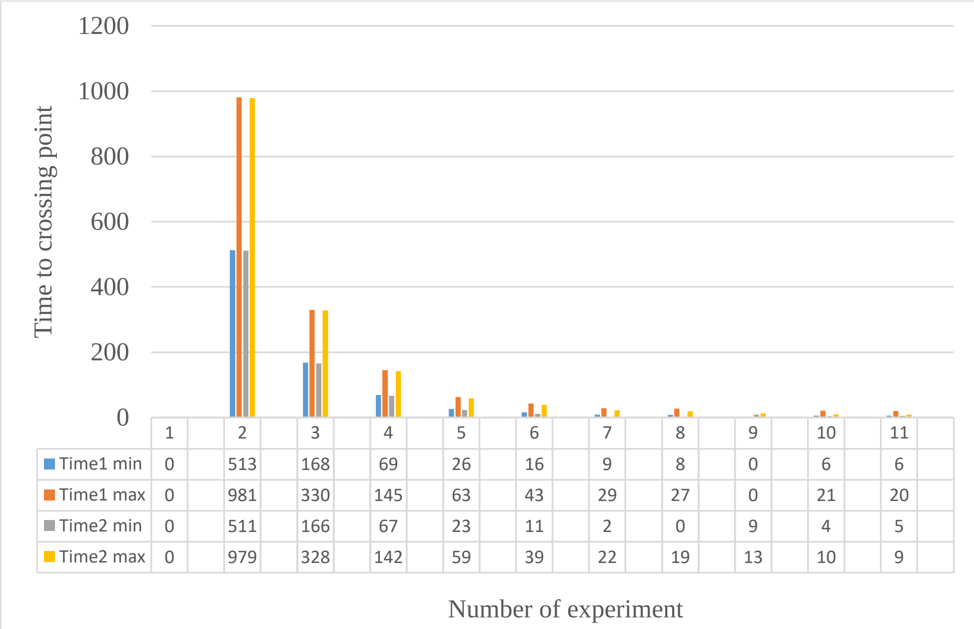
The comparison of the distance till the crossing point, detected during the computer simulations, is shown in the figure 5.14. The comparison of the available time for the reaction, detected during the computer simulations, is shown in the figure 5.15.



	1	2	3	4	5	6	7	8	9	10	11
Dist1 min	0	5126	1676	692	262	161	90	79	0	59	60
Dist1 max	0	9809	3304	1446	630	434	292	267	0	213	200
Dist 2 min	0	5107	1655	666	225	113	21	2	93	43	52
Dist2 max	0	9791	3282	1420	592	386	223	189	127	99	88

Number of experiment

5.14 fig. Object 1 distance till the crossing point.



	1	2	3	4	5	6	7	8	9	10	11
Time1 min	0	513	168	69	26	16	9	8	0	6	6
Time1 max	0	981	330	145	63	43	29	27	0	21	20
Time2 min	0	511	166	67	23	11	2	0	9	4	5
Time2 max	0	979	328	142	59	39	22	19	13	10	9

5.15 fig. Object 1 available time for the reaction.

As the results of the computer experiment show, the collision probability depends on the distance till the crossing point, available time for the reaction and vehicles' speed. For example, during computer simulation Nr. 1 no crossing points were detected, so the collision probability is equal zero. During the computer experiment Nr. 2, distance till the crossing point for the object 1 was from 5126 till 9809, but for the object 2 was from 5107 to 9791. Therefore, object 2 was also moving towards the crossing point with the same speed, that is why detected collision probability is 1.

During the computer experiment Nr. 9, object 1 didn't have distance till the crossing point, because it already was placed on the crossing point. It also didn't have any available time for the reaction. In difference with object 1, object 2 had available time for the reaction from 9 to 13 seconds. This is sufficient amount of time for object 1 to leave the crossing point with a specified speed.

5.2 table

Results of the computer simulation

Nr	Object 2 front angles		Object 2 rear angles		Cross. Point				Distance till the crossing point				Time till the crossing point				Collision prob.
	Left	Right	Left	Right	Left1 Left2	Left1 Right2	Right1 Left2	Right1 Right2	Dist1 min	Dist1 max	Dist 2min	Dist2 max	Time1 min	Time1 max	Time2 min	Time2 max	
1	-110; 68	-150; 68	- 110; -32	-150; -32	-	-	-	-	-	-	-	-	-	-	-	-	0
2	-110; 70	-149; 70	- 111; -30	-150; -30	7467	9759	5176	7468	5126	9809	5107	9791	513	981	511	979	1
3	-107; 73	-147; 75	- 112; -26	-152; -24	2489	3254	1726	2490	1676	3304	1655	3282	168	330	166	328	1
4	-101; 81	-141; 86	- 113; -19	-153; -14	1068	1396	742	1070	692	1446	666	1420	69	145	67	142	1
5	-86; 97	-124; 109	- 115; 1	-153; 13	443	580	312	449	262	630	225	592	26	63	23	59	1
6	-69, 109	-105; 127	- 113; 19	-149; 37	293	384	211	302	161	434	113	386	16	43	11	39	1
7	-34; 125	-64; 152	- 103; 52	-132; 79	183	242	140	199	90	292	21	223	9	29	2	22	1
8	-21; 128	-47; 158	-97; 62	-123; 92	164	217	129	182	79	267	2	189	8	27	-	19	1
9	26; 127	12; 164	-68; 93	-82; 130	125	167	110	153	-	-	93	127	-	-	9	13	0
10	33; 125	20; 164	-63; 96	-75; 134	121	163	109	151	59	213	43	99	6	21	4	10	1
11	67; 110	68; 150	-32; 110	-32; 150	110	150	110	150	60	200	52	88	6	20	5	9	1

5.5. Experimental testing of the proposed algorithm of neural network for collision probability evaluation and minimization for electric transport dangerous situation recognition and prevention task

Following attributes were used for this experiment:

- Mathematical model for the neural network (Chapter 2.7);
- Developed algorithm for neural network for collision probability evaluation and minimization for electric transport dangerous situation recognition and prevention task (Chapter 3.5);
- Computer model for testing the algorithm of neural network for collision probability evaluation and minimization for electric transport dangerous situation recognition and prevention task (Chapter 4.5).

5.5.1. Neural network experiment with training

Target function was used for the experiment, but neural network is trained to make a decision about the speed change, to prevent the collision between vehicles.

If the common number of vehicles is n, then neural network has paired input number 2n. Neural network input and output n amount is dynamic, because the number of vehicles can be changed.

Such a situation was chosen for the experiment: one train; one bus; trajectories of the train and bus have crossing point.

In this situation neural network consists of 4 input and 2 output layers.

Parameters taken for the training set are given in the 5.3 table.

Each element from the set is sent to the neural network input layer during training process. When changes of speed $\Delta v1$ and $\Delta v2$ for the train and bus are found, these values are evaluated by target function.

Neural network is already trained if:

- Collision possibility is reduced minimum to 0.005;
- Average speed change for the train is not bigger than 3 kmh.

5.3 table

Neural network training set

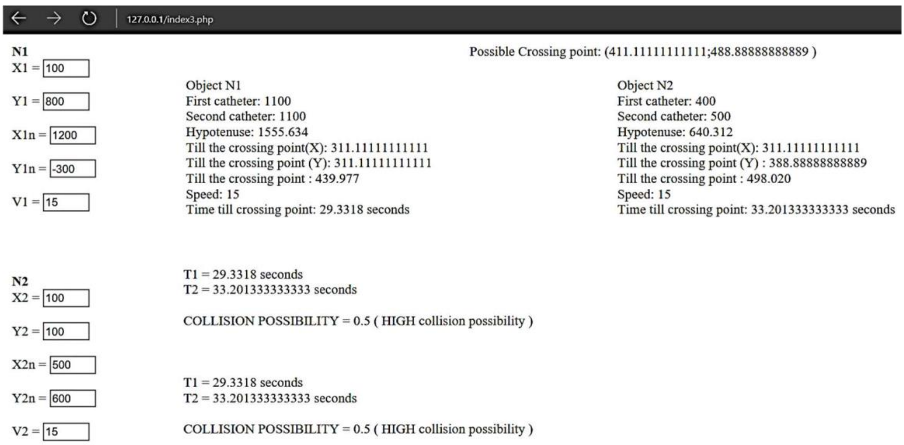
Number of the training set element	**Bus distance till the trajectories crossing point**	**Train distance till the trajectories crossing point**	**Bus average speed**	**Train average speed**
1	400	600	40	60
2	390	580	40	60
3	320	550	40	65
4	290	540	40	70

Number of the training set element	Bus distance till the trajectories crossing point	Train distance till the trajectories crossing point	Bus average speed	Train average speed
5	260	520	40	75
6	230	500	40	80
7	210	480	40	85
8	200	460	40	85
9	190	440	40	90
10	175	420	40	90

Two solutions x_{min}= -10, x_{max} = 0 un x_{min} = -15, x_{max} = 5 were used for each element of the training sets.

5.5.2. Neural network self-training experiment

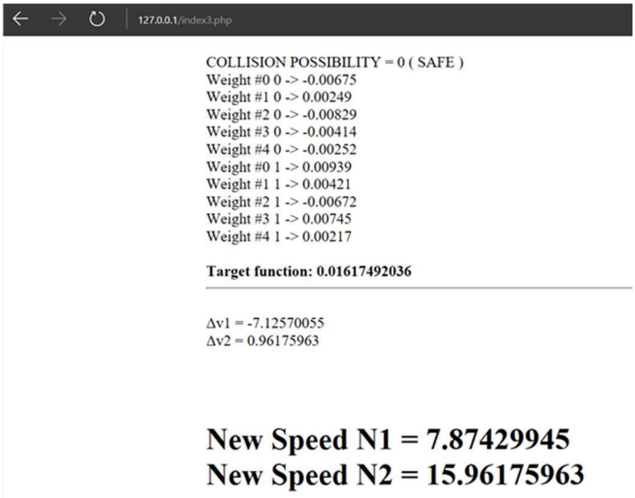
The experiment was based on the danger level estimation. For this purpose, coordinates X1, Y1, X1n, Y1n, X2, Y2, X2n, Y2n and speed V1, V2 of two objects N1 and N2 were entered.



5.16 fig. Request and collision possibility computation.

The proposed system calculates possible crossing point and collision prabability CP. In figure 5.16 is shown that possible crossing point of two objects N1 and N2 is point with coordinates (411.111; 488.889). The distance till the crossing point takes 29.332 sec for the first object and 33.201 sec for the second object. Collision probability is equal to 0.5. According to the task it is a high collision probability, and for reducing it, speeds must be changed.

As collision probability was high, system tried to change the speed of objects and to minimize collision probability. In figure 5.17 collision probability was reduced to zero by decreasing speed of the first object by 7 km/h and increasing speed of the second object by less than 1 km/h.



5.17 fig. Minimized collision possibility and objects speed change.

Experiments show that traditional neural network can be useful in danger level estimation tasks.

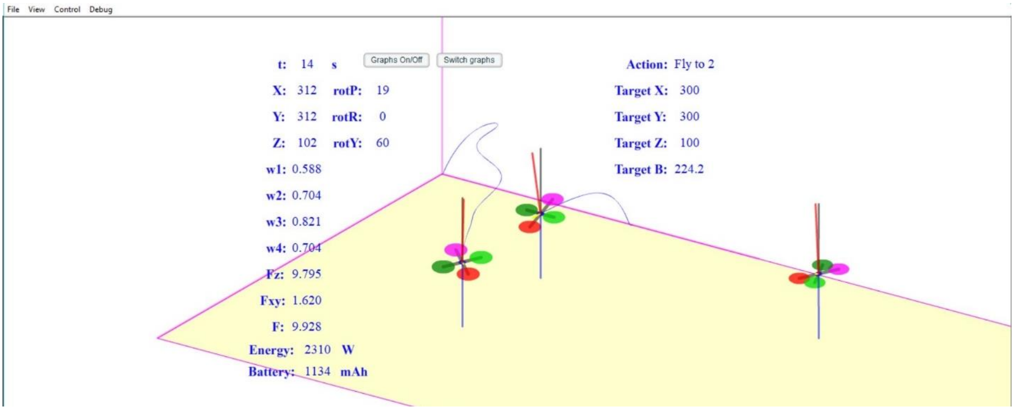
5.5.3. Experimental testing of the algorithm of multiple unmanned electrical vehicles for their collision prevention task

Computer experiments of the developed system of implemented algorithm were made.

The main idea of the experiment was to set the same coordinates of the target point for three different UAV unmanned aerial vehicles and to make sure, that proposed algorithm is working correctly and UAVs won't collide.

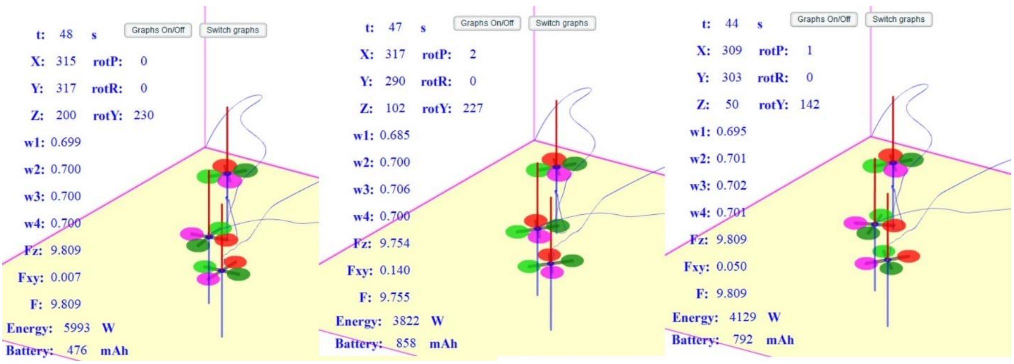
For the experiment target point (300; 300; 100) was chosen. Additionally, program shows parameters for each UAV:

- time (t) of UAV's engine work;
- current coordinates (X,Y,Z);
- Euler angles of motion (rotP – pitch, rotR – roll, rotY – yaw);
- engines speeds (w1, w2, w3, w4);
- force vector on z axis (Fz);
- force vector on xy axis (Fxy);
- traction force (F);
- energy;
- battery capacity.



5.18 fig. Start of the experiment of multiple UAV collision prevention algorithm.

Start of the experiment is shown in the figure 5.18. Three UAVs have different start points and each of them starts to move towards coordinates of the target point. After they are close enough to each other, they start to change height in case of prevention collision.



5.19 fig. UAV reaching the target point coordinates without collide.

In figure 5.19 the result of the experiment is shown. Three UAV reached the target point coordinates without collide and each of them has changed their height according to the developed algorithm. The first UAV took up a position at the height of 200, the second one is almost on the target height, but the third one is descended to the height 50. As the result, collision probability was reduced to zero by changing only one target coordinate Z - height.

5.6. Experimental testing of the proposed novel algorithm of the immune neural network for unsupervised collision probability evaluation and minimization for electrical vehicle dangerous situation recognition and prevention task

Following attributes were used for this experiment:

- Mathematical model for the immune neural network (Chapter 2.8);
- Novel developed algorithm for immune neural network for unsupervised collision probability evaluation and minimization for electrical vehicle dangerous situation recognition and prevention task (Chapter 3.6);
- Computer model for testing the developed novel algorithm of immune neural network for unsupervised collision probability evaluation and minimization for electrical vehicle dangerous situation recognition and prevention task (Chapter 4.6).

5.6.1. Experiment Nr.1

The real part of city transport system was taken as a model for computer experiment (5.20 fig.). The conditions for experiment are following: four routes are selected; two types of UEV transport, road (trolleybuses) and rail (trams), are selected in pairs in opposite direction; one cross point is defined; 40 km/h is defined as nominal UEV speed, speed is allowed to increase up to 60 km/h to avoid collision. Collision probability changes from 0 to 1, maximal value of target function is set as 0.04, motion speed criteria is defined as following - nominal speed minus speed difference, divided by nominal speed.

5.20 fig. Road intersection computer model.

The inputs for INN are motion speed of all UEV's and their distance to crossing point, including own for each UEV. According to this data, each UEV trains its own INN to get speed change satisfying the target function. The decision to accelerate or brake is adjustable by specific collision sensitivity index. The first set of weights appropriate to target function is taken to memory pool, at the beginning of self-training algorithm. The number of iterations is limited to 200, to find the optimal speed change decision. If no result, speed is decreased in double. All UEV collision contacts are recorded in array.

The INN calculates the necessary speed change, however each UEV has acceleration and deceleration rate, that does not allow to change the speed immediately.

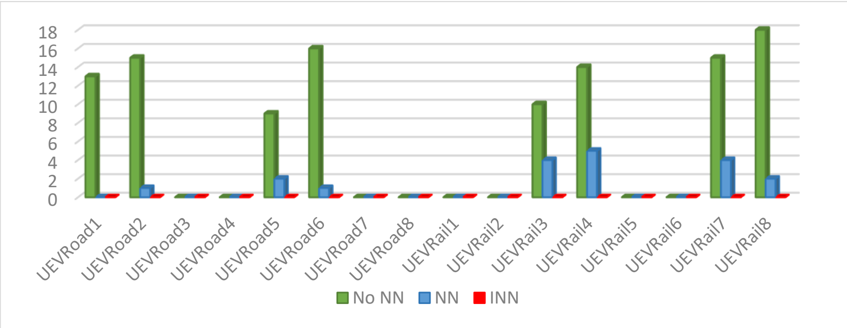
Three simulations with the same experimental conditions were made to prove the efficiency of the developed algorithm.

Experiment Nr.1 – no motion control is used. 110 collisions are detected for 30 minutes simulation.

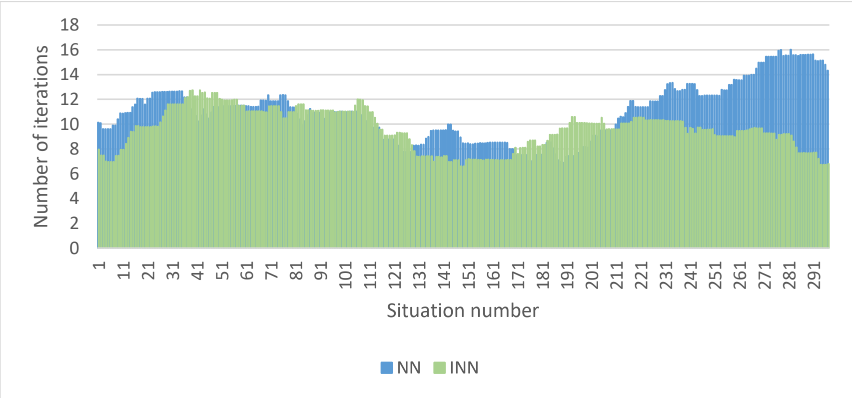
Experiment Nr.2 – neural network is used for motion control. IM is not applicable. Each vehicle uses random weights for the target function minimization. 19 collisions are detected for 30 minutes simulation.

Experiment Nr.3 – proposed INN with IM is used for motion control. No collisions are detected for 30 minutes simulation.

The comparison of the number of collisions in different experiments is shown in figure 5.21.



5.21 fig. Comparison of collision number in 3 different experiments.



5.22 fig. Comparison of number of iterations, while finding the better result.

Another parameter to compare ordinary neural network ANN and INN is a number of iterations for self-training to obtain target function satisfying decision.

The comparison of number of iterations for the one vehicle is shown in figure 5.22. The number of iterations for both structures are approximately the same at the beginning of experiment, because at the beginning of simulation INN is untrained and not much data of the best solutions are in the IM. INN finds out better solutions and update the IM, when training is done and IM is full enough with good examples.

The results show that ANN needs more iterations than INN to solve the same task. This significantly increases processing time and leads to a bigger number of collisions. This explains the number of fixed accidents by using ANN in figure 5.21.

5.6.2. Experiment Nr.2

Computer simulations were made to prove the efficiency of the proposed novel INN and developed computer model. Each computer simulation was made for the group of 8 quadcopters and was 10 minutes long. Each quadcopter has its own size and speed, but these parameters were not changed during the simulations. Quadcopters were able to change the trajectory of motion (XY coordinates) or to change their speed for collision prevention.

The use of 3 types of transport control were compared: No motion control; Traditional ANN was used for motion control; The proposed novel INN was used for motion control.

The proposed model simulates the behavior of real vehicles, that is why their decisions and output data differ.

Parameters of computer simulations are mentioned in 5.4 table, but output data is mentioned in 5.5 table.

5.4 table

Parameters of the computer simulation

Parameter	**Meaning**
XYerr	XY position error -XYerr < err < + +XYerr
Verr	speed error -Verr < err < + +Verr
Rerr	rotation error -Rerr < err < + +Rerr
TXdelay	data transmission delay, ms
Sdelay	own positioning data refresh delay, ms
AreaRate	safety zone area rate proportional to the size
Ap	collision probability weight according to the movement parameters change. The bigger is collision probability weight the smaller is movement parameters change weight
Av	speed change weight Av = (1-Ap)/2
Ar	trajectory change weight Ar = (1-Ap)/2
TrLim	maximal number of iterations for the one decision making process
Sens	collision sensibility

Parameter	Meaning
Psens	probability sensitivity
Vlim	maximal speed
Anglim	maximal possible trajectory (angle) change
Safedist	maximal distance till other UEV to start crash prevention and to calculate collision probability

5.5 table

Computer simulation output data

Parameter	Meaning
AvDur	Average duration of one flight
SmDur	The shortest duration of the flight
LngDur	The longest duration of the flight
AvDist	Average distance of one flight
SmDist	The smallest distance of the flight
LngDist	The biggest distance of the flight
Trips	Number of complete trips/flights during simulation
Collisions	Number of collisions during simulation

1. Results of simulations without motion control

   The first computer simulation was made without any motion control. All the objects were moving straight from one point to another and back again during the simmulation.

5.6 table

Output data

Simulation Nr.	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
1	11	8	19	569	420	888	419	60
2	11	8	19	569	420	888	419	60
3	11	8	19	569	420	888	419	60
4	11	8	19	569	420	888	419	60
5	11	8	19	569	420	888	419	60
AVG	**11**	**8**	**19**	**569**	**420**	**888**	**419**	**60**

Results of simulation:

- The results of computer simulation without any motion control show that 8 quadcopters done in average 419 trips and also 60 collision are detected during 10 minutes (5.6 tab.).

2. Comparison between the results of traditional ANN and the proposed novel INN
    2.1. No data transmission delays and errors were used in these simullations. Other parameters were the same during all simulations (5.7 tab.).

5.7 table

Parameters

ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
1	0	0	0	0	0	1	0.8	0.1	0.1	50	6	0.1	70	90	150

5.8 table

Output data (simulation with ANN)

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
6	1	23	11	86	624	421	1135	61	0
7	1	20	11	59	615	422	1601	47	0
8	1	24	11	54	681	435	1257	53	0
9	1	22	11	80	623	426	1133	60	0
10	1	22	11	52	657	426	1529	48	0
AVG		**22**	**11**	**66**	**640**	**426**	**1331**	**54**	**0**

5.9 table

Output data (simulation with INN)

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
11	1	17	11	39	632	423	1214	93	0
12	1	19	11	52	653	430	1367	78	0
13	1	19	11	55	666	421	1520	103	0
14	1	19	11	40	642	420	1268	96	0
15	1	19	11	65	651	431	1615	88	0
AVG		**19**	**11**	**50**	**649**	**425**	**1397**	**92**	**0**

Results of simulations:

- Output data show that number of collisions was reduced to zero during simulations with ANN, but the number of completed trips was much smaller than during simulation without any motion control. Futhermore, no lags or data transmission delays were used during this simulation (5.8 tab.).
- The INN based model was used for further simulations, on purpose to increase the number of trips. During simulations with the proposed INN, the same parameters were used. In the result no collisions were detected and the number of trips was increased till 92 trips per 10 minutes long simulation (5.9 tab.).

2.2. Data transmission delays were used in the next simulations. Other parameters were the same during all simulations (5.10 tab.).

5.10 table

Parameters

ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
2	0	0	0	2000	500	1	0.8	0.1	0.1	50	6	0.1	70	90	150

5.11 table

Output data (simulation with ANN)

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
16	2	21	11	78	654	437	1579	12	11
17	2	13	11	38	609	440	988	13	10
18	2	14	11	43	622	441	1012	11	8
19	2	22	11	78	656	437	1358	12	9
20	2	21	11	68	598	429	1082	12	11
AVG		**18**	**11**	**61**	**628**	**437**	**1204**	**12**	**10**

5.12 table

Output data (simulation with INN)

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
21	2	66	13	367	817	453	2017	46	16

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
22	2	37	12	76	770	473	1554	5	5
23	2	23	11	38	700	438	988	18	2
24	2	60	13	151	815	444	1704	30	8
25	2	24	12	39	705	443	988	16	11
AVG		**42**	**12**	**134**	**761**	**450**	**1450**	**23**	**8**

Results of simulations:

- Data transmission delay 2000 ms and own positioning data refresh delay 500 ms was used during these simulations.
- 12 trips were done on average during simulation with a traditional ANN (5.11 tab.), that is twice less than during simulations with the proposed INN (5.12 tab.). But collisions were detected in all simulations. Requirements were changed and the values of other parameters must be changed as well to provide safety drive.

2.3. The same data transmission delay was used in combination with other parameters (5.13 tab.).

5.13 table

Parameters

ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
3	0	0	0	2000	500	1	0.8	0.1	0.1	50	6	0.1	70	90	500

5.14 table

Output data (simulation with ANN)

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
26	3	25	11	40	679	436	1006	15	0
27	3	27	13	42	772	469	1453	5	0
28	3	29	11	73	761	439	1699	2	0
29	3	43	15	98	902	502	1435	9	0
30	3	26	11	41	749	430	962	5	0

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
AVG		**30**	**12**	**59**	**773**	**455**	**1311**	**7**	**0**

5.15 table

Output data (simulation with INN)

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
31	3	29	11	60	692	439	1234	15	0
32	3	35	11	146	771	435	1875	13	0
33	3	39	11	150	855	436	1934	13	0
34	3	28	11	75	645	439	1235	20	0
35	3	31	11	112	666	448	1568	8	0
AVG		**32**	**11**	**109**	**726**	**439**	**1569**	**14**	**0**

Results of simulations:

- An increased parameter "maximal distance till other UEV to start crash prevention and to calculate collision probability" was used during these simulations.
- Otput data show that number of trips was reduced, but they became safer and without collisions.
- The number of trips was twice bigger during simulations with the proposed INN (5.14 tab., 5.15 tab.).

2.4. Data transmission delays and errors were used in next simulations. Other parameters were the same during all simulations (5.16 tab.).

5.16 table

Parameters

ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
4	10	1	1	100	100	1	0.8	0.1	0.1	100	6	0.5	70	90	500

5.17 table

Output data (simulation with ANN)

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
36	4	92	15	370	1725	752	7852	19	0
37	4	62	11	526	1998	932	6620	18	0
38	4	85	14	305	2360	931	4494	14	0
39	4	88	14	401	2654	902	4065	15	0
40	4	92	16	386	1855	825	5016	19	0
AVG		**84**	**14**	**398**	**2118**	**868**	**5609**	**17**	**0**

5.18 table

Output data (simulation with INN)

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
41	4	55	11	311	1250	434	5569	32	0
42	4	49	11	422	1108	430	6058	27	0
43	4	55	11	325	1250	435	5569	34	0
44	4	54	11	302	1250	434	7112	40	0
45	4	48	11	593	1073	428	9236	22	0
AVG		**52**	**11**	**391**	**1186**	**432**	**6709**	**31**	**0**

Results of simulations:

- Data delays and errors were used, to reproduce the conditions of the real time experiment during these simulations.
- No collisions were detected in all simulations.
- The number of trips was almost twice bigger during simulations with the proposed INN (5.17 tab., 5.18 tab.).

5.6.3. Experiment Nr.3

Results of computer simulations described in previous chapter, prove that the proposed novel INN is good in collision probability minimization, but also helps to minimize necessary time for calculation and to provide the bigger number of trips.

The purpose of the next simulations is to compare simulation results with different set of parameters and to understand their influence on the output data.

Parameters of computer simulations are mentioned in 5.4 table, but output data is mentioned in 5.5 table of Experiment Nr.2.

1. Fixed size of vehicles

   1.1. Without data transmission delay and errors

   3 simulations were made for each set of parameters to make sure the chosen parameters are good for the assigned task.

      1.1.1. Number of iterations

      Number of iterations TrLim is the first parameter to be changed (5.19 tab.).

5.19 table

Parameters

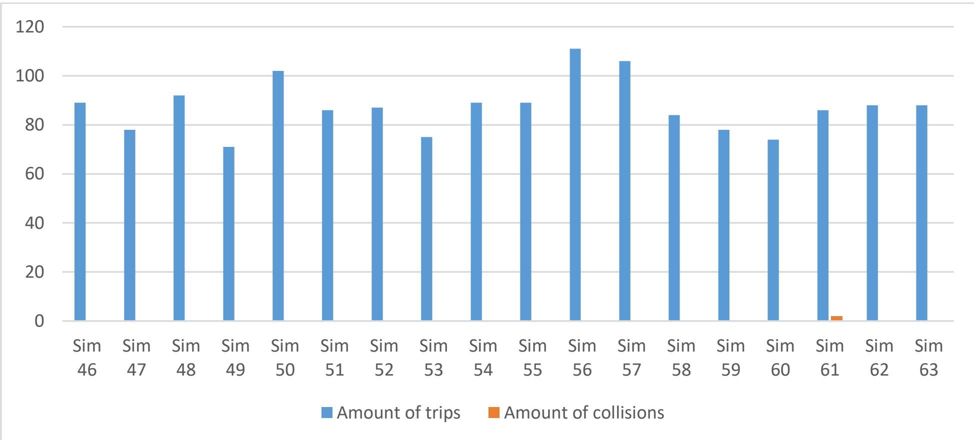
ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
4	0	0	0	0	0	1	0.8	0.1	0.1	30	6	0.1	70	90	150
5	0	0	0	0	0	1	0.8	0.1	0.1	50	6	0.1	70	90	150
6	0	0	0	0	0	1	0.8	0.1	0.1	80	6	0.1	70	90	150
7	0	0	0	0	0	1	0.8	0.1	0.1	100	6	0.1	70	90	150
8	0	0	0	0	0	1	0.8	0.1	0.1	150	6	0.1	70	90	150
9	0	0	0	0	0	1	0.8	0.1	0.1	200	6	0.1	70	90	150

5.20 table

Output data

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
46	4	19	11	55	658	425	1791	89	0
47	4	19	11	52	677	430	1620	78	0
48	4	18	11	39	632	423	1214	92	0
49	5	19	11	51	650	432	1368	71	0
50	5	19	11	55	666	421	1520	102	0
51	5	18	11	39	640	422	1266	86	0
52	6	19	11	65	651	431	1616	87	0
53	6	20	11	63	697	426	1921	75	0

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
54	6	19	11	53	654	421	1483	89	0
55	**7**	**20**	**11**	**54**	**673**	**433**	**1500**	**89**	**0**
56	**7**	**19**	**11**	**37**	**653**	**421**	**1238**	**111**	**0**
57	**7**	**20**	**11**	**72**	**692**	**433**	**1556**	**106**	**0**
58	8	20	11	50	688	421	1362	84	0
59	8	21	11	63	700	432	1470	78	0
60	8	18	11	46	631	433	1270	74	0
61	9	20	11	57	681	429	1536	86	2
62	9	17	11	41	616	422	1254	88	0
63	9	18	11	40	649	434	1109	88	0



5.23 fig. Comparison of the results of simulations.

Results of simulations (5.20 tab.):

- Two collisions were detected during the simulation with number of iterations = 200. The proposed system is a real time system that needs to make decisions fast, that is why a big number iterations is not the best solution for the proposed system.
- The biggest average amount of trips was stated during the simulation with number of iterations = 100 iterations. This parameter was used for further simulations.

1.1.2. Probability sensitivity

Probability sensitivity is the next parameter to be changed (5.21 tab.).

5.21 table

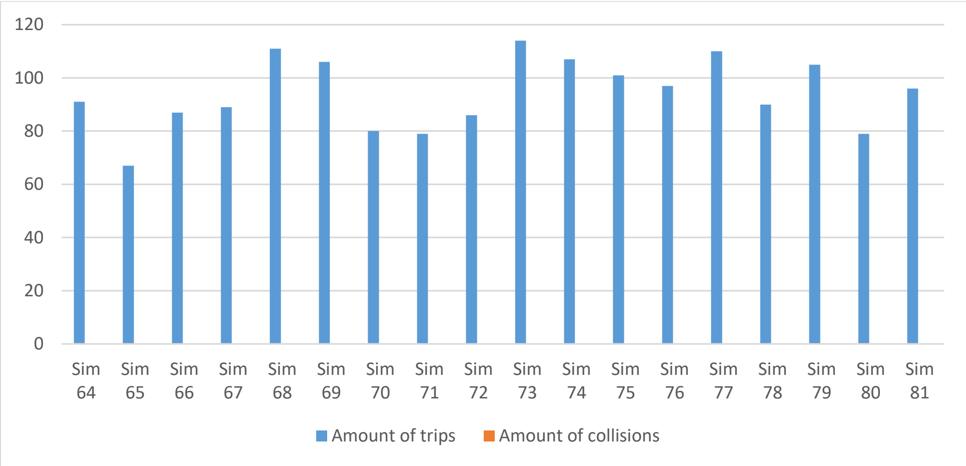
Parameters

ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
10	0	0	0	0	0	1	0.8	0.1	0.1	100	6	0.05	70	90	150
11	0	0	0	0	0	1	0.8	0.1	0.1	100	6	0.1	70	90	150
12	0	0	0	0	0	1	0.8	0.1	0.1	100	6	0.2	70	90	150
13	0	0	0	0	0	1	0.8	0.1	0.1	100	6	0.3	70	90	150
14	0	0	0	0	0	1	0.8	0.1	0.1	100	6	0.4	70	90	150
15	0	0	0	0	0	1	0.8	0.1	0.1	100	6	0.5	70	90	150

5.22 table

Output data

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
64	10	19	11	51	655	432	1402	91	0
65	10	20	11	42	659	423	1093	67	0
66	10	18	11	48	641	422	1567	87	0
67	11	20	11	54	673	433	1500	89	0
68	11	19	11	37	653	421	1238	111	0
69	11	20	11	72	692	433	1556	106	0
70	12	18	11	45	624	433	1348	80	0
71	12	19	11	45	675	424	1348	79	0
72	12	18	11	52	631	423	1229	86	0
73	**13**	**18**	**11**	**63**	**641**	**422**	**1339**	**114**	**0**
74	**13**	**19**	**11**	**47**	**684**	**424**	**1603**	**107**	**0**
75	**13**	**18**	**11**	**46**	**638**	**421**	**1602**	**101**	**0**
76	14	19	11	63	658	423	1272	97	0
77	14	18	11	43	645	421	1286	110	0
78	14	18	11	51	644	420	1631	90	0
79	15	18	11	42	640	420	1246	105	0
80	15	21	11	74	689	433	1430	79	0
81	15	19	11	43	654	427	1130	96	0



5.24 fig. Comparison of the results of simulations.

Results of simulations (5.22 tab.):

- Probability sensitivity does not influence the number of trips or increase the number of collisions.
- Perhaps a bigger number of simulations will provide more information and better material for analyses. This will be one of the purposes for the future research.
- Probability sensitivity 0.3 is used for further simulations.

1.1.3. Trajectory (angle)

Trajectory or angle change AngLim was the next parameter to be changed (5.23 tab.).

5.23 table

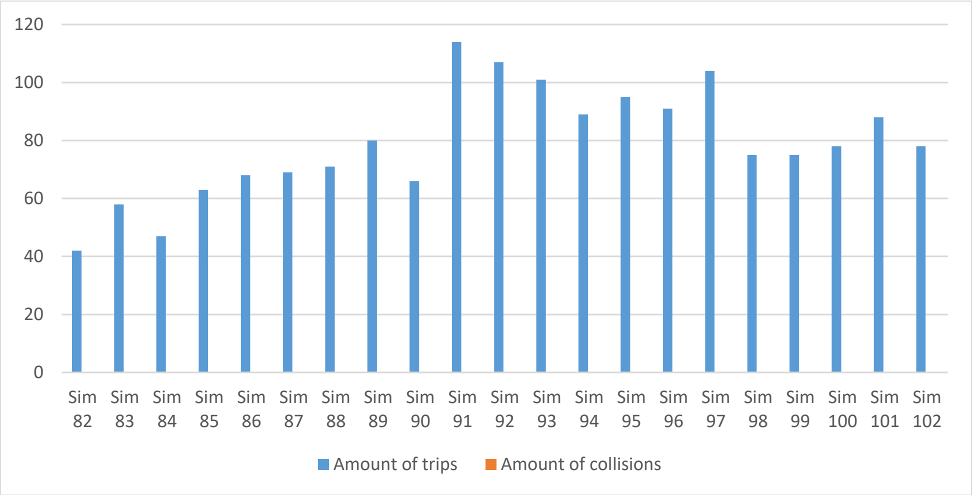
Parameters

ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
16	0	0	0	0	0	1	0.8	0.1	0.1	100	6	0.3	70	15	150
17	0	0	0	0	0	1	0.8	0.1	0.1	100	6	0.3	70	45	150
18	0	0	0	0	0	1	0.8	0.1	0.1	100	6	0.3	70	60	150
19	0	0	0	0	0	1	0.8	0.1	0.1	100	6	0.3	70	90	150
20	0	0	0	0	0	1	0.8	0.1	0.1	100	6	0.3	70	120	150
21	0	0	0	0	0	1	0.8	0.1	0.1	100	6	0.3	70	180	150
22	0	0	0	0	0	1	0.8	0.1	0.1	100	6	0.3	70	360	150

5.24 table

Output data

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
82	16	24	11	48	615	433	978	42	0
83	16	24	11	80	620	433	1222	58	0
84	16	20	11	75	564	432	946	47	0
85	17	19	10	51	616	422	1226	63	0
86	17	19	11	52	620	421	1447	68	0
87	17	19	11	48	610	432	1174	69	0
88	18	19	11	66	619	432	1398	71	0
89	18	18	11	38	600	422	996	80	0
90	18	19	11	42	639	433	1188	66	0
91	**19**	**18**	**11**	**63**	**641**	**422**	**1339**	**114**	**0**
92	**19**	**19**	**11**	**47**	**684**	**424**	**1603**	**107**	**0**
93	**19**	**18**	**11**	**46**	**638**	**421**	**1602**	**101**	**0**
94	20	20	11	56	700	422	1800	89	0
95	20	19	11	45	679	425	1413	95	0
96	20	19	11	47	683	422	1474	91	0
97	21	21	11	53	758	434	1679	104	0
98	21	20	11	58	700	423	1616	75	0
99	21	20	11	78	706	433	2321	75	0
100	22	21	11	48	740	422	1661	78	0
101	22	19	11	46	692	421	1493	88	0
102	22	21	11	68	763	433	2081	78	0



5.25 fig. Comparison of the results of simulations.

Results of simulations (5.24 tab.):

- The possibility to change the trajectory plays an important role. On the one hand, possibility to change the angle only slightly can increase the number of trips, because vehicles will deviate from the route slightly, but on the other hand it can decrease the number of trips, because the decision of slowing down can be made more often. That is why it is important to find the optimal angle change.
- No collisions were detected during the simulations. The biggest number of trips was made during simulations with 90˚ angle change.

1.1.4. Maximal distance till crossing point

Maximal distance Safedist till other UEV to start crash prevention and to calculate collision probability is the next parameter to compare (5.25 tab.).

5.25 table

Parameters

ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
23	0	0	0	0	0	1	0.8	0.1	0.1	100	6	0.3	70	90	50
24	0	0	0	0	0	1	0.8	0.1	0.1	100	6	0.3	70	90	100
25	0	0	0	0	0	1	0.8	0.1	0.1	100	6	0.3	70	90	150
26	0	0	0	0	0	1	0.8	0.1	0.1	100	6	0.3	70	90	200
27	0	0	0	0	0	1	0.8	0.1	0.1	100	6	0.3	70	90	250

ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
28	0	0	0	0	0	1	0.8	0.1	0.1	100	6	0.3	70	90	300

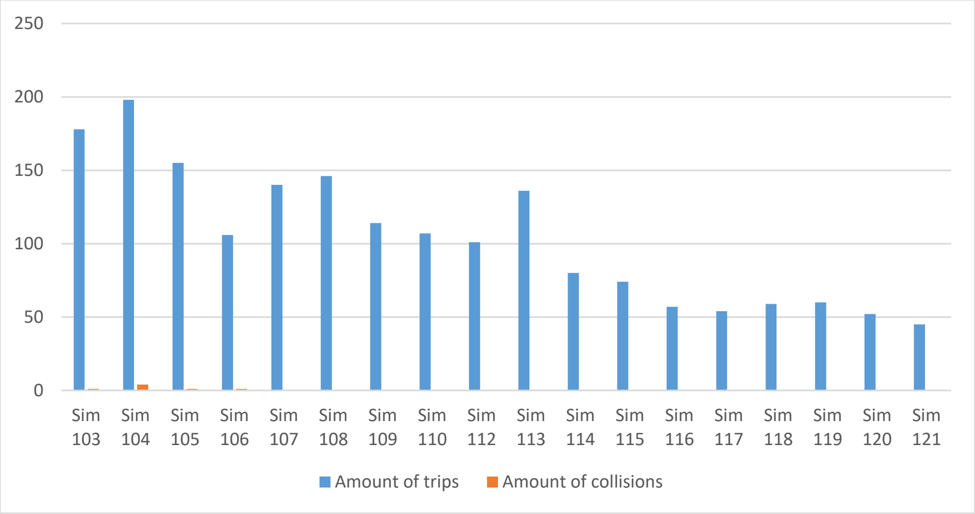
5.26 table

Output data

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
103	23	16	11	31	600	421	1047	178	1
104	23	15	11	25	585	421	936	198	4
105	23	16	11	30	596	421	1001	155	1
106	24	17	11	38	611	423	1035	106	1
107	24	17	11	35	619	421	1232	140	0
108	24	17	11	34	618	421	1249	146	0
109	**25**	**18**	**11**	**63**	**641**	**422**	**1339**	**114**	**0**
110	**25**	**19**	**11**	**47**	**684**	**424**	**1603**	**107**	**0**
112	**25**	**18**	**11**	**46**	**638**	**421**	**1602**	**101**	**0**
113	26	16	11	49	609	421	1249	136	0
114	26	20	11	60	683	433	1544	80	0
115	26	21	11	53	700	421	1646	74	0
116	27	24	11	68	774	433	1901	57	0
117	27	22	11	68	721	433	2196	54	0
118	27	23	11	56	748	433	1587	59	0
119	28	21	11	70	697	433	2336	60	0
120	28	27	11	80	835	434	2528	52	0
121	28	23	11	93	743	435	2558	45	0

Results of simulations (5.26 tab.):

- Results of simulation prove the hypothesis – the smaller is maximal distance till crossing point, the more trips are done, but the number of detected collisions increases.
- Distance = 150 was used for further simulations.



5.26 fig. Comparison of the results of simulations.

1.1.5. Collision probability weight

Collision probability weight Ap was the next parameter to be changed (5.27 tab.).

5.27 table

Parameters

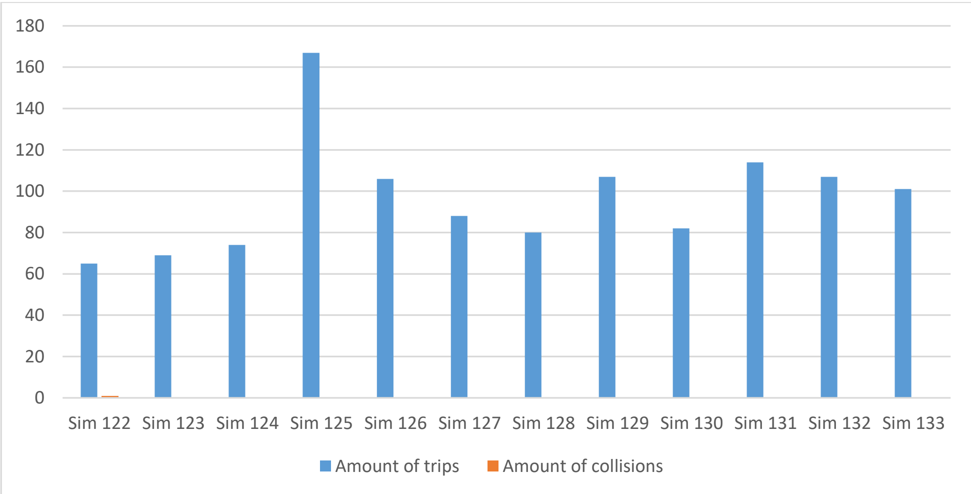
ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
29	0	0	0	0	0	1	0.2	0.4	0.4	100	6	0.3	70	90	150
30	0	0	0	0	0	1	0.4	0.3	0.3	100	6	0.3	70	90	150
31	0	0	0	0	0	1	0.6	0.2	0.2	100	6	0.3	70	90	150
32	0	0	0	0	0	1	0.8	0.1	0.1	100	6	0.3	70	90	150

5.28 table

Output data

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
122	29	20	11	41	696	424	1304	65	1
123	29	19	11	41	662	433	1239	69	0
124	29	18	11	38	656	433	1335	74	0

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
125	30	19	11	47	667	430	1550	167	0
126	30	18	11	48	641	423	1305	106	0
127	30	19	11	36	662	431	1099	88	0
128	31	18	11	64	652	425	1915	80	0
129	31	18	11	49	627	422	1457	107	0
130	31	18	11	52	647	430	1470	82	0
131	**32**	**18**	**11**	**63**	**641**	**422**	**1339**	**114**	**0**
132	**32**	**19**	**11**	**47**	**684**	**424**	**1603**	**107**	**0**
133	**32**	**18**	**11**	**46**	**638**	**421**	**1602**	**101**	**0**



5.27 fig. Comparison of the results of simulations.

Results of simulations (5.28 tab.):

- The smaller is weight of collision probability the bigger is collision probability, that is why despite of a big number of trips during the simulations 125-127, simulations 131-133, with a weight of collision probability = 0.8, was chosen as the best.

1.2. With data transmission delays

3 simulations were made for each set of parameters to make sure the chosen parameters are good for the assigned task. Data transmission delays were used in these simulations.

1.2.1. Safety zone area rate proportional to the size

Safety zone area rate AreaRate proportional to the size is the next parameter to be changed (5.29 tab.).

5.29 table

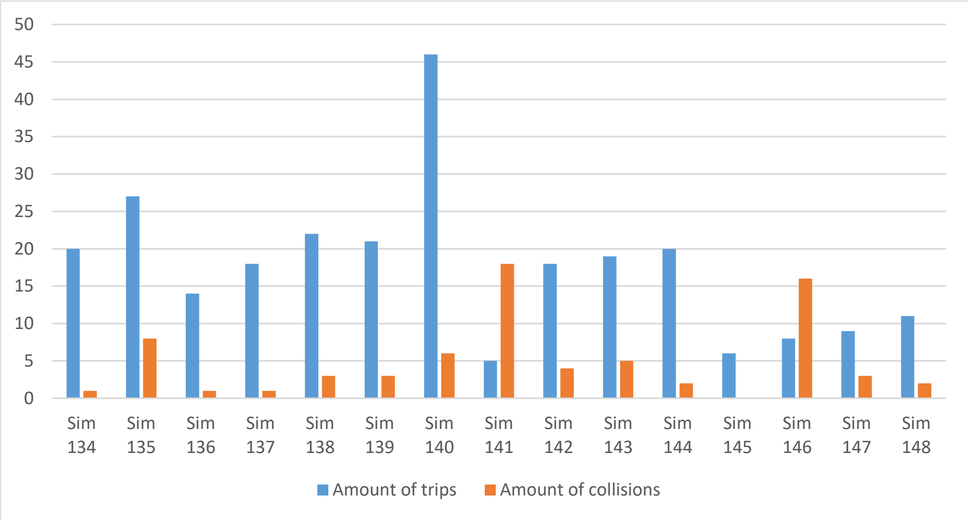
Parameters

ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
33	0	0	0	2000	500	1	0.8	0.1	0.1	100	6	0.3	70	90	150
34	0	0	0	2000	500	2	0.8	0.1	0.1	100	6	0.3	70	90	150
35	0	0	0	2000	500	3	0.8	0.1	0.1	100	6	0.3	70	90	150
36	0	0	0	2000	500	4	0.8	0.1	0.1	100	6	0.3	70	90	150
37	0	0	0	2000	500	5	0.8	0.1	0.1	100	6	0.3	70	90	150

5.30 table

Output data

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
134	33	20	11	41	626	422	980	20	1
135	33	26	11	57	734	435	1556	27	8
136	33	19	11	37	606	439	918	14	1
137	34	28	11	98	753	438	1859	18	1
138	34	33	11	115	808	439	1792	22	3
139	34	19	11	49	633	435	1271	21	3
140	35	23	11	106	699	434	1723	46	6
141	35	19	13	25	614	447	885	5	18
142	35	21	11	78	654	437	1579	18	4
143	36	16	11	28	549	440	1089	19	5
144	36	24	11	54	727	423	1286	20	2
145	36	24	11	70	739	437	1748	6	0
146	37	66	13	367	817	453	2017	8	16
147	37	37	12	76	770	473	1554	9	3
148	37	23	11	38	700	438	988	11	2



5.28 fig. Comparison of the results of simulations.

Results of simulations (5.30 tab.):

- Collisions were detected during almost every simulation. In simulations 134-136 the same parameters were used as in simulations 131-133, where no collisions were detected and the averaged number of trips was 107 trips during one simulation. The only difference is that in simulations 134-136 data transmission delay was used. That proves that data transmission delay is an important parameter, that needs to be taken in account during system developer.
- Next simulations were made with an aim to minimize the collision probability.

1.2.2. Maximal distance till crossing point

Maximal distance till other UEV to start crash prevention and to calculate collision probability is a next parameter to compare (5.31 tab.).

5.31 table

Parameters

ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
38	0	0	0	2000	500	1	0.9	0.05	0.05	100	6	0.3	70	90	150
39	0	0	0	2000	500	1	0.9	0.05	0.05	100	6	0.3	70	90	200
40	0	0	0	2000	500	1	0.9	0.05	0.05	100	6	0.3	70	90	250
41	0	0	0	2000	500	1	0.9	0.05	0.05	100	6	0.3	70	90	300
42	0	0	0	2000	500	1	0.9	0.05	0.05	100	6	0.3	70	90	350

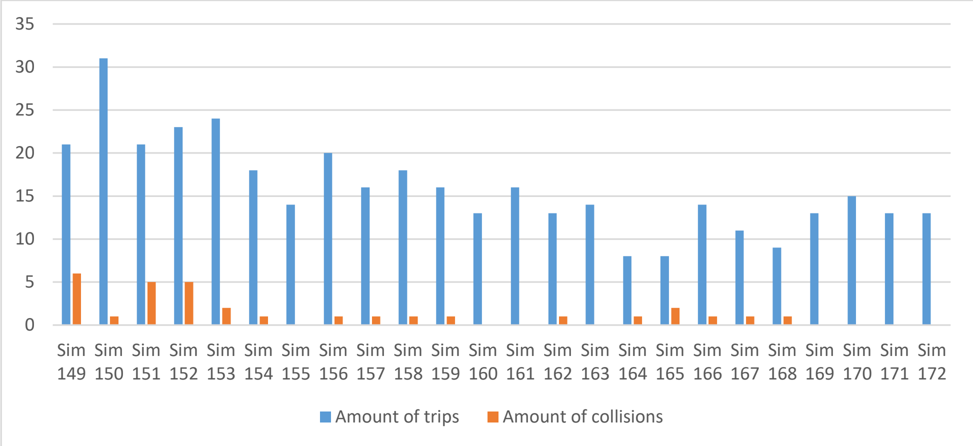
ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
43	0	0	0	2000	500	1	0.9	0.05	0.05	100	6	0.3	70	90	400
44	0	0	0	2000	500	1	0.9	0.05	0.05	100	6	0.3	70	90	450
45	0	0	0	2000	500	1	0.9	0.05	0.05	100	6	0.3	70	90	500

5.32 table

Output data

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
149	38	21	11	53	651	434	1275	21	6
150	38	22	11	60	656	430	1277	31	1
151	38	20	11	36	670	434	1216	21	5
152	39	20	11	36	668	434	1216	23	5
153	39	26	11	80	679	436	1197	24	2
154	39	25	11	79	735	438	1282	18	1
155	40	30	11	86	778	445	1272	14	0
156	40	29	11	90	732	435	1659	20	1
157	40	32	11	97	790	446	1880	16	1
158	41	33	11	70	764	441	1535	18	1
159	41	34	11	169	877	439	3445	16	1
160	41	23	11	43	679	436	1147	13	0
161	42	27	14	44	768	468	1451	16	0
162	42	28	11	70	761	437	1698	13	1
163	42	44	18	95	912	502	1420	14	0
164	43	28	11	44	753	434	978	8	1
165	43	38	16	129	799	466	1913	8	2
166	43	24	11	47	664	438	965	14	1
167	44	24	11	88	663	439	1924	11	1
168	44	34	11	76	882	436	1611	9	1
169	44	59	19	190	1133	454	2835	13	0
170	**45**	**29**	**11**	**60**	**692**	**439**	**1234**	**15**	**0**
171	**45**	**35**	**11**	**146**	**771**	**435**	**1875**	**13**	**0**

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
172	**45**	**39**	**11**	**150**	**855**	**436**	**1934**	**13**	**0**



5.29 fig. Comparison of the results of simulations.

Results of simulations (5.32 tab.):

- The smaller the safety distance is, the more collisions were detected.
- No collisions were detected only during the safety distance = 500, that is why this value was used for further simulations.

1.3. With data transmission delays and data errors

3 simulations were made for each set of parameters to make sure the chosen parameters are good for the assigned task. Data transmission delays and data errors were used in these simulations.

1.3.1. Probability sensitivity

Probability sensitivity is the next parameter to be changed (5.33 tab.).

5.33 table

Parameters

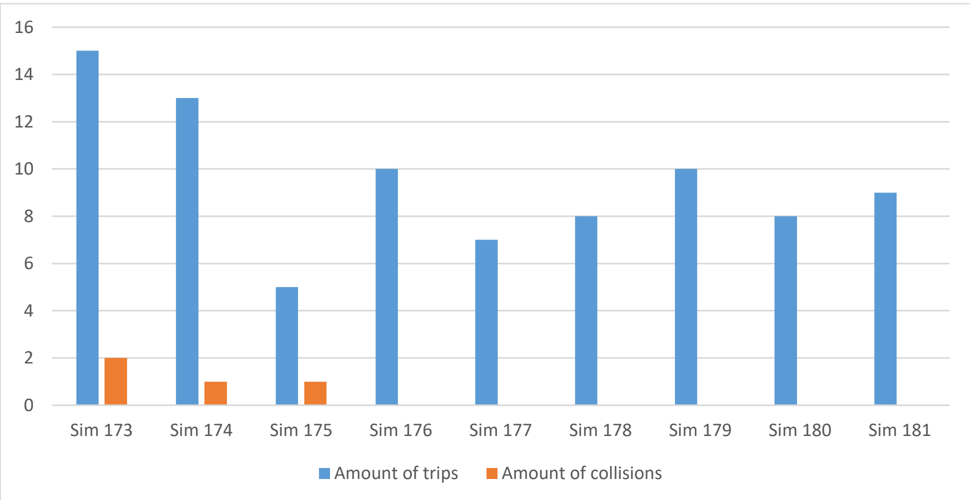
ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
46	50	5	3	2000	500	1	0.9	0.05	0.05	100	6	0.3	70	90	500

ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
47	50	5	3	2000	500	1	0.9	0.05	0.05	100	6	0.4	70	90	500
48	50	5	3	2000	500	1	0.9	0.05	0.05	100	6	0.5	70	90	500

5.34 table

Output data

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
173	46	34	11	131	807	436	2051	15	2
174	46	50	11	242	1016	440	3690	13	1
175	46	44	21	87	1104	581	1684	5	1
176	47	34	11	192	807	438	3393	10	0
177	47	70	11	332	1233	443	4081	7	0
178	47	64	11	332	1136	443	4081	8	0
179	**48**	**34**	**16**	**68**	**805**	**447**	**1276**	**10**	**0**
180	**48**	**33**	**15**	**61**	**760**	**455**	**969**	**8**	**0**
181	**48**	**34**	**17**	**98**	**790**	**463**	**1574**	**9**	**0**



5.30 fig. Comparison of the results of simulations.

Results of simulations (5.34 tab.):

- Collisions were detected in simulations 173-175, where probability sensitivity = 0.3.
- The biggest number of safety trips was done in simulations 179-181, where probability sensitivity = 0.5. This value was used for further simulations.

1.3.2. Errors and delays

The values of errors and delays were changed in next simulations, in purpose to check the behavior of the proposed INN system (5.35 tab.).

5.35 table

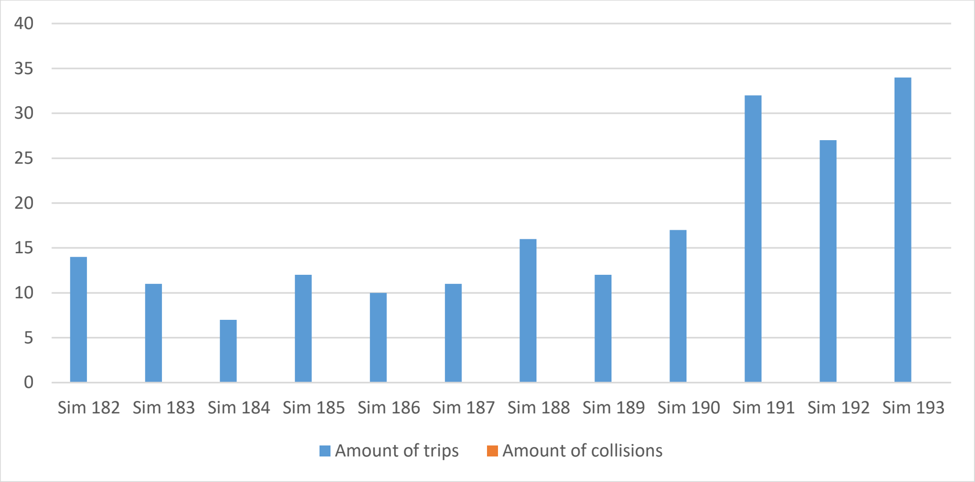
Parameters

ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
49	40	4	2.5	1500	400	1	0.9	0.05	0.05	100	6	0.5	70	90	500
50	30	3	2	1000	300	1	0.9	0.05	0.05	100	6	0.5	70	90	500
51	20	2	1.5	500	200	1	0.9	0.05	0.05	100	6	0.5	70	90	500
52	10	1	1	100	100	1	0.9	0.05	0.05	100	6	0.5	70	90	500

5.36 table

Output data

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
182	49	30	12	98	731	441	1574	14	0
183	49	29	12	89	730	437	1920	11	0
184	49	27	12	58	684	437	1049	7	0
185	50	20	11	47	609	438	1252	12	0
186	50	42	12	110	977	439	1981	10	0
187	50	34	11	119	812	436	2036	11	0
188	51	25	11	63	706	436	1342	16	0
189	51	31	11	123	726	445	1539	12	0
190	51	25	11	92	680	437	2079	17	0
191	52	28	11	122	795	435	3022	32	0
192	52	25	11	86	750	434	1987	27	0
193	52	37	11	196	1059	435	4807	34	0



5.31 fig. Comparison of the results of simulations.

Results of simulations (5.36 tab.):

- Errors and delays were reduced in simulations 182-193. No collisions were detected. The number of trips was biggest in the last simulations, were errors and delays were small enough.

2. Random size of vehicles

Random size of vehicles was used for further simulations. Only one simulation with each parameter value set was done.

2.1. Without data transmission delay and errors

No data transmission delays or errors were used in these simulations.

2.1.1. Number of iterations

Number of iterations is the first parameter to be changed (5.37 tab.).

5.37 table

Parameters

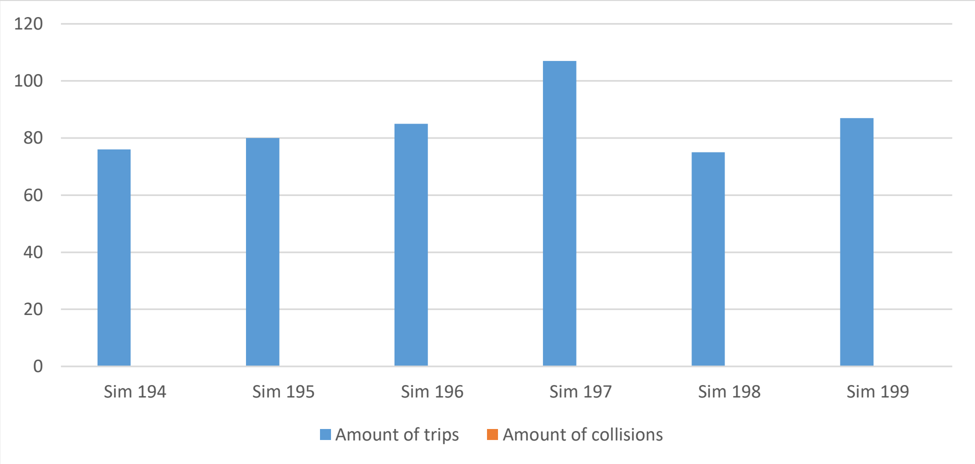
ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
53	0	0	0	0	0	1	0.8	0.1	0.1	30	6	0.1	70	90	150
54	0	0	0	0	0	1	0.8	0.1	0.1	50	6	0.1	70	90	150
55	0	0	0	0	0	1	0.8	0.1	0.1	80	6	0.1	70	90	150
56	0	0	0	0	0	1	0.8	0.1	0.1	100	6	0.1	70	90	150

ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
57	0	0	0	0	0	1	0.8	0.1	0.1	150	6	0.1	70	90	150
58	0	0	0	0	0	1	0.8	0.1	0.1	200	6	0.1	70	90	150

5.38 table

Output data

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
194	53	20	9	90	691	423	1843	76	0
195	54	21	8	59	678	424	1318	80	0
196	55	19	8	47	726	423	1211	85	0
197	**56**	**18**	**8**	**51**	**630**	**422**	**1170**	**107**	**0**
198	57	21	9	81	761	425	1419	75	0
199	58	21	9	81	693	437	1533	87	0



5.32 fig. Comparison of the results of simulations.

Results of simulations (5.38 tab.):

- The biggest number of trips was during the simulation with number of iterations = 100. This number of iterations will be used for further simulations.

2.1.2. Probability sensitivity

Probability sensitivity is the next parameter to be changed (5.39 tab.).

5.39 table

Parameters

ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
59	0	0	0	0	0	0	0.8	0.1	0.1	100	6	0.05	70	90	150
60	0	0	0	0	0	0	0.8	0.1	0.1	100	6	0.1	70	90	150
61	0	0	0	0	0	0	0.8	0.1	0.1	100	6	0.2	70	90	150
62	0	0	0	0	0	0	0.8	0.1	0.1	100	6	0.3	70	90	150
63	0	0	0	0	0	0	0.8	0.1	0.1	100	6	0.4	70	90	150
64	0	0	0	0	0	0	0.8	0.1	0.1	100	6	0.5	70	90	150

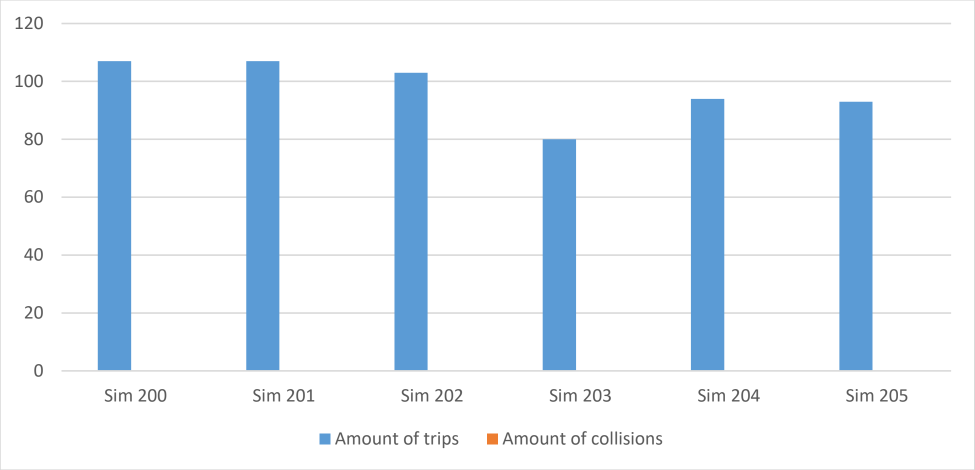
5.40 table

Output data

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
200	59	19	9	59	693	428	1308	107	0
201	**60**	**18**	**8**	**51**	**630**	**422**	**1170**	**107**	**0**
202	61	18	8	59	667	423	1399	103	0
203	62	20	9	45	711	424	1130	80	0
204	63	20	8	93	754	421	1624	94	0
205	64	18	8	70	666	421	1668	93	0

Results of simulations (5.40 tab.):

- During simulations no collisions were detected. The best result was received while using the minimal probability sensitivity 0.05-0.1.
- For further simulations probability sensitivity = 0.1 was used.



5.33 fig. Comparison of the results of simulations.

2.1.3. Trajectory (angle)

Trajectory or angle change was the next parameter to be changed (5.41 tab.).

5.41 table

Parameters

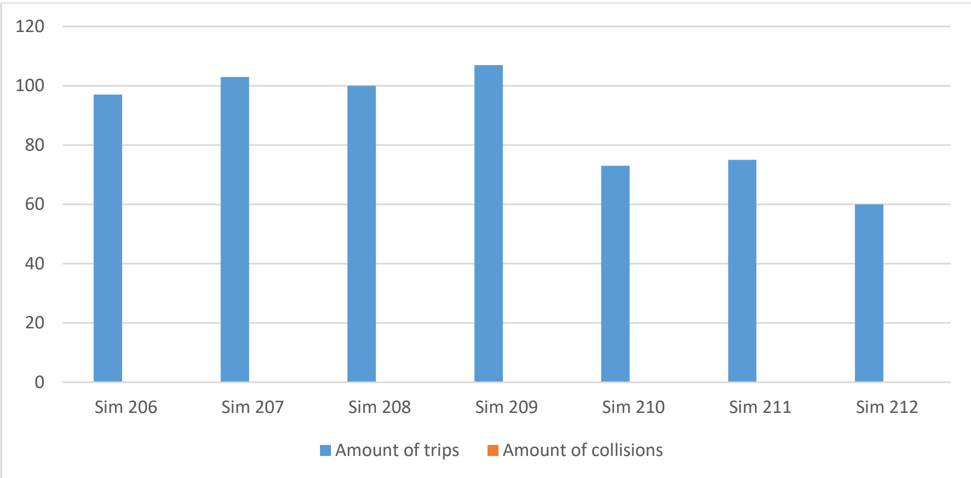
ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
65	0	0	0	0	0	0	0.8	0.1	0.1	100	6	0.1	70	15	150
66	0	0	0	0	0	0	0.8	0.1	0.1	100	6	0.1	70	45	150
67	0	0	0	0	0	0	0.8	0.1	0.1	100	6	0.1	70	60	150
68	0	0	0	0	0	0	0.8	0.1	0.1	100	6	0.1	70	90	150
69	0	0	0	0	0	0	0.8	0.1	0.1	100	6	0.1	70	120	150
70	0	0	0	0	0	0	0.8	0.1	0.1	100	6	0.1	70	180	150
71	0	0	0	0	0	0	0.8	0.1	0.1	100	6	0.1	70	360	150

5.42 table

Output data

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
206	65	18	8	70	668	421	1668	97	0

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
207	66	18	8	59	668	423	1390	103	0
208	67	17	9	54	699	424	1311	100	0
209	**68**	**18**	**8**	**51**	**630**	**422**	**1170**	**107**	**0**
210	69	20	9	60	751	425	1856	73	0
211	70	19	8	84	768	421	2395	75	0
212	71	25	8	61	1001	422	2445	60	0



5.34 fig. Comparison of the results of simulations.

Results of simulations (5.42 tab.):

- Results of simulations are shown in figure 5.34. No collisions were detected during simulations. The best result was still received during the simulation with trajectory angle change = +-90˚. This value was used for further simulations.

2.1.4. Maximal distance till crossing point

Maximal distance till other UEV to start crash prevention and to calculate collision probability is a next parameter to compare (5.43 tab.).

5.43 table

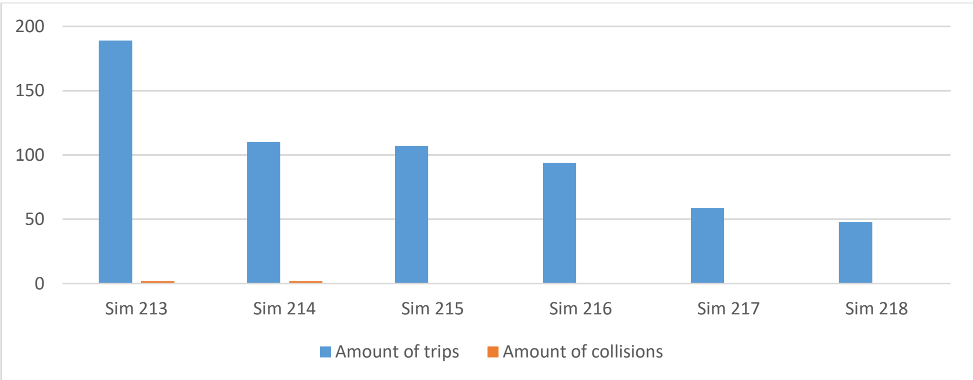
Parameters

ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
72	0	0	0	0	0	0	0.8	0.1	0.1	100	6	0.1	70	90	50
73	0	0	0	0	0	0	0.8	0.1	0.1	100	6	0.1	70	90	100
74	0	0	0	0	0	0	0.8	0.1	0.1	100	6	0.1	70	90	150
75	0	0	0	0	0	0	0.8	0.1	0.1	100	6	0.1	70	90	200
76	0	0	0	0	0	0	0.8	0.1	0.1	100	6	0.1	70	90	250
77	0	0	0	0	0	0	0.8	0.1	0.1	100	6	0.1	70	90	300

5.44 table

Output data

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
213	72	13	8	34	613	421	1383	189	2
214	73	17	8	43	666	422	1352	110	2
215	**74**	**18**	**8**	**51**	**630**	**422**	**1170**	**107**	**0**
216	75	20	8	70	780	443	1971	94	0
217	76	24	8	73	900	436	2003	59	0
218	77	35	9	109	1082	436	2629	48	0



5.35 fig. Comparison of the results of simulations.

Results of simulations (5.44 tab.):

- Results of simulations show that the smaller safety distance is, the biggest number of trips UEVs make. But this leads to possible collisions.
- 189 trips were made, and 2 collisions were detected during simulation 213. 110 trips were made and also 2 collisions were detected during simulation 214. No collisions were detected during all the other simulations.
- The results of simulation 215 are the best in this comparison and safety distance = 150 was used for further simulations.

2.1.5. Collision probability weight

Collision probability weight was the next parameter to be changed (5.45 tab.).

5.45 table

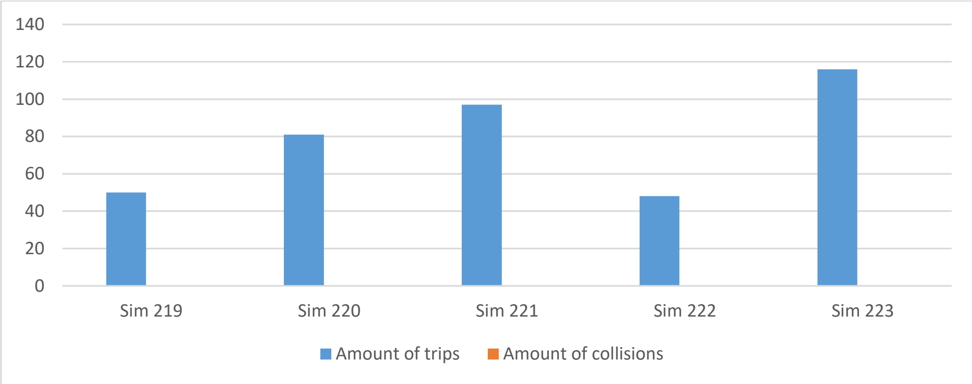
Parameters

ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
78	0	0	0	0	0	0	0.5	0.25	0.25	100	6	0.1	70	90	150
79	0	0	0	0	0	0	0.6	0.2	0.2	100	6	0.1	70	90	150
80	0	0	0	0	0	0	0.7	0.15	0.15	100	6	0.1	70	90	150
81	0	0	0	0	0	0	0.8	0.1	0.1	100	6	0.1	70	90	300
82	0	0	0	0	0	0	0.9	0.05	0.05	100	6	0.1	70	90	150

5.46 table

Output data

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
219	78	35	9	109	1072	436	2629	50	0
220	79	22	9	74	830	428	2205	81	0
221	80	18	8	70	670	421	1666	97	0
222	81	35	9	109	1082	436	2629	48	0
223	**82**	**16**	**8**	**46**	**636**	**423**	**1323**	**116**	**0**



5.36 fig. Comparison of the results of simulations.

Results of simulations (5.46 tab.):

- No collisions were detected during these simulations.
- The biggest number of trips = 116 was detected during simulation 223, where collision probability weight = 0.9.

2.2. With data transmission delays

Data transmission delays were used in the next simulations.

2.2.1. Safety zone area rate proportional to the size

Safety zone area rate proportional to the size is the next parameter to be changed (5.47 tab.).

5.47 table

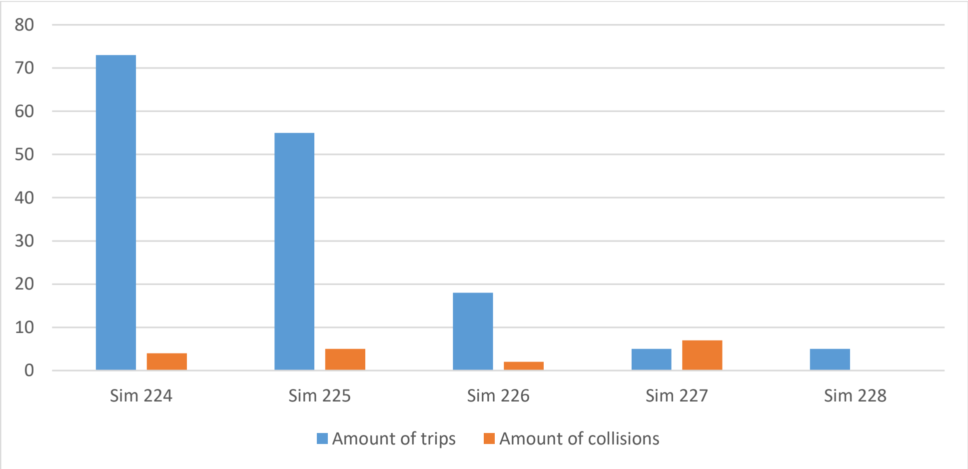
Parameters

ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
83	0	0	0	2000	500	1	0.9	0.05	0.05	100	6	0.3	70	90	150
84	0	0	0	2000	500	2	0.9	0.05	0.05	100	6	0.3	70	90	150
85	0	0	0	2000	500	3	0.9	0.05	0.05	100	6	0.3	70	90	150
86	0	0	0	2000	500	4	0.9	0.05	0.05	100	6	0.3	70	90	150
87	0	0	0	2000	500	5	0.9	0.05	0.05	100	6	0.3	70	90	150

5.48 table

Output data

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
224	**83**	**22**	**11**	**132**	**712**	**421**	**2749**	**73**	**4**
225	84	22	11	90	697	421	1861	55	5
226	85	31	11	117	837	439	3313	18	2
227	86	27	11	232	760	433	2172	5	7
228	87	86	13	416	1270	456	2688	5	0



5.37 fig. Comparison of the results of simulations.

Results of simulations (5.48 tab.):

- Collisions were detected during almost every simulation. In simulation 224 the same parameters were used as in simulation 223, where no collisions were detected and the averaged number of trips was 116 trips. The only difference is that in simulations 223 no data transmission delay was used.
- No collisions were detected in the simulation 228, but the number of trips was also reduced, that is why for further simulations safety zone area rate = 1 was used and aim to minimize the number of collisions will be gained by changing values of the other parameters.

2.2.2. Maximal distance till crossing point

Maximal distance till other UEV to start crash prevention and to calculate collision probability is a next parameter to compare (5.49 tab.).

5.49 table

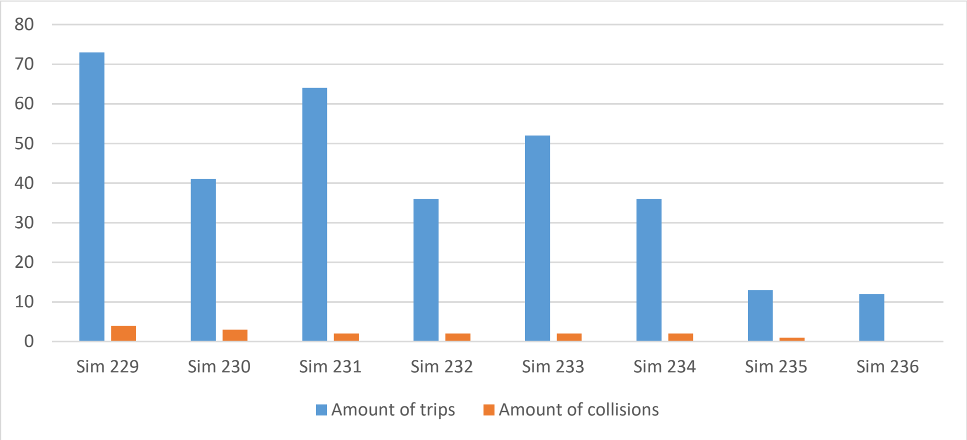
Parameters

ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
88	0	0	0	2000	500	1	0.9	0.05	0.05	100	6	0.3	70	90	150
89	0	0	0	2000	500	1	0.9	0.05	0.05	100	6	0.3	70	90	200
90	0	0	0	2000	500	1	0.9	0.05	0.05	100	6	0.3	70	90	250
91	0	0	0	2000	500	1	0.9	0.05	0.05	100	6	0.3	70	90	300
92	0	0	0	2000	500	1	0.9	0.05	0.05	100	6	0.3	70	90	350
93	0	0	0	2000	500	1	0.9	0.05	0.05	100	6	0.3	70	90	400
94	0	0	0	2000	500	1	0.9	0.05	0.05	100	6	0.3	70	90	450
95	0	0	0	2000	500	1	0.9	0.05	0.05	100	6	0.3	70	90	500

5.50 table

Output data

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
229	88	22	11	132	712	421	2749	73	4
230	89	32	11	100	890	429	1752	41	3
231	90	26	12	59	799	448	1374	64	2
232	91	30	11	129	795	440	2624	36	2
233	92	31	11	215	926	432	3798	52	2
234	93	30	11	129	795	440	2624	36	2
235	94	38	14	124	863	434	2181	13	1
236	95	48	21	113	1032	514	1858	12	0



5.38 fig. Comparison of the results of simulations.

Results of simulations (5.50 tab.):

- Collisions were detected in the simulations 229-235. No collisions and 12 trips were detected in simulation 236.
- Maximal distance = 500 was used for further simulations.

2.2.3. Collision probability weight

Collision probability weight was the next parameter to be changed (5.51 tab.).

5.51 table

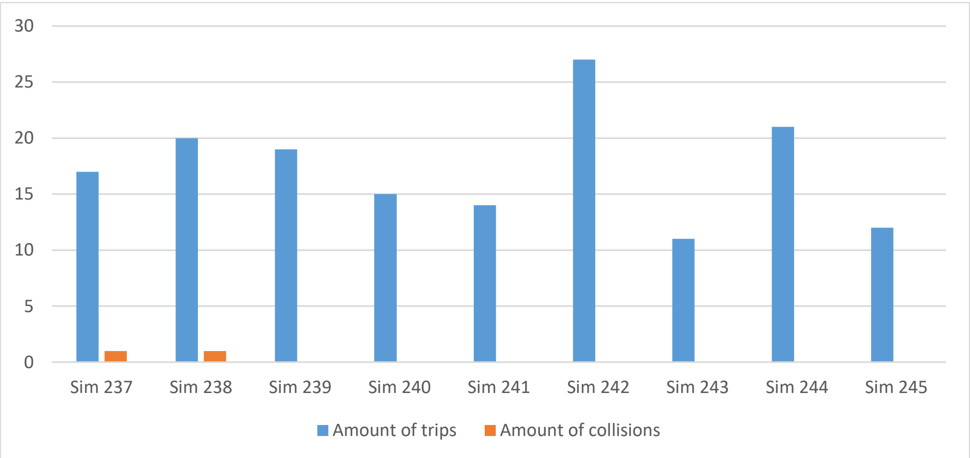
Parameters

ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
96	0	0	0	2000	500	1	0.1	0.45	0.45	100	6	0.1	70	90	500
97	0	0	0	2000	500	1	0.2	0.4	0.4	100	6	0.1	70	90	500
98	0	0	0	2000	500	1	0.3	0.35	0.35	100	6	0.1	70	90	500
99	0	0	0	2000	500	1	0.4	0.3	0.3	100	6	0.1	70	90	500
100	0	0	0	2000	500	1	0.5	0.25	0.25	100	6	0.1	70	90	500
101	0	0	0	2000	500	1	0.6	0.2	0.2	100	6	0.1	70	90	500
102	0	0	0	2000	500	1	0.7	0.15	0.15	100	6	0.1	70	90	500
103	0	0	0	2000	500	1	0.8	0.1	0.1	100	6	0.1	70	90	500
104	0	0	0	2000	500	1	0.9	0.05	0.05	100	6	0.1	70	90	500

5.52 table

Output data

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
237	96	61	14	644	1632	448	14537	17	1
238	97	41	11	286	942	445	5387	20	1
239	98	50	12	174	1198	446	4107	19	0
240	99	61	13	213	1247	438	3411	15	0
241	100	58	18	215	1284	459	2835	14	0
242	**101**	**27**	**11**	**209**	**716**	**422**	**2984**	**27**	**0**
243	102	42	13	128	945	454	3048	11	0
244	103	54	11	1026	1102	423	14017	21	0
245	104	48	21	113	1032	514	1858	12	0



5.39 fig. Comparison of the results of simulations.

Results of simulations (5.52 tab.):

- Collisions were detected during simulations 237 and 238.
- The best results were received in simulation 242, where no collisions were detected and 27 trips were detected, that is why collision probability weight 0.6 was used for further simulations.

2.3. With data transmission delays and data errors

Data transmission delays and data errors were used in these simulations.

2.3.1. Number of iterations

Number of iterations is the first parameter to be changed (5.53 tab.).

5.53 table

Parameters

ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
105	50	5	3	2000	500	1	0.6	0.2	0.2	30	6	0.1	70	90	500
106	50	5	3	2000	500	1	0.6	0.2	0.2	50	6	0.1	70	90	500
107	50	5	3	2000	500	1	0.6	0.2	0.2	80	6	0.1	70	90	500
108	50	5	3	2000	500	1	0.6	0.2	0.2	100	6	0.1	70	90	500
109	50	5	3	2000	500	1	0.6	0.2	0.2	150	6	0.1	70	90	500
110	50	5	3	2000	500	1	0.6	0.2	0.2	200	6	0.1	70	90	500

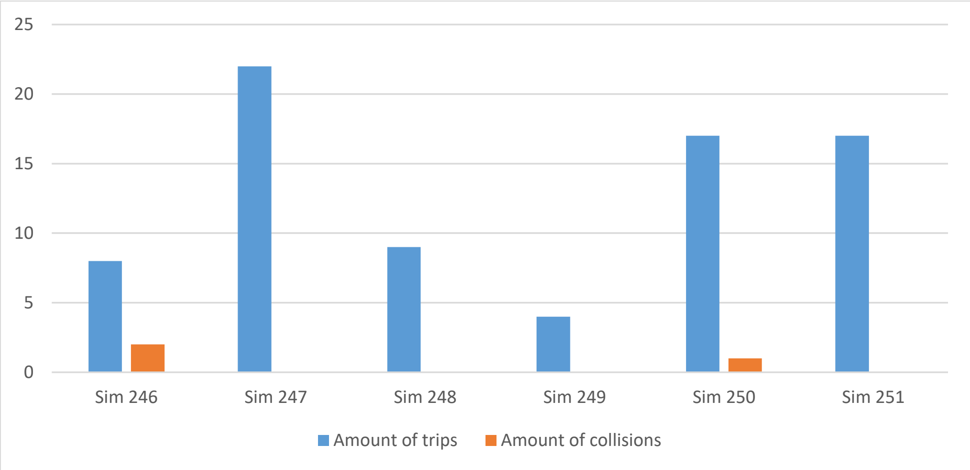
5.54 table

Output data

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
246	105	86	15	240	1527	506	4350	8	2
247	**106**	**52**	**11**	**148**	**1195**	**444**	**3962**	**22**	**0**
248	107	53	15	187	1188	562	3054	9	0
249	108	77	22	186	1666	735	3473	4	0
250	109	45	11	291	918	439	4249	17	1
251	110	51	11	321	1118	436	5862	17	0

Results of simulations (5.54 tab.):

- Results of simulations show that change of the possible number of iterations does not influence the output data a lot. This happens because of data transmission delays and data errors as well as random size of vehicles in each simulation process.
- Number of iterations 50 was used for further simulations.



5.40 fig. Comparison of the results of simulations.

2.3.2. Probability sensitivity

Probability sensitivity is the next parameter to be changed (5.55 tab.).

5.55 table

Parameters

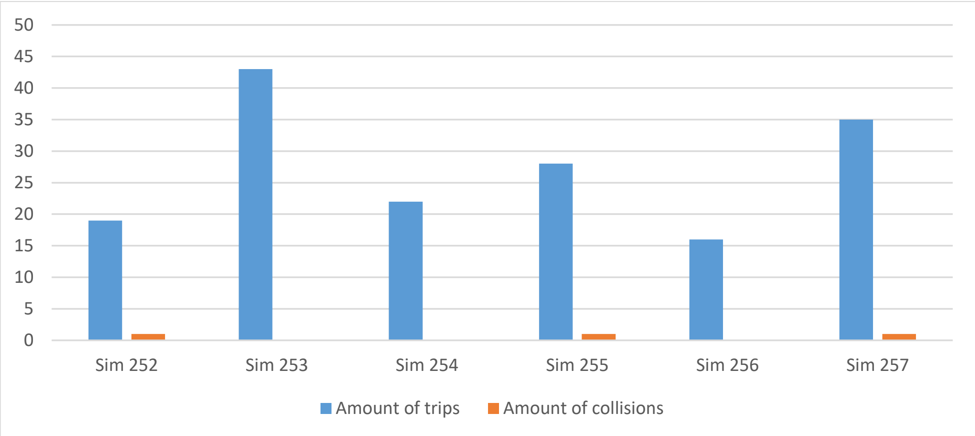
ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
111	50	5	3	2000	500	1	0.6	0.2	0.2	50	6	0.05	70	90	500
112	50	5	3	2000	500	1	0.6	0.2	0.2	50	6	0.1	70	90	500
113	50	5	3	2000	500	1	0.6	0.2	0.2	50	6	0.2	70	90	500
114	50	5	3	2000	500	1	0.6	0.2	0.2	50	6	0.3	70	90	500
115	50	5	3	2000	500	1	0.6	0.2	0.2	50	6	0.4	70	90	500
116	50	5	3	2000	500	1	0.6	0.2	0.2	50	6	0.5	70	90	500

5.56 table

Output data

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
252	111	68	17	248	1279	497	3830	19	1
253	**112**	**52**	**11**	**148**	**1195**	**444**	**3962**	**43**	**0**

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
254	113	71	11	329	1494	438	6027	22	0
255	114	63	11	500	1323	439	7286	28	1
256	115	48	14	146	933	446	2451	16	0
257	116	43	11	338	951	441	5768	35	1



5.41 fig. Comparison of the results of simulations.

Results of simulations (5.56 tab.):

- Results of simulations show that change of the probability sensitivity does not influence the output data. The same as in the previous simulations, this happens because of data transmission delays and data errors as well as random size of vehicles in each simulation process.
- Probability sensitivity 0.1 was used for further simulations.

2.3.3. Trajectory (angle)

Trajectory or angle change was the next parameter to be changed (5.57 tab.).

5.57 table

Parameters

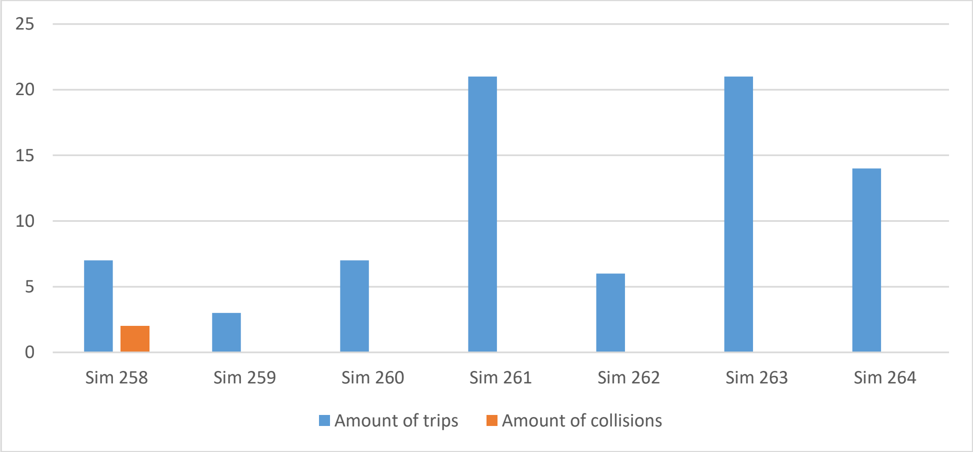
ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
117	50	5	3	2000	500	1	0.6	0.2	0.2	50	6	0.1	70	15	500

ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
118	50	5	3	2000	500	1	0.6	0.2	0.2	50	6	0.1	70	45	500
119	50	5	3	2000	500	1	0.6	0.2	0.2	50	6	0.1	70	60	500
120	50	5	3	2000	500	1	0.6	0.2	0.2	50	6	0.1	70	90	500
121	50	5	3	2000	500	1	0.6	0.2	0.2	50	6	0.1	70	120	500
122	50	5	3	2000	500	1	0.6	0.2	0.2	50	6	0.1	70	180	500
123	50	5	3	2000	500	1	0.6	0.2	0.2	50	6	0.1	70	360	500

5.58 table

Output data

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
258	117	56	11	240	895	434	2410	7	2
259	118	132	65	239	1696	772	2988	3	0
260	119	51	17	96	1046	462	2335	7	0
261	**120**	**52**	**11**	**148**	**1195**	**444**	**3962**	**21**	**0**
262	121	51	17	96	1046	462	2335	6	0
263	122	57	19	245	1343	475	4850	21	0
264	123	79	11	337	1878	435	7696	14	0



5.42 fig. Comparison of the results of simulations.

Results of simulations (5.58 tab.):

- Results of simulations show that change of the trajectory of motion does not influence the output data. The same as in the previous simulations.
- Possible trajectory change 90˚ was used for further simulations.

2.3.4. Maximal distance till crossing point

Maximal distance till other UEV to start crash prevention and to calculate collision probability is the next parameter to compare (5.59 tab.).

5.59 table

Parameters

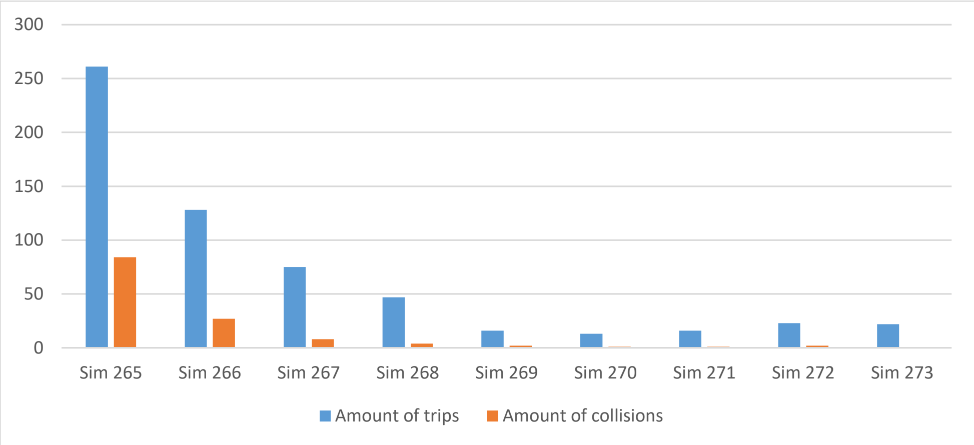
ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
124	50	5	3	2000	500	1	0.6	0.2	0.2	50	6	0.1	70	90	50
125	50	5	3	2000	500	1	0.6	0.2	0.2	50	6	0.1	70	90	100
126	50	5	3	2000	500	1	0.6	0.2	0.2	50	6	0.1	70	90	200
127	50	5	3	2000	500	1	0.6	0.2	0.2	50	6	0.1	70	90	250
128	50	5	3	2000	500	1	0.6	0.2	0.2	50	6	0.1	70	90	300
129	50	5	3	2000	500	1	0.6	0.2	0.2	50	6	0.1	70	90	350
130	50	5	3	2000	500	1	0.6	0.2	0.2	50	6	0.1	70	90	400
131	50	5	3	2000	500	1	0.6	0.2	0.2	50	6	0.1	70	90	450
132	50	5	3	2000	500	1	0.6	0.2	0.2	50	6	0.1	70	90	500

5.60 table

Output data

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
265	124	15	11	31	587	420	1093	261	84
266	125	19	11	45	646	421	1253	128	27
267	126	23	11	90	728	424	1658	75	8
268	127	23	11	65	720	423	1621	47	4
269	128	34	15	70	912	487	1809	16	2
270	129	39	11	116	930	440	2445	13	1
271	130	43	12	202	883	461	2779	16	1

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
272	131	48	17	229	1169	476	5793	23	2
273	**132**	**52**	**11**	**148**	**1195**	**444**	**3962**	**22**	**0**



5.43 fig. Comparison of the results of simulations.

Results of simulations (5.60 tab.):

- Collisions were detected in the simulations 265-272. No collisions and 22 trips were detected in simulation 273. Safety distance 500 was used in further simulations.

2.3.5. Collision probability weight

Collision probability weight was the next parameter to be changed (5.61 tab.).

5.61 table

Parameters

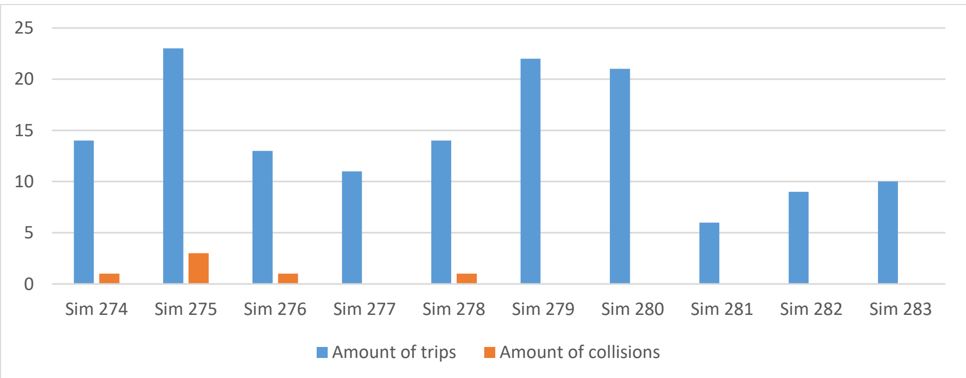
ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
133	50	5	3	2000	500	1	0.1	0.45	0.45	50	6	0.1	70	90	500
134	50	5	3	2000	500	1	0.2	0.4	0.4	50	6	0.1	70	90	500
135	50	5	3	2000	500	1	0.3	0.35	0.35	50	6	0.1	70	90	500
136	50	5	3	2000	500	1	0.4	0.3	0.3	50	6	0.1	70	90	500
137	50	5	3	2000	500	1	0.5	0.25	0.25	50	6	0.1	70	90	500

ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
138	50	5	3	2000	500	1	0.6	0.2	0.2	50	6	0.1	70	90	500
139	50	5	3	2000	500	1	0.7	0.15	0.15	50	6	0.1	70	90	500
140	50	5	3	2000	500	1	0.8	0.1	0.1	50	6	0.1	70	90	500
141	50	5	3	2000	500	1	0.9	0.05	0.05	50	6	0.1	70	90	500
142	50	5	3	2000	500	1	1	0	0	50	6	0.1	70	90	500

5.62 table

Output data

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
274	133	43	10	213	1114	435	5486	14	1
275	134	48	17	229	1169	476	5793	23	3
276	135	106	21	700	1991	624	9252	13	1
277	136	43	11	212	1027	427	4802	11	0
278	137	50	18	181	1082	458	3522	14	1
279	**138**	**52**	**11**	**148**	**1195**	**444**	**3962**	**22**	**0**
280	139	48	11	593	1073	428	9236	21	0
281	140	51	11	102	1021	430	1797	6	0
282	141	75	17	268	1318	435	4554	9	0
283	142	87	20	480	1458	456	7311	10	0



5.44 fig. Comparison of the results of simulations.

Results of simulations (5.62 tab.):

- Collisions were detected during simulations 274-276 and 278.
- The best results were received in simulation 279, where no collisions were detected and 22 trips were detected, that is why collision probability weight 0.6 was used for further simulations.

2.3.6. Errors and delays

The values of errors and delays were changed in next simulations, in purpose to check the behavior of the proposed INN system (5.63 tab.).

5.63 table

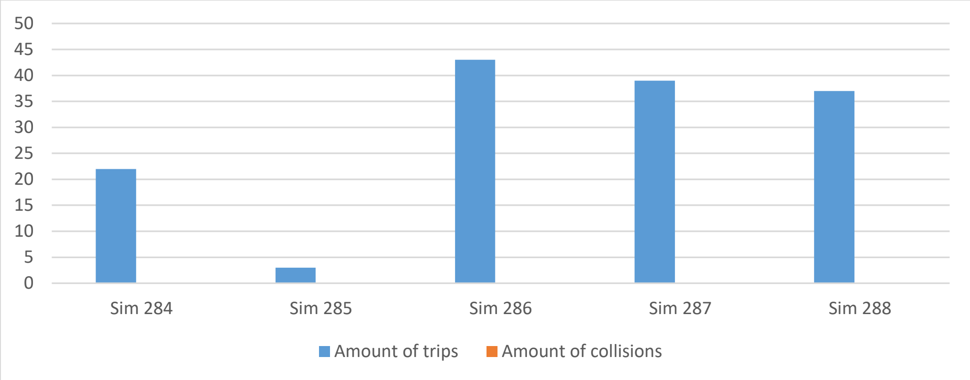
Parameters

ID	XYerr	Verr	Rerr	Txdelay	Sdelay	AreaRate	Ap	Av	Ar	TrLim	Sens	Psens	Vlim	Anglim	Safedist
143	50	5	3	2000	500	1	0.6	0.2	0.2	50	6	0.1	70	90	500
144	40	4	2.5	1500	400	1	0.6	0.2	0.2	50	6	0.1	70	90	500
145	30	3	2	1000	300	1	0.6	0.2	0.2	50	6	0.1	70	90	500
146	20	2	1.5	500	200	1	0.6	0.2	0.2	50	6	0.1	70	90	500
147	10	1	1	100	100	1	0.6	0.2	0.2	50	6	0.1	70	90	500

5.64 table

Output data

Simulation Nr.	ParamID	AvDur	SmDur	LngDur	AvDist	SmDist	LngDist	Trips	Collisions
284	**143**	**52**	**11**	**148**	**1195**	**444**	**3962**	**22**	**0**
285	144	112	53	220	2360	931	4494	3	0
286	145	27	11	98	821	424	2514	43	0
287	146	42	11	281	1108	433	5780	39	0
288	147	54	11	302	1250	434	5569	37	0



5.45 fig. Comparison of the results of simulations.

Results of simulations (5.64 tab.):

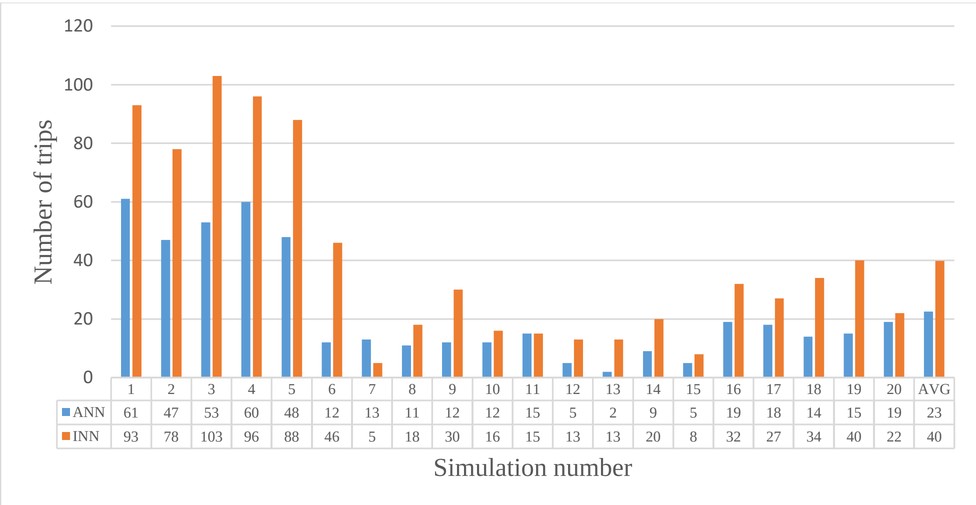
- Errors and delays were reduced in simulations 284-288. No collisions were detected. The number of trips was biggest in the last simulations, were errors and delays were small enough. But the result is not so stable as in the simulations with a fixed vehicle size.

5.7. Chapter 5 summary

Results of the experimental testing prove the efficiency of the proposed algorithms:

1. Experimental testing of the proposed algorithm of traffic light red signal recognition method showed that after the proposed system was trained to distinguish the red signal, it recognized red signal in real time without any mistakes.
2. Experimental testing of the proposed algorithm of CNN for object recognition showed that CNN is a suitable method for electrical transport dangerous situation recognition and prevention task. To minimize necessary calculation time, CNN must be trained in advance.
3. The results of the experimental testing of the proposed algorithm of electrical transport collision probability evaluation task showed, that collision probability depends on the distance till the crossing point, available time for the reaction and vehicles' speed.
4. Several experimental testings of the proposed algorithm of ANN for collision probability evaluation and minimization were done. They proved that ANN can be used in danger level estimation and minimization tasks. Several experimental testings of the proposed novel algorithm of the INN for unsupervised collision probability evaluation and minimization for electrical vehicle dangerous situation recognition and prevention task were done. Results of first and second experiments showed that ANN and proposed novel INN are both suitable methods for collision prevention task, but ANN takes much more time, to make necessary calculations, that influence the number of trips. Different input parameters were used in the second experiment:

a. Without data transmission delays and errors (5.46 fig. simulations 1-5). The output data shows that the number of collisions was reduced to zero during simulations with the ANN. However, the number of completed trips was much lower compared to the simulations without any motion control (5.8 table). For the simulations with the proposed INN, the same parameters were used. As a result, no collisions were detected, and the number of trips increased to an average of 92 trips per 10-minute simulation (5.9 table).
b. With data transmission delays, without errors (5.46 fig. simulations 6-10). During these simulations, a data transmission delay of 2000 ms and an own positioning data refresh delay of 500 ms were used. On average, 12 trips were completed during simulations with a traditional ANN (5.11 table), which is half as many as during simulations with the proposed INN (5.12 table). However, collisions were detected in all simulations, indicating that requirements need to be changed and other parameter values must be adjusted to ensure safe driving.
c. With data transmission delays, without errors, and with improved input parameters (5.46 fig. simulations 11-15). An increased value of the parameter “maximal distance till other UEV to start crash prevention and to calculate collision probability” was used during these simulations. The output data shows that although the number of trips was reduced, they were safer and without collisions. Moreover, the number of trips was twice as large during simulations with the proposed INN (5.14 table, 5.15 table).



5.46 fig. Comparison of the results of simulations.

d. With data transmission delays and errors (5.46 fig. simulations 15-20). During these simulations, data delays and errors were introduced to replicate the real-time experiment's conditions. No collisions were detected in any of the simulations. Furthermore, the number of trips completed was almost twice as high during simulations with the proposed INN (5.17 table, 5.18 table).

Considering that the main task is to provied safety electrical vehicle drive that is real time process, time for the reaction in one of the main parameters.

The results of the third experiment showed that it is necessary to choose appropriate parameters to achieve the best results. The best parameters are recognized to depend on the additional factors, and the following parameters are considered to be the best for the given additional factors:

- Ideal conditions (without errors or data transmission delays) for the collision prevention algorithm include the following parameters: weight for collision probability based on changes in movement parameters should be 0.2, weight for speed change should be 0.4, weight for trajectory change should be 0.4, maximum number of iterations should be 100, probability sensitivity should be 0.3, and the maximum distance to another UEV for starting crash prevention and calculating collision probability should be 150.
- With data transmission delays: collision probability weight according to the movement parameters change must be increased till 0.9, in turn, speed change weight and trajectory change weight must be decreased till 0.05 each. Maximal distance till other UEV to start crash prevention and to calculate collision probability must be increased till 500.
- With data transmission delays and errors: probability sensitivity must be increased till 0.5.

It is impossible to determine definitively which parameter values are best because the output data depends on unpredictable input parameters such as errors and delays. Additionally, the size of the UEV also influences the results, as a larger vehicle will have a larger crossing area. Nevertheless, the results of the experiments demonstrate that, despite different conditions, the proposed novel INN based algorithm is functional and the system is capable of self-learning and minimizing collisions between vehicles.

CONCLUSIONS

Analyzing the obtained results, it can be concluded that the goal of the doctoral thesis has been achieved. During the development of the doctoral thesis, the following has been done:

1. Industrial and scientific research analysis was done. It shows that despite the big number of inventions, the developed systems for unsupervised electrical vehicle control do not fulfil safety tasks completely. That is why the topic of the transport safety level increasing by using the artificial intelligence systems is actual and needs to be well studied.
2. Centralized, decentralized and distributed control system models were compared. The results of comparison show that distributed system is preferable than centralized or decentralized. Distributed models are easier to implement, they have less components, they are cheaper for infrastructure owner, and they are not connected to the specific area, they have also decreased time for reaction and decreased risk at system failure. That is why distributed system structure was used in this research.
3. The system structure of the proposed system was developed and described. All the functions are performed by the microcontroller or embedded computer, integrated in each electrical vehicle, where the object recognition process and risk assessment are done, as well as opportunity assessment and decision making about necessary movement parameters change are done. Such solution helps to minimize data processing time, because there is no need to transmit the data to the common center and backward.
4. System structure was divided into subsystems, for better understanding particular processes:
    a. Two subsystems were based on the known methods: artificial neural network (ANN) for supervised collision probability estimation and minimization, and convolutional neural network (CNN) for the object recognition task.
    b. The third one was developed by the author in the scope of this study: immune neural network (INN) based technology of machine learning for unsupervised safe vehicle control.

    In this research, traditional neural network is included to compare its results with those of the proposed novel INN immune neural network. The objective is to draw conclusions on whether the novel network is better or worse than the traditional one.
5. Developed mathematical models were also divided according to the provided tasks:
    a. models for objects and signals recognition task;
    b. models for collision probability evaluation and location of the possible crossing point calculation task;
    c. models for collision probability minimization task.
6. Several algorithms were developed.
    a. The proposed algorithm for immune neural network for unsupervised collision probability evaluation and minimization for electric transport dangerous situation recognition and prevention task is a novel algorithm and is foreseen to use for electrical vehicle unmanned control.

   b. All the other algorithms are used as helper methods in providing the autonomous safety drive of the electrical vehicles.
7. Several computer models and prototypes were developed and described in doctoral thesis to prove the workability of the developed algorithms. The proposed computer models were used to test the algorithms and to solve following tasks:
   a. traffic light red signal recognition;
   b. object recognition;
   c. collision probability evaluation and minimization;
   d. unsupervised collision probability evaluation and minimization.
8. An electrical circuit diagram of the collision prevention device with an unsupervised immune memory for unmanned electrical vehicle based on a single board computer was developed and described. The proposed electrical circuit was developed for the electrical vehicle – quadcopter, but it can be applied for other electyrical vehicles type also, because the developed collision prevention device is multifunctional and can be used with different types of electrical vehicles.
9. A comparison of ANN and INN based algorithms were done, considering the impact on traffic safety and necessary time for decision calculation, where INN presents better results described below. Results approve the proposed hypothesis - an immune neural network can make control decisions to prevent vehicle collisions with better performance than a traditional neural network in this task.

The experiments done during the doctoral work and the obtained results allow to make the following conclusions:

1. The proposed algorithm of traffic light red signal recognition method can distinguish red signal from other signals without mistakes, after system is trained.
2. CNN is a suitable method for object recognition process for electrical transport dangerous situation recognition and prevention task. CNN must be trained in advance to minimize necessary calculation time.
3. The collision probability depends on the distance till the crossing point, available time for the reaction and vehicles’ speed.
4. ANN method and algorithm are suitable for the collision probability evaluation and minimization for electric transport dangerous situation recognition and prevention task. It is possible to use previously trained ANN or to use self-training.
5. Novel INN based technology of machine learning for unsupervised safe vehicle control is also suitable for the collision probability evaluation and minimization for electric transport dangerous situation recognition and prevention task. Proposed INN does not need to be trained in advance. Collision probability minimization process can be started even with empty immune memory.
6. Proposed INN can be used for minimizing the collision probability, improving unsupervised transport safety and faster data processing in real time conditions with minimal deviation from task performance.

7. The proposed INN based algorithm is multifunctional and can be implemented into control systems of different types of electrical vehicles. Depending on the electrical vehicles specification, system can obtain different input parameters, such as speed, location (altitude, latitude, longitude) and trajectory of motion, and produce different output data, such as speed change or movement direction change.
8. Proposed INN is better than traditional ANN in electrical transport dangerous situation recognition and prevention task, because of the minimized number of detected collisions, that leeds to the safer transportation process. Result of computer simulation shows that during the experiment where UEV were able to change only its' speed, but not the trajectory of motion, 19 collisions were detected during 30 minutes long simulation with ANN and no collisions were detected during simulation with the proposed INN.
9. Proposed INN is better than traditional ANN in electrical transport dangerous situation recognition and prevention task, because of reduced calculation time, that leeds to the bigger number of safe trips. Results of computer simulations where UEV were able to change their speed and trajectory of motion:
    a. without data transmission delays and errors, show that the use of INN helps to increase number of trips by 70 % compared to the use of traditional ANN;
    b. with data transmission delays and inapproporiate maximal distance till other UEV to start crash prevention show that the use of INN helps to increase number of trips by 92 % compared to the use of traditional ANN and to decrease number of collisions by 25 % compared to the use of traditional ANN;
    c. with data transmission delays and approporiate maximal distance till other UEV to start crash prevention show that the use of INN helps to increase number of trips by 100 % compared to the use of traditional ANN;
    d. with data transmission delays and errors show that the use of INN helps to increase number of trips by 82 % compared to the use of traditional ANN.
10. INN system parameters also influence the result. It is impossible to determine definitively which parameter values are best because the output data depends on unpredictable input parameters such as errors and delays. INN system parameters must be adjustable depending on the situation.

Future research perspectives:

1. The theme of cybersecurity and loss of signal or communication was not considered in this research. It is considered as a prospect for future scientific research.
2. It is necessary to develop prediction algorithms for the location and velocity to continue the calculation if the data receiving is delayed.
3. Results of simulations show that the INN reduces the number of iterations and calculation time. It is necessary to analyze whether it will be sufficient for using low-powered systems.
4. It is necessary to make simulations by using multiple microcontrollers that will imitate UEVs, and to compare already received results with those ones.

REFERENCES

[1] S. De Nadai et al., "Enhancing safety of transport by road by on-line monitoring of driver emotions," 2016 11th System of Systems Engineering Conference (SoSE), Kongsberg, Norway, 2016, pp. 1-4, doi: 10.1109/SYSOSE.2016.7542941.

[2] A. Nikolajevs and M. Mezitis, "Level crossing time prediction," 2016 57th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON), Riga, Latvia, 2016, pp. 1-3, doi: 10.1109/RTUCON.2016.7763105.

[3] Sell, R., Nikitenko, A., Ziravecka, A., Berkolds, K., Vitols, K., Czekalski, P. Unmanned Electrical Vehicles and Autonomous System Simulation. Riga: RTU Press, 2021. 212 p. ISBN 978-9934-22-667-0. e-ISBN 978-9934-22-668-7.

[4] Repole, Donato. Research of Parallel Computing Neuro-fuzzy Networks for Unmanned Vehicles. PhD Thesis. Rīga: [RTU], 2021. 249 p.

[5] "Tesla Model S That Crashed Into Fire Truck Had Autopilot Engaged", 2019, [Online]. Available: https://www.consumerreports.org/car-safety/tesla-model-s-that-crashed-into-fire-truck-had-autopilot-engaged/.

[6] "Field testing a new delivery system with Amazon Scout", 2019, [Online]. Available: https://blog.aboutamazon.com/transportation/meet-scout.

[7] "'Project Titan' Apple Car may have wide sliding doors and an adaptive stability system", 2019, [Online]. Available: https://appleinsider.com/articles/19/08/20/project-titan-apple-car-may-have-wide-sliding-doors-and-an-adaptive-stability-system.

[8] "Current topics from the world of Audi", 2023, [Online]. Available: https://www.audi.com/en/experience-audi/mobility-and-trends/iaa-2019.html.

[9] "NVIDIA DRIVE End-to-End Platform for Software-Defined Vehicles", 2023, [Online]. Available: https://www.nvidia.com/en-us/self-driving-cars/drive-platform/.

[10] "40+ Corporations Working On Autonomous Vehicles", 2020, [Online]. Available: https://www.cbinsights.com/research/autonomous-driverless-vehicles-corporations-list/.

[11] "Dual Motor Model S and Autopilot", 2014, [Online]. Available: https://www.tesla.com/blog/dual-motor-model-s-and-autopilot.

[12] "Dual Motor Model S and Autopilot", 2019, [Online]. Available: https://www.forbes.com/sites/greggardner/2019/09/03/ntsb-finds-teslas-autopilot-failed-in-2018-crash-with-fire-truck/#53e34a2d6b96.

[13] "Driver Errors, Advanced Driver Assistance System Design, Led to Highway Crash", 2019, [Online]. Available: https://www.ntsb.gov/news/press-releases/Pages/NR20190904.aspx.

[14] "SharkSpotter combines AI and drone technology to spot sharks and aid swimmers on Australian beaches", 2018, [Online]. Available: http://theconversation.com/sharkspotter-combines-ai-and-drone-technology-to-spot-sharks-and-aid-swimmers-on-australian-beaches-92667.

[15] “Amazon delivered its first customer package by drone”, 2016, [Online]. Available: https://eu.usatoday.com/story/tech/news/2016/12/14/amazon-delivered-its-first-customer-package-drone/95401366/.

[16] “Safran group”, 2023, [Online]. Available: https://www.safran-group.com.

[17] “Tesla Deaths”, 2023, [Online]. Available: https://www.tesladeaths.com/.

[18] Kornejevs, A., Gorobecs, M. Neural Network Based UAV Optimal Control Algorithm for Energy Efficiency Maximization. In: 2020 IEEE 61st International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON 2020): Conference Proceedings, Latvia, Rīga, 5-7 November, 2020. Piscataway: IEEE, 2020, Article number 9316556. ISBN 978-1-7281-9511-7. e-ISBN 978-1-7281-9510-0. Available from: doi:10.1109/RTUCON51174.2020.9316556.

[19] Dong Hwa Kim and Kyu Young Lee, "Neural networks control by immune network algorithm based auto-weight function tuning" Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290), Honolulu, HI, USA, 2002, pp. 1469-1474 vol.2, doi: 10.1109/IJCNN.2002.1007734.

[20] Li, Y., Feng, W., Zhu, X.K., Tan, K.C., Guan, X. and Ang, K.H. (1998) PIDeasy and automated generation of optimal PID controllers. In: 3rd Asia-Pacific Conference on Control and Measurement, Dunhuang, China, 31 Aug - 4 Sep 1998, pp. 29-33.

[21] K. J. Åström and T. Hägglund. “PID Controllers: Theory, Design, and Tuning, 2nd Edition”. ISA, 1995.

[22] Haykin S.: Neural networks. A comprehensive foundation, Second edition. - Prenctice Hall, 2006 – 842 p.

[23] R. Pasti and L. Nunes de Castro, "A Neuro-Immune Network for Solving the Traveling Salesman Problem" The 2006 IEEE International Joint Conference on Neural Network Proceedings, Vancouver, BC, Canada, 2006, pp. 3760-3766, doi: 10.1109/IJCNN.2006.247394.

[24] D. Wang, C. Huo, Z. Tong, Y. Yang and Y. Wang, "Research on Vehicle Anti-collision Algorithm Based on Fuzzy Control," 2019 Chinese Control And Decision Conference (CCDC), Nanchang, China, 2019, pp. 2361-2366, doi: 10.1109/CCDC.2019.8833461.

[25] Z. Liu, J. Chen, F. Lan, H. Xia, “Methodology of hierarchical collision avoidance for high-speed self-driving vehicle based on motion-decoupled extraction of scenarios”. Published in IET Intelligent Transport Systems (Volume: 14 , Issue: 3 , 3 2020) 172-181 pp.

[26] C. Rosales, J. Gimenez, F. Rossomando, C. Soria, M. Sarcinelli-Filho and R. Carelli, "UAVs Formation Control With Dynamic Compensation Using Neuro Adaptive SMC," 2019 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 2019, pp. 93-99, doi: 10.1109/ICUAS.2019.8798282.

[27] L. Ling, Y. Niu and H. Zhu, "Lyapunov method-based collision avoidance for UAVs," The 27th Chinese Control and Decision Conference (2015 CCDC), Qingdao, China, 2015, pp. 4716-4720, doi: 10.1109/CCDC.2015.7162758.

[28] M.Gorobetz, Research of the genetic algorithms for optimal control of electric transport, doctoral thesis, Riga, 2008.

[29] S. Roelofsen, A. Martinoli and D. Gillet, "3D collision avoidance algorithm for Unmanned Aerial Vehicles with limited field of view constraints," 2016 IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, USA, 2016, pp. 2555-2560, doi: 10.1109/CDC.2016.7798647.

[30] R. Ke, Z. Li, J. Tang, Z. Pan and Y. Wang, "Real-Time Traffic Flow Parameter Estimation From UAV Video Based on Ensemble Classifier and Optical Flow," in IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 1, pp. 54-64, Jan. 2019, doi: 10.1109/TITS.2018.2797697.

[31] Samarjit Kar, Sujit Das, Pijush Kanti Ghosh, “Applications of neuro fuzzy systems: A brief review and future outline”, Applied Soft Computing, Volume 15, 2014, Pages 243-259, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2013.10.014.

[32] S.V. Ioannou, A.T. Raouzaiou, V.A. Tzouvaras, R.P. Mailis, K.C. Karpouzis, S.D.Kollias, Emotion recognition through facial expression analysis based on aneuro fuzzy network, Neural Networks 18 (2005) 423–435.

[33] N.M. Thanh, M.S. Chen, Image denoising using adaptive neuro-fuzzy system, IAENG International Journal of Applied Mathematics 36 (1) (2007).

[34] Y. Chakrapani, K. Soundararajan, Adaptive neuro-fuzzy inference systembased fractal image compression, International Journal of Recent Trends inEngineering 2 (1) (2009) 47–51.

[35] V. John, K. Yoneda, Z. Liu and S. Mita, "Saliency Map Generation by the Convolutional Neural Network for Real-Time Traffic Light Detection Using Template Matching," in IEEE Transactions on Computational Imaging, vol. 1, no. 3, pp. 159-173, Sept. 2015, doi: 10.1109/TCI.2015.2480006.

[36] M. Z. Abedin, P. Dhar and K. Deb, "Traffic Sign Recognition using SURF: Speeded up robust feature descriptor and artificial neural network classifier," 2016 9th International Conference on Electrical and Computer Engineering (ICECE), Dhaka, Bangladesh, 2016, pp. 198-201, doi: 10.1109/ICECE.2016.7853890.

[37] R. Qian, Q. Liu, Y. Yue, F. Coenen and B. Zhang, "Road surface traffic sign detection with hybrid region proposal and fast R-CNN," 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Changsha, China, 2016, pp. 555-559, doi: 10.1109/FSKD.2016.7603233.

[38] Z. Yi, Y. Wang, D. Tian, G. Lu and H. Xia, "A Road Safety Evaluation Method Based on Clustering Neural Network," 2010 International Conference on Optoelectronics and Image Processing, Haikou, China, 2010, pp. 108-111, doi: 10.1109/ICOIP.2010.60.

[39] L. Nassar and F. Karray, "Fuzzy Logic in VANET context aware Congested Road and Automatic Crash Notification," 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Vancouver, BC, Canada, 2016, pp. 1031-1037, doi: 10.1109/FUZZ-IEEE.2016.7737801.

[40] R. Qian, Y. Yue, F. Coenen and B. Zhang, "Traffic sign recognition with convolutional neural network based on max pooling positions," 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Changsha, China, 2016, pp. 578-582, doi: 10.1109/FSKD.2016.7603237.

[41] W. Zhang, L. Chen, W. Gong, Z. Li, Q. Lu and S. Yang, "An Integrated Approach for Vehicle Detection and Type Recognition," 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), Beijing, China, 2015, pp. 798-801, doi: 10.1109/UIC-ATC-ScalCom-CBDCom-IoP.2015.157.

[42] R. Ghosh, A. Mishra, G. Orchard and N. V. Thakor, "Real-time object recognition and orientation estimation using an event-based camera and CNN," 2014 IEEE Biomedical Circuits and Systems Conference (BioCAS) Proceedings, Lausanne, Switzerland, 2014, pp. 544-547, doi: 10.1109/BioCAS.2014.6981783.

[43] D. C. Ciresan, U. Meier, L. M. Gambardella and J. Schmidhuber, "Convolutional Neural Network Committees for Handwritten Character Classification," 2011 International Conference on Document Analysis and Recognition, Beijing, China, 2011, pp. 1135-1139, doi: 10.1109/ICDAR.2011.229.

[44] L. Chen, X. Guo and C. Geng, "Human face recognition based on adaptive deep Convolution Neural Network," 2016 35th Chinese Control Conference (CCC), Chengdu, China, 2016, pp. 6967-6970, doi: 10.1109/ChiCC.2016.7554454.

[45] A. Tuama, F. Comby and M. Chaumont, "Camera model identification with the use of deep convolutional neural networks," 2016 IEEE International Workshop on Information Forensics and Security (WIFS), Abu Dhabi, United Arab Emirates, 2016, pp. 1-6, doi: 10.1109/WIFS.2016.7823908.

[46] C. -D. Huang, C. -Y. Wang and J. -C. Wang, "Human action recognition system for elderly and children care using three stream ConvNet," 2015 International Conference on Orange Technologies (ICOT), Hong Kong, China, 2015, pp. 5-9, doi: 10.1109/ICOT.2015.7498476.

[47] H. Guan, W. Xingang, W. Wenqi, Z. Han and W. Yuanyuan, "Real-time lane-vehicle detection and tracking system," 2016 Chinese Control and Decision Conference (CCDC), Yinchuan, China, 2016, pp. 4438-4443, doi: 10.1109/CCDC.2016.7531784.

[48] S. Rawat, Z. A. Faridi and P. Kumar, "Analysis and proposal of a novel approach to collision detection and avoidance between moving objects using artificial intelligence," 2016 International Conference System Modeling & Advancement in Research Trends (SMART), Moradabad, India, 2016, pp. 135-138, doi: 10.1109/SYSMART.2016.7894505.

[49] J. A. Douthwaite, A. De Freitas and L. S. Mihaylova, "An interval approach to multiple unmanned aerial vehicle collision avoidance," 2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF), Bonn, Germany, 2017, pp. 1-8, doi: 10.1109/SDF.2017.8126384.

[50] R. Darío Fonnegra Tarazona, F. R. Lopera and G. -D. G. Sánchez, "Anti-collision system for navigation inside an UAV using fuzzy controllers and range sensors," 2014 XIX Symposium on Image, Signal Processing and Artificial Vision, Armenia, Colombia, 2014, pp. 1-5, doi: 10.1109/STSIVA.2014.7010153.

[51] G. Stana, P. Apse-Apsitis and V. Brazis, "Virtual energy simulation of induction traction drive test bench," 2014 IEEE 2nd Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE), Vilnius, Lithuania, 2014, pp. 1-6, doi: 10.1109/AIEEE.2014.7020330.

[52] Ribickis, L., Gorobecs, M., Ļevčenkovs, A. Neuro-Immune Algorithm for Embedded Real-Time Control System in Transport Safety Tasks. Frontiers in Artificial Intelligence and Applications, 2018, Vol. 310, pp.251-265. ISSN 0922-6389. e-ISSN 1879-8314. Available from: doi:10.3233/978-1-61499-929-4-251.

[53] Robert Fuller. 2001. Neuro-Fuzzy Methods for Modeling & Fault Diagnosis, Lisbon, Portugal, 77 pages.

[54] K. Li, "The Challenges and Potential of Risk Assessment for Active Safety of Unmanned Tram," 2018 International Conference on Control, Automation and Information Sciences (ICCAIS), Hangzhou, China, 2018, pp. 22-27, doi: 10.1109/ICCAIS.2018.8570696.

[55] R. Ke, Z. Li, J. Tang, Z. Pan and Y. Wang, "Real-Time Traffic Flow Parameter Estimation From UAV Video Based on Ensemble Classifier and Optical Flow," in IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 1, pp. 54-64, Jan. 2019, doi: 10.1109/TITS.2018.2797697.

[56] Korņejevs, A., Gorobecs, M., Alps, I., Ribickis, L. Adaptive Traction Drive Control Algorithm for Electrical Energy Consumption Minimisation of Autonomous Unmanned Aerial Vehicle. Electrical, Control and Communication Engineering, 2019, Vol. 15, No. 2, pp. 62-70. ISSN 2255-9140. e-ISSN 2255-9159. Available from: doi:10.2478/ecce-2019-0009.

[57] Strupka, G. Algorithm for Unmanned Aerial Vehicle to Supervise Applications for Civil and Power Engineering Tasks. In: 17th International Symposium "Topical Problems in the Field of Electrical and Power Engineering" and "Doctoral School of Energy and Geotechnology III": Proceedings, Estonia, Kuressaare, 15-20 January, 2018. Tallinn: Tallinn University of Technology, 2018, pp.149-152. ISBN 978-9949-832-13-2.

[58] M. Mofaddel, D. Tavangarian, A Distributed System with a Centralized Organization, Rostock, Germany, 1997.

[59] R. Dewan, N. Pahuja, S. Kukreja, Distributed operating system – an overview, International Journal of Research (IJR) Vol-1, Issue-10, 2014.

[60] P. Graybeal, M.Franklin, D. Cooper, Principles of accounting, Volume 2: Managerial Accounting, 2019.

[61] Alom, Md Z.; Taha, Tarek M.; Yakopcic, Chris; Westberg, Stefan; Sidike, Paheding; Nasrin, Mst S.; Hasan, Mahmudul; Van Essen, Brian C.; Awwal, Abdul A.S.; Asari,

Vijayan K. 2019. "A State-of-the-Art Survey on Deep Learning Theory and Architectures" Electronics 8, no. 3: 292. https://doi.org/10.3390/electronics8030292.

[62] Hinton, G.E.; Osindero, S.; Teh, Y.-W. A fast learning algorithm for deep belief nets. Neural Comput. 2006, 18, 1527–1554.

[63] Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 807–814.

[64] Hu, Yancai & Park, Gyei-Kark. (2020). Collision risk assessment based on the vulnerability of marine accidents using fuzzy logic. International Journal of Naval Architecture and Ocean Engineering. 12. 541-551. 10.1016/j.ijnaoe.2020.06.005.

[65] Sahin, Bekir & Senol, Yunus. (2014). A Novel Process Model for Marine Accident Analysis by using Generic Fuzzy-AHP Algorithm. Journal of Navigation. 68. 162-183. 10.1017/S0373463314000514.

[66] Sahin, Bekir & Yip, Tsz Leung. (2017). Shipping technology selection for dynamic capability based on improved Gaussian fuzzy AHP model. Ocean Engineering. 136. 233-242. 10.1016/j.oceaneng.2017.03.032.

[67] Chen, T., Grabs, E., Pētersons, E., Efrosinin, D., Ipatovs, A., Bogdanovs, N., Rjazanovs, D. Multiclass Live Streaming Video Quality Classification Based on Convolutional Neural Networks. Automatic Control and Computer Sciences, 2022, Vol. 54, No. 5, pp.455-466. ISSN 0146-4116. e-ISSN 1558-108X. Available from: doi:10.3103/S0146411622050029.

[68] Grabs, E., Pētersons, E., Efrosinin, D., Ipatovs, A., Klūga, J., Sturm, V. Accuracy evaluation of supervised machine learning classification models for wireless network traffic. International Journal of Communication Networks and Distributed Systems, 2022, Vol. 28, No. 6, pp.655-678.

[69] Bondarenko, Andrejs. Development of Knowledge Extraction Methodology from Trained Artificial Neural Networks. PhD Thesis. Rīga: [RTU], 2020. 158 p.

[70] Namatevs, I., Sudars, K., Polaka, I. Automatic Data Labeling by Neural Networks for the Counting of Objects in Videos. In: Procedia Computer Science. Vol.149: ICTE in Transportation and Logistics 2018 (ICTE 2018), Lithuania, Klaipeda, 1-1 January, 2019. Amsterdam: Elsevier B.V., 2019, pp.151-158. ISSN 1877-0509. Available from: doi:10.1016/j.procs.2019.01.118.

[71] Kadikis, Roberts. Efficient Methods for Detection and Characterization of Moving Objects in Video. PhD Thesis. Rīga: [RTU], 2018. 132 p.

[72] Kreicbergs, J., Smirnovs, J., Lama, A., Smirnovs, J., Zariņš, A. Road Traffic Safety Development Trends in Latvia. The Baltic Journal of Road and Bridge Engineering, 2021, Vol. 16, No. 4, pp.58-78. ISSN 1822-427X. e-ISSN 1822-4288. Available from: doi:10.7250/bjrbe.2021-16.539.

[73] Amare, V., Smirnovs, J. Road Traffic Safety Analysis of Different Junction Types on the State Roads. IOP Conference Series: Materials Science and Engineering, 2021, Vol.

1202, Article number 012034. ISSN 1757-8981. e-ISSN 1757-899X. Available from: doi:10.1088/1757-899X/1202/1/012034/pdf.

[74] Bitins, A., Maklakovs, J., Bogdane, R., Chatys, R., Sestakovs, V. Using Adverse Event Pyramids to Assess Probabilities in Airline Safety Management. Transactions on Aerospace Research, 2021, No. 2, pp. 71-83. e-ISSN 2545-2835. Available from: doi:10.2478/tar-2021-0012.

[75] Jasiuniene, V., Cygas, D. Analysis of Older Pedestrian Accidents: A Case Study of Lithuania. The Baltic Journal of Road and Bridge Engineering, 2020, Vol. 15, No. 1, pp. 147-160. ISSN 1822-427X. e-ISSN 1822-4288. Available from: doi:10.7250/bjrbe.2020-15.465.

[76] Jasiuniene, V., Vaiskunaite, R. Road Safety Assessment Considering the Expected Fatal Accident Density. The Baltic Journal of Road and Bridge Engineering, 2020, Vol. 15, No. 2, pp. 31-48. ISSN 1822-427X. e-ISSN 1822-4288. Available from: doi:10.7250/bjrbe.2020-15.471.

[77] Dung, N. Developing Models for Managing Drones in the Transportation System in Smart Cities. Electrical, Control and Communication Engineering, 2019, Vol. 15, No. 2, pp. 71-78. ISSN 2255-9140. e-ISSN 2255-9159. Available from: doi:10.2478/ecce-2019-0010.

[78] Buss, D., Abishev, K., Baltabekova, A. Driver's Reliability and Its Effect on Road Traffic Safety. In: Procedia Computer Science. Vol.149: ICTE in Transportation and Logistics 2018 (ICTE 2018), Lithuania, Klaipeda, 1-1 January, 2019. Amsterdam: Elsevier B.V., 2019, pp.463-466. ISSN 1877-0509. Available from: doi:10.1016/j.procs.2019.01.163.

[79] Freimane, J., Mezitis, M., Mihailovs, F. Maneuver Movements' Safety Increase Using Maneuver Locomotive Identification and Distance Control. Procedia Computer Science, 2017, Vol. 104, pp.375-379. ISSN 1877-0509. Available from: doi:10.1016/j.procs.2017.01.148.

[80] Gorobecs, M., Ļevčenkovs, A. Intelligent Electric Vehicle Motion and Crossroad Control Using Genetic Algorithms. In: Proceedings of 10th International Conference on Intelligent Technologies in Logistics and Mechatronics Systems (ITELMS), Lithuania, Panevezys, 21-22 May, 2015. Panevezys: 2015, pp.1-6.

[81] “Elon Musk says next year's Tesla cars will be able to self-drive 90 percent of the time”, 2014, [Online]. Available: https://www.theverge.com/2014/10/2/6894875/elon-musk-says-next-years-tesla-cars-will-be-able-to-self-drive-90-percent-of-the-time.

[82] “Elon Musk says Tesla’s fully autonomous cars will hit the road in 3 years”, 2015, [Online]. Available: https://www.businessinsider.com/elon-musk-on-teslas-autonomous-cars-2015-9.

[83] “Elon Musk’s road to Twitter is paved with broken promises”, 2022, [Online]. Available: https://www.washingtonpost.com/technology/2022/04/15/elon-musk-promises/.

[84] "Elon Musk is 'extremely confident' Tesla will release full autonomy in 'some jurisdictions' next year", 2020, [Online]. Available: https://electrek.co/2020/12/02/elon-musk-extremely-confident-tesla-release-full-autonomy-some-jurisdictions-2021/.

[85] "Elon Musk is once again promising Teslas will drive themselves in the near future — a claim he's been making since at least 2015", 2022, [Online]. Available: https://www.businessinsider.com/elon-musk-history-of-full-self-driving-promise-2022-1.

[86] "Elon Musk Promises Full Self-Driving Teslas in 2022", 2022, [Online]. Available: https://www.thestreet.com/lifestyle/cars/elon-musk-promises-full-self-driving-teslas-in-2022.

[87] Mike Erskine, David Milburn. Digital Train Control Functional Safety For AI Based Systems. In: Proceedings of International Railway Safety Council Conference 2019 13-18 October 2019, Perth, Australia.

[88] Railway Gazette, SNCF targets autonomous trains in five years, September 2018.

[89] "Germany to introduce driverless trains by 2023", 2016, [Online]. Available: https://www.themanufacturer.com/articles/germany-to-introduce-driverless-trains-by-2023/.

[90] Smart Rail World, DB tells staff and unions to prepare for driverless operations by 2021, June 2016.

[91] Simmons A, Furness N, The main line ATO journey, IRSE News, Issue 251, January 2019.

[92] "Latvijas nacionālais terminoloģijas portāls", 2023, [Online]. Available: https://termini.gov.lv/.

Anna Beinaroviča was born in 1989 in Daugavpils. She obtained a Professional Bachelor's degree and an engineer's qualification in Railway Transport (2014) and a Professional Master's degree in Railway Transport (2016) from Riga Technical University.
From 2016 to 2020, she was the chief specialist of the Business Process Research and Development Department at LDZ CARGO Ltd. From 2017 to 2020, she was a research assistant at Riga Technical University. Currently, Anna Beinaroviča is a systems analyst at Meditec Ltd and a research assistant at Riga Technical University.
In 2017, she enrolled in the professional development programme "Programming languages C/C++" at "ITLAT Mācību centrs" Ltd. In 2022, she received a certificate of Certified Business Analysis Professional (CBAP) from the International Institute of Business Analysis (IIBA). Anna Beinaroviča is a co-author of 15 publications indexed in Web of Science and Scopus.
Her scientific interests are related to the application of artificial intelligence in transport systems, ensuring safe unmanned movement.