Machine Learning Algorithm of Immune Neuro-Fuzzy Anti-collision Embedded System for Autonomous Unmanned Aerial Vehicles Team

Mikhail Gorobetz
Institute of Industrial Electronics and Electrical Engineering
Riga Technical University, Riga, Latvia
Mihails.Gorobecs@rtu.lv

Leonids Ribickis
Institute of Industrial Electronics and Electrical Engineering
Riga Technical University, Riga, Latvia
Leonids.Ribickis@rtu.lv

Anatoly Levchenkov
Institute of Industrial Electronics and Electrical Engineering
Riga Technical University, Riga, Latvia
Anatolijs.Levcenkovs@rtu.lv

Anna Beinarovica
Institute of Industrial Electronics and Electrical Engineering
Riga Technical University, Riga, Latvia
Anna.Beinarovica@rtu.lv

ABSTRACT

This study is dedicated to solve a collision prevention task for autonomous unmanned aerial vehicles' (UAVs) team by Immune Neuro-Fuzzy Network (INFN) application. It is a part of the project aimed at the development of intelligent safety and optimal control systems of autonomous electric vehicles and transport in general.

The goal of the current research is to develop the machine learning algorithm for autonomous UAV, that will give possibility for UAVs to train themselves without a teacher to avoid the collisions in the most effective way by changing the UAVs trajectory of the flight and without human intervention, i.e. the system should be self-organized. For this purpose, authors have improved previously developed immune neuro-fuzzy logic method to minimize collision probability. The experiments prove the workability and advantages of the developed algorithm.

CCS CONCEPTS

• Computing methodologies • Machine learning approaches • Bio-inspired approaches

KEYWORDS

Machine learning, neural network, fuzzy logic, immune network, anti-collision system, autonomous vehicle, unmanned aerial vehicle, vehicle's team

APPIS 2019, January 7–9, 2019, Las Palmas de Gran Canaria, Spain

ACM ISBN 978-1-4503-6085-2/19/01...$15.00
https://doi.org/10.1145/3309772.3309797

1 Introduction

Nowadays UAVs are available almost to everyone. It provides bigger demand and development of this product. One of the possibilities to use UAV is entertainment or personal use. For this purpose, in most cases, man-operator, who sets the speed and the trajectory of motion by using the remote control, controls the aerial vehicle. Another possibility to use UAV is different industrial purposes. As we know, in the world, the desire to optimize work prevails and the main goal is to reduce costs and increase profits. The autonomous system is one of the tools to achieve this task and UAVs are not an exception. For example in [1] authors propose to use UAVs for unmanned security and surveillance operations. Also UAVs can be used to deliver goods [2] or parcels in case of emergency situations [3]. The use of UAVs continue to spread to different areas.

Scientists and developers use different methods to gain autonomous UAV drive. For example in [4] based on back propagation neural networks, an adaptive PID controller is proposed for quadrotor UAV with unknown variable payloads. But authors of [5] have designed a self-learning fuzzy autopilot using the fuzzy C-means clustering technique for controlling the position of a hexacopter drone. In difference with those researches in this paper neural network[6] is combined with fuzzy logic[7], that is used for the risk assessment and immune memory, which helps to remember the situations and to find situation with the similar parameters in the database when it is necessary.

2 Problem Formulation

Another problem appears – how to avoid collision between multiple UAVs with different goals working in the same area. Authors of the [8] paper made a research to solve the multiple UAVs autonomous formation problem by study the wolf pack mechanism. They used the hierarchy behavior in wolf pack to analyze the bionic mechanism of UAVs formation flight. This

paper provides an algorithm with no any hierarchy or any leader. Each UAV is autonomous and makes decisions only for himself.
Previously authors of this research have described control algorithms for UAVs collision prevention by changing the altitude [9] or the motion speed [10] of the flight. Three computer experiments, based on different methods, were made in [10]: no any motion control was used, neural network (NN) was used for motion control and immune memory was not applicable, immune neural network (INN) with immune memory was used for motion control. For the first experiment lots of collisions were detected. At the beginning of second and third experiments the number of iterations for both structures were approximately the same, because at the beginning of simulation INN was untrained and no much data of the best solutions were in the immune memory. When training was done and immune memory was full enough with good examples, INN found out better solutions and updated the immune memory. At the end of experiment the results showed, that NN needs more iterations as INN to solve the same task, this significantly increases processing time and leads to a bigger amount of collisions.
The algorithm proposed in this paper presents the next evolution step of the authors' developed methods for transport anti-collision tasks providing the anti-collision solution by optimal change of the motion trajectory. UAVs have different target points, which are placed outside of the crossing point area, as it is shown in the figure 1.
The mathematical model for collision prevention for multiple UAVs working in the same area by changing trajectory of flight is developed, target function for unsupervised machine learning of the immune neuro-fuzzy control system is defined. While fuzzy logic provides an inference mechanism under cognitive uncertainty in reactions to the UAVs' team situation danger level, computational immune neural networks offer exciting advantages, such as learning, adaptation, fault tolerance, parallelism and generalization [11].

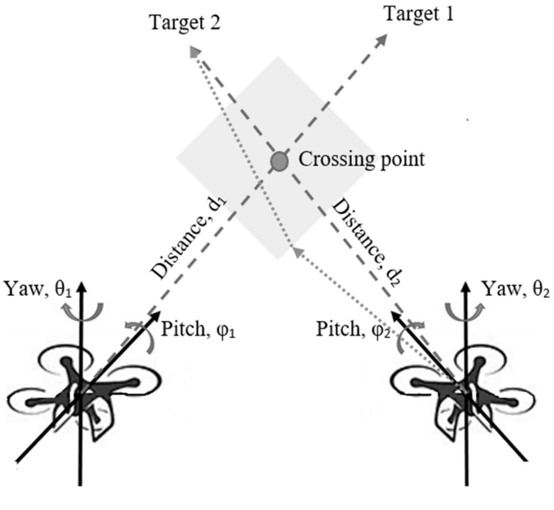

Figure 1: UAV collision prevention problem formulation

Proposed system structure is not centralized and is distributed among the UAVs of the team, so each autonomous device should make an optimal decision between safety and performance criteria, i.e. for safe trajectory change leading towards the team goal achievement. Each UAV is communicating with other team members in the defined area and learn to make better decisions by unsupervised machine learning immune neuro-fuzzy algorithm. The algorithm is implemented in the developed embedded device of each UAV that corrects the flight trajectory and sets necessary rotation speed of electric motors to avoid the collisions. Experiments show that INFN helped to minimize amount of collisions of UAV's team members and this proves the effectiveness of the proposed algorithm.

3 Anti-collision and INN System Structure

The proposed unsupervised machine learning anti-collision system structure is shown in Figure 2.
Abbreviations, mentioned in Figure 2, are explained in table 1.

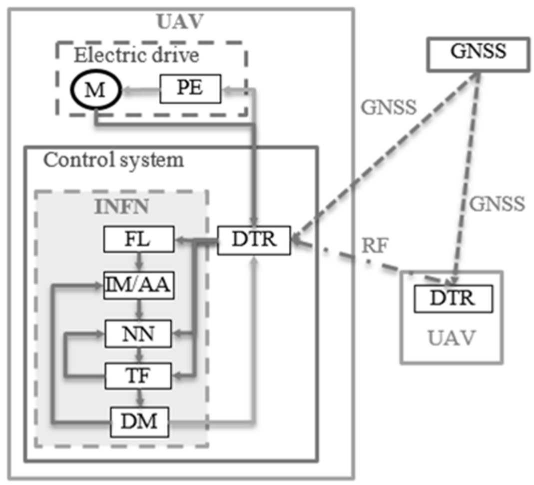

Figure 2: Anti-collision system structure

INFN performs the collision prevention function by the following steps:
Step 1. Risk assessment is made by using FL. If the collision probability is higher than specified, then Step 2 is done and necessary trajectory changes are calculated by applying INFN, otherwise no further calculations and changes are needed.
Step 2. The AA compares the similarity of the current situation with all situations that stored in the IM and finds the situation with the similar parameters and lowest discrepancy.
Step 3. INFN specialized neurons use stored weights of the found similar situation and calculate changes of the UAV flight yaw and pitch angles.
Step 4. TF calculates the collision probability Pmax of UAV taking in account changed yaw and pitch angles of the flight.

Table 1: Defined variables and abbreviations

Abbr.	Explanation
V_i	speed of the UAV
θ_i	yaw angle (azimuth, bearing, heading) of the UAV flight. An angle in a horizontal plane in relation to the lines of meridian (north–south lines). The units are degrees from north in a clockwise direction. North is 0°, east is 90°, south is 180°, and west is 270°
φ_i	pitch angle of the UAV flight. It is the angle between the horizontal axis and the longitudinal axis of the aircraft. Range from -90° to +90° where 0° = Horizon, +90° = vertically up and –90° = vertically down
χ_i	latitude χχ of the UAV position
ψ_i	longitude ψψ of the UAV position
η_i	altitude ηη of the UAV position
$\Delta\theta$	necessary changes of the yaw angle of the UAV flight
$\Delta\varphi_i$	necessary changes of the pitch angle of the UAV flight
d_i	distance of i-th UAV till the possible collision point
α	number of the situation. Situations are saved in the database and each of them has own serial number.
β	new values of the weights
μ	neurons with immune memory
X	Antigen
Pmax	collision probability
N	output layer neurons
AA	affinity algorithm
IM	immune memory
TF	target function
INFN	immune neuro-fuzzy network
NN	single layer adaptive neural network
FL	fuzzy logic
TA	training algorithm
DM	decision making algorithm
DTR	control components to obtain the location, to calculate the motion parameters, to communicate with other devices and to control the electric drive of UAV
UAV	unmanned aerial vehicle
GNSS	satellite positioning system
RF	radio frequency modules
PE	power electronics elements
M	electric motor

Step 5. DM evaluates the found solution by multiple criteria optimization minimizing of collision probability Pmax and UAV's flight yaw and pitch angles change.

Step 6. If the criteria are satisfied (Pmax is close to zero and yaw and pitch angles changes are as small as possible), data about the situation and INFN weights are saved in the IM and necessary angles changes yaw and pitch are sent to the DTR device.

Step 7. DTR controls the UAVs electric drive (PE and M) to change the trajectory of the flight. It is possible if the capacity UAV energy storage is sufficient [12] that should be measured using methods with increased precision [13] as well as power electronic components are working properly [14]. It has the influence on the safety of UAV team, especially in case, if the UAV plays the role of the coordinator, it transmits the information about the suggested changes to other vehicles.

4 Mathematical model of anti-collision immune neural network and fuzzy logic

The mathematical model of the UAVs' team represents the set of autonomous unmanned aerial vehicles whose actions agree with certain rules and with only common interests.

The set of UAVs is given:

$$UAV_s = (UAV_1, \dots, UAV_n) \tag{1}$$

The utility function as a common interest function is:

$$U = w(x, a_1, a_2, \dots, a_n) \to opt \tag{2}$$

Where,
U – utility function – common interest function
x – state of the environment
a_i – action of the i-th UAV

The information of each UAV depends on the state of environment

$$y_i = \gamma_i(x) \tag{3}$$

Where,
y_i – information of i-th UAV.

The decision rule of i-th UAV results an action of i-th quadcopter and depends on the information:

$$a_i = \rho_i(y_i) \tag{4}$$

Where,
ρ_i – decision rule of i-th UAV.

Interaction between i-th and j-th UAV:

$$q_{ij} = \partial \mathrm{w} / \partial \mathrm{a}_i \, \partial \mathrm{a}_j \tag{5}$$

A set of decision rules is optimal if
$E(w(x, (\rho_1(y_1), \dots, \rho_n(y_n))) \to$ max for a given probability distribution on x.

For anti-collision test the set of possible points of potential collisions is defined:

$$P = (p_1, p_2, \dots, p_c) \tag{6}$$

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

$$\begin{aligned} \chi_c^{UAVS} &= \{\chi_c^{UAV_1}, \chi_c^{UAV_2}, \dots, \chi_c^{UAV_n}\}, \\ \psi_c^{UAVS} &= \{\psi_c^{UAV_1}, \psi_c^{UAV_2}, \dots, \psi_c^{UAV_n}\}, \\ \eta_c^{UAVS} &= \{\eta_c^{UAV_1}, \eta_c^{UAV_2}, \dots, \eta_c^{UAV_n}\} \end{aligned} \tag{7}$$

Where,
χ_c^{UAV} – latitude of the current point;
ψ_c^{UAV} – longitude of the current point;
η_c^{UAV} –altitude of the current point.

The trajectory T^{UAVS} of UAVs is represented by two subsets $< \Theta^{UAVS}, \Phi^{UAVS} >$, that are yaw angle of the flight θ and pitch angle of the flight φ:

$$\Theta^{UAVS} = \{\theta^{UAV_1}, \theta^{UAV_2}, \dots, \theta^{UAV_n}\},$$
$$\Phi^{UAVS} = \{\varphi^{UAV_1}, \varphi^{UAV_2}, \dots, \varphi^{UAV_n}\} \quad (8)$$

Where,
θ^{UAV_i} – yaw angle of i-th UAV flight;
φ^{UAV_i} – pitch angle of i-th UAV flight.
The safety criterion is following:

$$D = |UAV_i UAV_j| = \sqrt{(\chi_c^j - \chi_c^i)^2 + (\psi_c^j - \psi_c^i)^2 + (\eta_c^j - \eta_c^i)^2} > D_{safe} \quad (9)$$

Where,
D_{safe} is safety distance limit for each pair of $< UAV_i, UAV_j >$, $i = 1..n$, $j = 1..n$, $i \neq j$
Restrictions for the pitch angle change were also defined. Permissible angle values of the flight depend on the UAV specifications, wind speed and other circumstances. For this research the most common permissible angles were taken:
$-35° < \varphi^{UAV_i} < 35°$
Restrictions for the yaw angle were also defined. These restrictions do not influence the quality of the flight, as do restrictions of the pitch angle, but it helps not to get too far from the target point. For example, the collision may be easily prevented if drone will flight backwards, but this will significantly increase distance till target point. So following restrictions for the yaw angles were defined: $-45° < \Delta\theta_i < +45°$
So, the common target function with anti-collision criteria is following:

$$\begin{cases} P_{max}(\chi_c^{UAVS}, \psi_c^{UAVS}, \eta_c^{UAVS}, \Delta\Theta, \Delta\Phi, \Delta V) = \max(P_{IJ}) \rightarrow \text{I} \\ \Delta\Theta_\Sigma(\Delta\Theta) = \sum_{i=1}^{m} \Delta\Theta_i \rightarrow \min \\ \Delta\Phi_\Sigma(\Delta\Phi) = \sum_{i=1}^{m} \Delta\Phi_i \rightarrow \min \\ \Delta V_\Sigma(\Delta V) = \sum_{i=1}^{m} \Delta V_i \rightarrow \min \\ D = |UAV_i UAV_j| > S \\ -45° < \Delta\theta_i < 45° \\ -35° < \varphi^{UAV_i} + \Delta\varphi_i < 35° \\ i = 1..n, \\ j = 1..n, \\ i \neq j \end{cases} \quad (10)$$

Where,
P_{max} - the maximum collision probability
$\Delta\Theta = (\Delta\theta_1, \dots, \Delta\theta_n)$- set of yaw angle changes of all UAVs
$\Delta\Phi = (\Delta\varphi_1, \dots, \Delta\varphi_n)$ - set of pitch angle changes of all UAVs
$\Delta V = (\Delta V_1, \dots, \Delta V)$ - set of speed changes of all UAVs
PIJ = (P(<UAV_1, UAV_2>),…,P(<UAV_i, UAV_j>),…,P(<UAV_{n-1}, UAV_n>)) - set of probabilities of collision for all pairs of UAVs <UAV_i, UAV_j>, $i \neq j$, i,j = 1..n

The defined target function (10) describes the common target for all UAV team members. It is possible, if the team has the coordinator that makes a decision for all participants.
But usually each UAV makes decision about its own actions and can not decide for another member. In that case the INFN is distributed between all UAVs, and each UAV has own parts of the network. It allows to reduce the dimensions of the task by solving it using multi-processor approach.
So, each processor, i.e. each i-th UAV is looking for its own trajectory change solution $< \Delta\theta_i, \Delta\varphi_i, \Delta V_i >$.
To generalize the situation description for the neural network the location coordinates as well as angles values can not be used as an input, because these values are specific and do not allow to find out similarities with other stored situations in the immune memory.
The situation X for i-th UAV in general may be described by the set of the following subsets:

$$X^{UAV_i} = (< v_1, \delta\theta_{i1}, \delta\phi_{i1}, d_{i1} >, \dots, < v_n, \delta\theta_{in}, \delta\phi_{in}, d_{in} >) \quad (11)$$

Where,
v_j – speed of the j-th UAV, including own speed, obtained directly from the j-th UAV
$\delta\theta_{ij} = \theta^{UAV_i} - \theta^{UAV_j}$ – relative yaw angle between i-th UAV and j-th UAV, converted in a range $0° \leq \delta\theta_{ij} \leq 360°$
$\delta\varphi_{ij} = \varphi^{UAV_i} - \varphi^{UAV_j}$ – relative pitch angle between i-th UAV and j-th UAV, converted in a range of $-70° \leq \delta\varphi_{ij} \leq 70°$
d_{ij} – distance till the possible collision point of the i-th and j-th UAVs.
This set should be arranged by $\delta\theta_{ij}$ and $\delta\varphi_{ij}$ to make the situations look more similar.
In the task, where only speed has been a subject to change [10], but the trajectory was not changing, the distance to crossing point was a constant, while solving the exact situation [10].
In current research, the location of the crossing points is a variable value, that makes the solution more complicated, as the found solution $< \Delta\theta_i, \Delta\varphi_i, \Delta V_i >$ has an influence on the distance to the crossing point, i.e.

$$d'_{ij} = f(\theta^{UAV_i} + \Delta\theta_i, \theta^{UAV_j}, \varphi^{UAV_i} + \Delta\varphi_i, \varphi^{UAV_i}) \quad (12)$$

Thus, the evaluation of the target function requires the additional inputs $< \chi_c^{UAVS}, \psi_c^{UAVS}, \eta_c^{UAVS} >$ obtained directly from the j-th UAV to re-calculate the crossing point and distance to it:

$$P_{max}(\chi_c^{UAVS}, \psi_c^{UAVS}, \eta_c^{UAVS}, \Delta\Theta, \Delta\Phi, \Delta V) \quad (13)$$

In this paper, immune neuro-fuzzy control proposed to be taken as intelligent control method. The proposed INN can be used as one of the unsupervised tools to assess and improve the situation on a point of potential collision and UAV traffic optimization. The structure of INN is presented in Figure 3.
Abbreviations, mentioned in figure 3, are explained in table 1.
The prosed INN for UAV consists of one input layer, one layer with specialized neurons μ and one output layer.
Input data V_i, θ_i, φ_i, d_i, V_0, θ_0, φ_0, d_{01}, …, d_{0n} enters the input layer. From the input layer these data are sent to the AA and μ layer, which is made of μ neurons.

AA checks all the similar situations, stored in the database in the IM and calculates the discrepancies. Situation with a smallest discrepancy is chosen and it's number α is sent to the μ neurons.

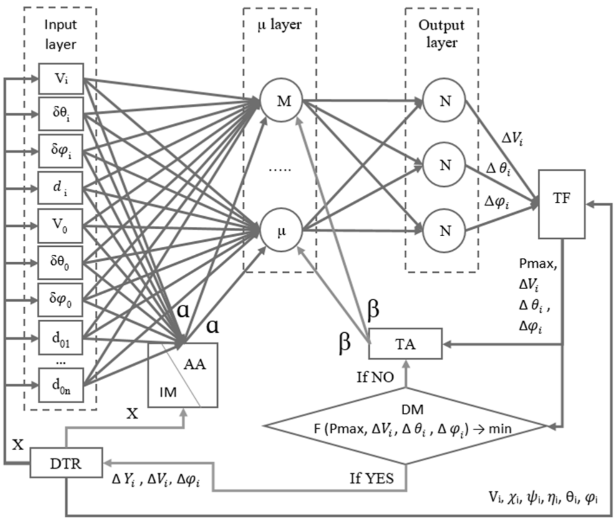

Figure 3: Immune neural network system structure of UAV

Each μ neuron has an immune memory. Here situations' numbers are stored together with weights w_α, which were used while solving the exact problem i.e. processing the similar input data.
After μ neuron received from the AA number of the situation α, the coefficient $\beta = w_\alpha$ is selected from the immune memory of the μ neuron and training is started with found weights application.
The activation function of each traditional neuron generates an output for flight angles change $\Delta\theta_i$, $\Delta\varphi_i$ and speed change ΔV_i for each UAV. P_{max} is also calculated by using TF.
If the collision probability is bigger than specified then weights of μ neurons are changed by the training algorithm β = TA(β) and new values of weights are sent to the μ layer and training is repeated.
If the P_{max} is less or equal to specified, then the information about the number of the situation and weights wα, which were used to solve the optimization task, are saved into the immune memory of μ neurons, number of the situation and input data are saved into the IM database and decision about necessary flight angles change or speed change of the UAV is transmitted to the UAV's DTR device.

5 Algorithm for INFN training model of anti-collision immune neural network and fuzzy logic

The algorithm of INFN consists of the following steps.
Step 0. The FL algorithm assesses the risk of the collision in fuzzy values - low, medium, high, dangerous. This evaluation is the base for initializing of input parameters for the learning algorithm according to the FL membership functions of danger level depending on the distance and time to the crossing point, rules and defuzzification to set the time limits for decision making. Also technical details of embedded device should be taken in account, such as CPU clock rate, RAM size, data storage size etc.
Initialized parameters are:
T_{max} - maximal number of INFN training iterations
$\varepsilon_{\lim}$ – maximal match error, responsible for creation new record in IM or replacing the existing.
P_{safe} – maximal acceptable (safe) value of collision probability
r – coefficient for speed reduction to avoid the collision if solution is not found 0 < r <= 1
Step 1. Receive input data from n UAVs located in the area of visibility. These data are locations and trajectories of UAV.

$$I = (\chi_c^{UAVS}, \psi_c^{UAVS}, \eta_c^{UAVS}, \theta^{UAVS}, \varphi^{UAVS}, V^{UAVS}) \qquad (14)$$

Input data for NN is X= (x_{11}, x_{12}, x_{13}, x_{14}, …, x_{n1}, x_{n2}, x_{n3}, x_{n4}), where $x_{i1} = v_i$, $x_{i2} = \delta\theta_i$, $x_{i3} = \delta\varphi_i$, $x_{i4} = d_i$ received about n objects approaching the crossing point (point of potential collision).
Step 2. The AA (X, S) checks all m situations stored in data base $S = \{s_1, s_2, \dots, s_m\}$ calculates the set of discrepancies $\mathcal{E} = (\varepsilon_1, \dots, \varepsilon_m)$, where

$$\varepsilon_j = \sum_{i=0}^{n} \sum_{k=1}^{2} \left(\frac{x_{ik} - x_{ik}^j}{x_{ik}} \right)^2 \qquad (15)$$

and finds the closest match α, where

$$\varepsilon_\alpha = \min(\mathcal{E}) \qquad (16)$$

All the data in IM is stored in clusters for easier and faster match finding process [10]. For example, if three vehicles are participating in the possible collision situation, there is no need to find the 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.
Step 3. The value α is one of activating input of each μ neuron. When α is received, the coefficient $\beta = w_\alpha$ is selected from the immune memory of the μ neuron. Iteration counter t = 0.
Step 4. Neuron is activated only if α is received or if new β are received from the TA to feedforward the gained input value = $x \cdot \beta$ and increase iteration counter t=t+1.
Step 5. The activation function of each traditional neuron generates an output for own trajectory change Δθ, Δφ or speed change ΔV.
Step 6. The TF calculates P_{max}.
Step 6.1. The TF function defines the trajectory each UAV including own:

$$\frac{\chi - \chi_c}{\cos\varphi \cos\theta} = \frac{\psi - \psi_c}{\cos\varphi \sin\theta} = \frac{\eta - \eta_c}{\sin\varphi} \qquad (17)$$

Step 6.2. Next step is to detect the crossing point (χ_p, ψ_p) in horizontal projection on $\chi\psi$ plane.
Step 6.3. If the crossing (χ_p, ψ_p) is found and is located on the way of motion, then go to Step 6.4. else go to Step 6.6.
Step 6.4. The distance between altitudes of i-th and own UAV is calculated for the (χ_p, ψ_p) point $\Delta\eta = \eta_p^i - \eta_p^{own}$.
Step 6.5. If $\Delta\eta \le D_{safe}$ then it is assumed that potential dangerous point exists and the probability of collision is calculated as described in authors' previous works [15].

Step 6.6. If the crossing (χ_p, ψ_p) is not found, then the trajectories are parallel and D_{safe} should be checked for safe passing
Step 7. If $P_{max} > P_{safe}$ then coefficients of μ neurons are changed by the training algorithm β = TA(β) and if $t<T_{max}$ repeat from Step 4.
Step 8. If $P_{max} <= P_{safe}$ then the activating signal γ is transmitted to both situations database and each μ neurons.
Step 9. If $t>=T_{max}$ then situation can not be solved in the defined time, so the safe solution is necessary. In this paper such solution is speed reduction $\Delta v_i = -r \cdot v$ and END algorithm else go to Step 10.
Step 10. When γ is active and
if $\varepsilon_\alpha > \varepsilon_{lim}$, then each μ neuron saves the existing coefficient as $w_{m+1} = \beta$, the IM saves the situation X as $s_{m+1} = X$ and m=m+1
else if $\varepsilon_\alpha \le \varepsilon_{lim}$, then $w_\alpha = \beta$, the record α in the IM is updated. $s_\alpha = X$,
Step 11. END of the algorithm.

6 Computer models and experiment

The model presenting the workability of developed algorithm is currently implemented in 2D plane, assuming that all UAVs are moving at the same altitude. In the experiment, the team of 8 autonomous UAVs is carrying out the mission in one area and their mission is to patrol the area. Each of UAVs is flying continuously between two target points - from the first point to the second one and back again. The number of UAVs is not changed during the experiments.
UAVs speed, yaw angle, as well as distance till the possible collision point with all the others UAVs are used as input data. Each UAV calculates other vehicles movement direction in relation to his own and finds the point of possible collision. The experiments show the real-time unsupervised machine learning of ANN and INFN to prevent the collisions in the most efficient way making decisions about necessary movement direction (yaw angle) change or speed change.
Following parameters were used for the computer experiment in the proposed system:

- Calculations of the collision probability minimization starts if collision probability is bigger or equal than 0.1;
- Preset speed is 50 m/s, maximal speed 100 m/s;
- Maximal yaw angle change ±90˚;
- Collision probability calculations is done, if another UAV is closer than 150 m to trajectory crossing point;
- Calculations are done with 0.15 seconds pause, for the faster system work;
- Possible number for iterations is 50. If the collision probability was not minimized by jaw angle change in 50 iterations, then speed of the UAV is changed.

The screenshot of the visual part of the simulation is presented in figure 4. It is possible to see the motion of each UAV as well as the speed and trajectory changing in order to avoid the collisions.

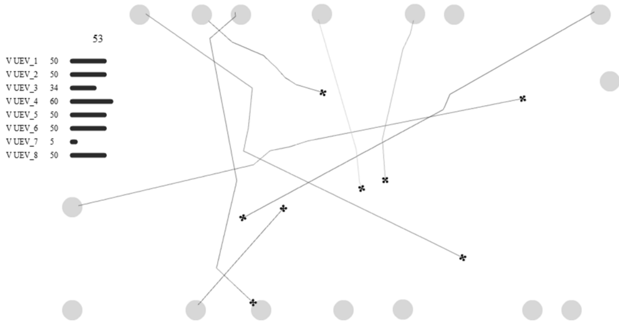

Figure 4: Screenshot from the simulation of real-time UAV machine learning using INFN

Three simulations with the same experimental conditions were made to prove the efficiency of the developed algorithm.
Experiment Nr.1 – no any motion control is used. 135 collisions are detected in 15 minutes of simulation.
Experiment Nr.2 – usual neural network is used for motion control. Immune memory and fuzzy logic are not applicable. Each UAV uses random weights for the target function minimization. 9 collisions are detected in 15 minutes of simulation.
Experiment Nr.3 – proposed Immune Neuro-Fuzzy Network with Immune memory is used for motion control. In 15 minutes of simulation no collisions is detected.
The comparison of the amount of collisions in different experiments is shown in figure 5.

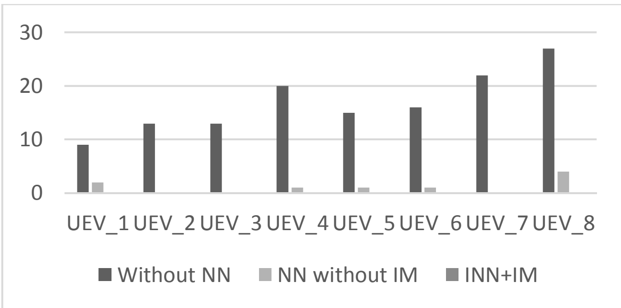

Figure 5: Comparison of collision amount in 3 different experiments

Another parameters to compare were: amount of flights, average distance of the flight and average time of the flight. The comparison of these parameters is proposed in the figure 6.
The biggest number of flights was made during the experiment without any motion control – 656 flights in sum. It happens, because UAVs are moving straight to the target points without changing the trajectory of the flight or speed of the flight. Experiment also proved that amount of flights is bigger while using the motion control with INN than using traditional artificial NN. It happens because while solving the task of collision probability

minimization, UAVs decide to increase the speed and it reaches the target point faster.

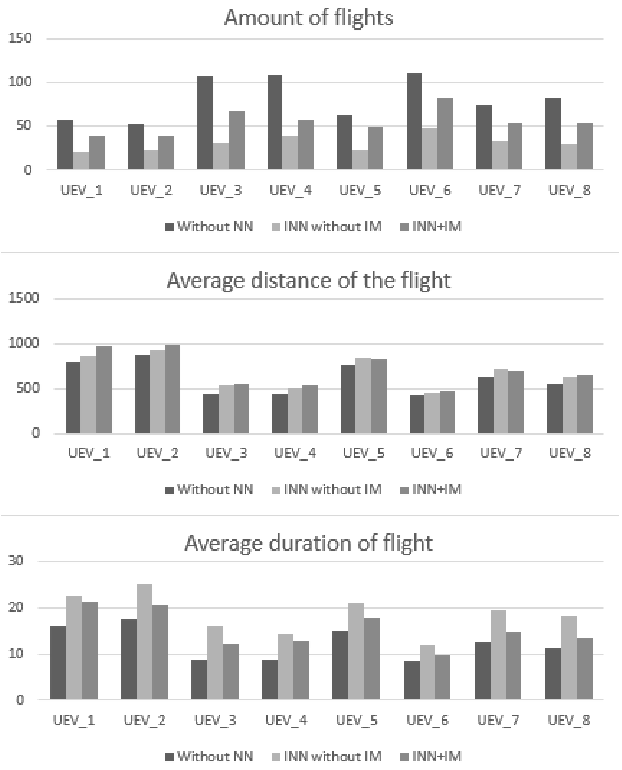

Figure 6: Comparison of flight amount, average flight distance and duration in 3 different experiments

Average distance of the flight is almost the same. It is 615 m per a flight, for the computer experiment without any motion control. The average distance of the flight is 684 m, for the experiment with artificial NN without immune memory. Experiment with INN showed the result of 712 m per flight.

Results of the computer experiments showed that the smallest average duration of the flight was during the experiment without any motion control, because UAVs are just flying straight to the target points without changing the trajectory of the flight. INN with immune memory showed the better results than artificial neural network.

Conclusions

Developed unsupervised machine learning Immune Neuro-Fuzzy Network algorithm can be used to solve a collision prevention tasks for autonomous UAVs team. Developed unsupervised learning method is successfully used for INFN. It helps to minimize time for processing and decision making. Proposed additional layer, immune memory and developed affinity algorithm of INFN allow to increase performance compared to traditional ANN. INFN is able to significantly reduce the collision probability. Proposed system is distributed among the UAVs of the team, so each autonomous device should make an optimal decision for himself.

Computer experiments prove the efficiency of the proposed algorithm. During the experiment with INFN no any collisions were detected, while 135 collisions of UAVs were detected during the computer experiment without any motion control and 9 collisions were detected during the experiment with traditional ANN.

The amount of flights during the experiment with proposed INFN is for 213 less than in the experiment without any motion control but for 194 bigger than with ANN. This proves, that proposed INFN can also increase the amount of safety flights.

Experiment showed that average distance of the flight is approximately the same, but the biggest indicator was detected in the experiment with INN. This parameter is worse than in experiments based on other methods, because while changing trajectory of the flight, also the distance become bigger, but this also helps to reduce the collision probability completely.

Average duration of the flight is the biggest in the computer experiment based on ANN, this proves, that proposed INFN can also reduce the necessary time for the safety flight.

It is necessary to develop prediction algorithms for the location and velocity to continue the calculation, if the data receiving is delayed.

The usage of multi-core system is recommended to increase the performance solving tasks of data transmission, reception and calculation in parallel.

REFERENCES

[1] G. Belloni, M. Feroli, A. Ficola, S. Pagnottelli and P. Valigi. 2007. An Autonomous Aerial Vehicle for Unmanned Security and Surveillance Operations: Design and Test. *In Proceedings of International Workshop on Safety, Security and Rescue Robotics*, Rome, Italy, 4 pages. https://doi.org/10.1109/SSRR.2007.4381277

[2] Md R Haque, M Muhammad, D Swarnaker, M Arifuzzaman. 2014. Autonomous Quadcopter for Product Home Delivery. *In Proceedings of International Conference on Electrical Engineering and Information & Communication Technology*, Dhaka, Bangladesh, 5 pages. https://doi.org/10.1109/ICEEICT.2014.6919154

[3] Zhihua Song, Han Zhang, Ying Wang and Linghui Zhang. 2017. Unmanned Aerial Vehicles Rapid Delivery Routing of the Emergency Rescue in the Complex Mountain Region. *In Proceedings of 13th International Conference on Computational Intelligence and Security (CIS)*, Hong Kong, China, 4 pages. https://doi.org/10.1109/CIS.2017.00081

[4] Yan-Fei Teng, Bin Hu, Zhi-Wei Liu, Jian Huang and Zhi-Hong Guan. 2017. Adaptive Neural Network Control for Quadrotor Unmanned Aerial Vehicles. *In Proceedings of 11th Asian Control Conference (ASCC)*, Gold Coast, QLD, Australia, 5 pages. https://doi.org/10.1109/ASCC.2017.8287305

[5] Fendy Santoso, Matthew A. Garratt and Sreenatha G. Anavatti. 2017. A Self-learning TS-fuzzy System Based on the C-means Clustering Technique for Controlling the Altitude of a Hexacopter Unmanned Aerial Vehicle. *In Proceedings of International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA)* Surabaya, Indonesia, 6 pages. https://doi.org/10.1109/ICAMIMIA.2017.8387555

[6] M. Tim Jones, AI Application Programming, Charles River Media, 2003.

[7] T.J.Ross, "Fuzzy Logic with Engineering Applications", 2nd Edition, University of New Mexico, USA, 2004.

[8] Haohan Chen and Haibin Duan. 2016. Multiple Unmanned Aerial Vehicle Autonomous Formation via Wolf Packs Mechanism. *In Proceedings of IEEE International Conference on Aircraft Utility Systems (AUS)*, Beijing, China, 5 pages. https://doi.org/10.1109/AUS.2016.7748123

[9] Anna Beinarovica, Mikhail Gorobetz and Anatoly Levchenkov. 2018. Control Algorithm of Multiple Unmanned Electrical Aerial Vehicles for Their Collision Prevention. *In Proceedings of 12th International Conference on Intelligent Technologies in Logistics and Mechatronics Systems, ITELMS 2018 Panevėžys, Lithuania*, 37-43 pp.

[10] Anna Beinarovica, Mikhail Gorobetz and Anatoly Levchenkov. 2018. Self-Organized Learning Algorithm for Immune Neuro-Fuzzy Anti-collision System of Autonomous Unmanned Aerial Vehicles' Team. *In Proceedings of 22nd international scientific conference TRANSPORT MEANS 2018*, Trakai, Lithuania, 8 pages.

[11] Robert Fuller. 2001. Neuro-Fuzzy Methods for Modeling & Fault Diagnosis, Lisbon, Portugal, 77 pages.

[12] Girts Stana, Viesturs Brazis and Marcis Prieditis, 2018. Simulation-Based Comparisons of Mobile and Stationary Energy Storage Systems Applied for Electric Transport. *In Proceedings of 20th European Conference on Power Electronics and Applications (EPE'18 ECCE Europe)*, Latvia, Riga, 17-21 September, 2018. IEEE, 2018, 9 pages, ISBN 978-1-5386-4145-3. e-ISBN 978-9-0758-1528-3.

[13] Peteris Apse-Apsitis, Kristaps Vitols, Edgars Grinfogels, Armands Senfelds and Ansis Avotiņs. 2018. Electricity Meter Sensitivity and Precision Measurements and Research on Influencing Factors for the Meter Measurements. IEEE Electromagnetic Compatibility Magazine, 2018, in Press, 1.-5.lpp. ISSN 2162-2264.

[14] Kristaps Vitols, Pavels Suskis and Oskars Krievs. 2018. Practical Approach of Testing Various Thermal Interface Materials for Power Electronics. *In Proceedings of 20th European Conference on Power Electronics and Applications (EPE'18 ECCE Europe)*, Latvia, Riga, 17-21 September, 2018. IEEE, 2018, 9 pages, ISBN 978-1-5386-4145-3. e-ISBN 978-9-0758-1528-3.

[15] Leonids Ribickis, Anatoly Levchenkov and Mikhail Gorobetz. 2018. Neuro-Immune Algorithm for Embedded Real-time Control System in Transport Safety Tasks. *In journal book Frontiers of Artificial Intelligence and Applications (FAIA)*, IOS Press, Vol. 310, 251-265 pp. https://doi.org/10.3233/978-1-61499-929-4-251.