Design of the intelligent adaptive hybrid control system for autonomous mobile robot on the basis of neuro-fuzzy networks

V. Finaev, I. Kobersy, D. Beloglazov, I. Shapovalov, E. Kosenko, and V. Soloviev

**Abstract**— The paper is devoted to design of the adaptive hybrid system for motion control of an autonomous mobile robot. We designed the control system on the basis of artificial neuro-fuzzy networks. The aim of control consists in safe obstacle avoidance during movement and regulation of velocity without operator participation. The structure of adaptive hybrid control system consisting of three modules is given. The stages of each module design are defined and the structures of neuro-fuzzy networks for the modules implementation are chosen. Here we present the operation algorithms of movement direction control module, velocity control module, and obstacle identification module.

**Keywords**— Adaptation, hybrid system, neuro-fuzzy networks, uncertainty, vehicles.

I. INTRODUCTION

The problem of the autonomous mobile robot (vehicle) motion along a desired path with the obstacle avoidance is one of the most important tasks in robotics [1]-[5]. It can be solved by different methods, for example, by methods of the classical control theory, position-trajectory control [1], [6], etc. There is a strong interest in alternative solutions, the use of artificial intelligence methodology despite the positive results of above mentioned approaches.

We propose to use the artificial neuro-fuzzy networks (NFNs) of modified architecture [7], [9] to solve the problem of vehicle control.

Fuzzy control is one of the most promising development lines of advanced control theory. The use of fuzzy logic allows providing the opportunity of adaptive hybrid control system (AHCS) operation with incomplete, inaccurate data about the state of vehicle and environment [10] – [15].

The task of autonomous vehicle control system development can be represented in the form of a specific sequence of subtasks [13], [15]: the creation of idea about the environment state by transmitting and receiving sensor signals; the classification of situations based on the sensor data about the environment state; making decisions to change velocity, direction of the vehicle movement.

II. THE ALGORITHM OF MOVEMENT DIRECTION CONTROL MODULE OPERATION

The structure of the AHCS is given in Fig.1. The module of motion direction control was implemented on the basis of neuro-fuzzy network 1 (NFN1). NFN2 is used to control the vehicle velocity. The NFN3 carries out classification of obstacles on the way of vehicle based on the distance between obstacles and vehicle.

![Fig. 1 Structure of hybrid vehicle control system](image)

Distinctive feature of these neuro-fuzzy networks (NFN1, NFN2, NFN3) is the use of artificial neural networks as a defuzzification subsystem in their structure.

The overall structure of adaptive neuro-fuzzy control system is similar in structure to a simple feed-forward neural network (FFNN) [7] (individual neurons are the neuro-fuzzy networks).

The design of AHCS modules using fuzzy logic consists of the following stages: selection of module architecture; development of module structure; development of module training algorithm; development of AHCS training algorithms. The process of parameters adaptation in individual modules and in the control system as a whole depends on the algorithms used, as they determine the quality of training.

Data from the navigation system, the modules of velocity...
control, and obstacle identification is received at the input of direction control module (NFN). The NFN with three inputs and one output allows control only “motion direction” parameter. The algorithm of NFN architecture design consists of the following steps.

**Step 1.** In order to solve the vehicle motion direction control task it is necessary to define input and output parameters of the module, the term sets of these parameters. We used the triangular form of membership functions as a basic form of fuzzy variable term sets, because this versatile form is the most commonly used by developers [16] - [18].

Let’s define the term sets of input and output linguistic variables.

The first linguistic variable $T_{11}$ defines the obstacle position in space relative to the vehicle and has the term set $T_{11} = \{ t_1^1, t_2^1, t_3^1, \ldots, t_{26}^1 \} = \{ \text{front area: in a center, far from center} – \text{close to right area, very far from center} – \text{very close to center}; \text{right area: in a center – very close to front area, far from center} – \text{close to center – close to back area, very far from center} – \text{very close to center} – \text{very close to center}; \text{back area: in a center, far from center – very far from center} – \text{very close to back area, close to center, very close to left area, close to center, very close to center – very close to back area, close to center, very close to center – very close to back area, close to center, very close to center – very close to right area, very far from center – very close to right area, far from center – very close to left area, very far from center – very close to left area, close to center, very close to center – very close to center, close to center, very close to center} \}$. The term set $T_{11} = \{ t_1^1, t_2^1, t_3^1, \ldots, t_{26}^1 \}$ includes twenty six fuzzy variables defining the linguistic variable “obstacle position”. The structure of fuzzy variables composing the linguistic variable $T_{11}$ is shown in Fig. 2.

![Fig. 2. The separation of external space](image)

Measured vehicle velocity value $T_{12}$ comes to the second input of NFN and fuzzy variables from the term set $T_{12} = \{ t_1^2, t_2^2, t_3^2, \ldots, t_{5}^2 \} = \{ \text{high velocity; very high velocity; low velocity; very low velocity; stop} \}$ define this value.

The third linguistic variable $T_{13}$ describes the distance from the vehicle to the destination point. This linguistic variable has the term set $T_{13} = \{ t_1^3, t_2^3, t_3^3, \ldots, t_{7}^3 \} = \{ \text{very far from destination; far from destination; middle of route; not very far from destination; not very close to destination; very close to destination; destination} \}$. The term set $T_{13}$ includes seven fuzzy variables.

The process of setting membership functions parameters begins after defining the linguistic variables and their term sets.

**Step 2.** At this step the number of fuzzy neurons in the first layer of NFN is calculated. The number $L_1$ of fuzzy neurons in the first layer of NFN is equal to sum of cardinal numbers of input linguistic variables term set: $L_1=T_{11}+T_{12}+T_{13}=38$.

The membership function of fuzzy neurons is described by expression:

$$
\mu_{A_k} = \begin{cases}
0, & x \in [a_k^i, b_k^i] \\
\frac{x - a_k^i}{b_k^i - a_k^i}, & x \in [a_k^i, b_k^i] \\
\frac{x - c_k^i}{b_k^i - c_k^i}, & x \in [b_k^i, c_k^i] \\
0, & x \in [c_k^i, +\infty]
\end{cases}
$$

(1)

The following parameters define the shape of such membership function: $a_k^i$, $b_k^i$, and $c_k^i$, where $b_k^i$ is the center; $a_k^i$ and $c_k^i$ are the borders of membership function.

**Step 3.** The number of fuzzy neurons in the second layer of NFN is calculated. The output block defining the degree of fulfillment of the fuzzy rules conditions is implemented in the second layer of NFN.

$$
\tau_k = \min_{i=1,\ldots,n}\{\mu_{A_k}(x_i)\}
$$

(2)

The number of elements in the second layer is equal to the number of fuzzy rules: $L_2=N_1\times N_2\times N_3$. In our case: $L_2=5\times9\times24=1080$.

**Step 4.** The quantity of fuzzy sets defines the number of elements in the third layer by formula:

$$
L_3 = \frac{L_2}{M}
$$

(3)

As the number of rules in the second layer is $L_2$, the number of fuzzy neurons is $L_3=360$.

All the layers have weights of connections. These weights are usually equal to 0 (no connection) or 1. Such approach is convenient for connection of the third and the second layer according to a principle “each with each”. If the third layer contains $r$ elements, we can write

$$
y_r = \max_{i=1,\ldots,n}\{\tau_k, w_{kr}\},
$$

(4)

where $r=1,\ldots,m$ is the number of element in the third layer, $1,\ldots,N$ is the rule number, $w_{kr}$ are the weights of connections between element $k$ in the second layer and element $r$ in the
third layer.

Fig. 3 shows the structure of the process of fuzzy rules conclusions goodness calculation based on the given conditions.

In the NFN_1 each element of the second layer is connected to only one element of the third layer. The same condition cannot have several conclusions in the base of fuzzy rules, but the same conclusion can be drawn based on different conditions. The second, the third, and the fourth steps are the solution of fuzzification and fuzzy rule base development tasks.

Step 5. The fuzzification task is complex. We face the problem of robot rotational displacement relative to a current motion trajectory in the task of vehicle motion direction control.

We use the rotational displacement with the range of value (-a, a) as an output variable. If we have the membership functions \( \mu \{ -a, 0, 0, 0, 0, a \} \) as a result of output block operation, we would expect the numerical value of control signal close to \( a \) or to \(-a\). The majority of fuzzification methods cannot handle this condition and give the angle value equal to \(0\).

To solve this problem, we will use neural networks, which are capable to implement different mathematical dependencies.

The input layer of neural network from the fuzzification block is connected to the last, the third layer of previously mentioned module. Let's denote weight vectors of these \( i\)-th neurons connection as \( w^{(1)}_i \). The number of elements in this layer is defined by the number of previous layer fuzzy rules. Values from the first layer of defuzzification neural network enter to inputs of the second hidden layer of defuzzification neural network.

The calculation of the number of neurons in the hidden layer is a very important part of a whole neural network architecture design. For this calculation the next rules can be used [20]: the number of hidden neurons should be in a range of the input and output layer size; the number of hidden neurons should be \(2/3\) of the input and output layer size; the number of hidden neurons should be twice less than the output layer size. We chose the number of hidden neurons equal to 180.

We denote the connection between input and hidden layers as \( w^{(2)}_i \). The neurons number of output (last) layer is equal to 1. This layer produces control signal that is defuzzificated fuzzy conclusions.

The NFN_1 module training consists in tuning of parameters of the neural network responsible for control signal defuzzification. The complete structure of NFN_1 is shown in Fig. 4.

The NFN (from the first to the third layer) purpose consists in definition the degree of input data compliance with each inference rule. Different learning algorithms are used in order to adapt this network to given task. These algorithms define the difference between the given (reference) value and the real value of output signal.

III. THE ALGORITHMS OF VELOCITY CONTROL MODULE OPERATION.

The second stage of the vehicle control system design is a development of the velocity control module. The fuzzy control system as a standard neural network is described by the membership function of fuzzy set \( B_k \)

\[
\mu_{B_k}(\bar{x}_k) = \prod_{j=1}^{n} \exp \left[ -\left( \frac{x_j - \bar{x}_k}{\sigma_j} \right)^2 \right].
\]  

The membership function describing defuzzification operation has a form

\[
\bar{y} = \frac{\sum_{k=1}^{N} \bar{y}^k \exp \left( h^k \left( \sum_{i=1}^{n} \bar{x}_i \bar{x}_i^k - I \right) \right)}{\sum_{i=1}^{N} \exp \left( h^k \left( \sum_{i=1}^{N} \bar{x}_i \bar{x}_i^k - I \right) \right)}.
\]

The structure shown on Fig. 5 is a modification of neuro-fuzzy control system with an artificial neural network as a defuzzification block.
Advances in Engineering Mechanics and Materials

The inputs of NFN2 are the following: $X_2$ is the signal from sensors of distance to obstacle; $Y_1$ is the output of NFN1 controlling the motion direction; $Y_2$ is the output of NFN2 classifying obstacles based on their distance to the vehicle. The output of NFN2 module is the velocity control signal $Y_2$.

The first layer is formed by classical neurons with a weighted sum of inputs and an exponential activation function. Each neuron of the first layer corresponds to one fuzzy rule. The distance from obstacle to vehicle, the motion direction, and obstacle classifier are the linguistic variables $T_{2i}$ ($i=1, 2, 3$) of module.

The first input linguistic variable «distance from vehicle to obstacle» $T_{21}$ has the term set $T_{21} = \{l^1_1, l^1_2, l^1_3, \ldots, l^1_7\} = \{\text{very large distance; large distance; medium distance; not very large distance; not very small distance; very small distance; almost collision}\}$.

The second linguistic variable «motion direction» $T_{22}$ has the term set $T_{22} = \{l^2_1, l^2_2, l^2_3, l^2_4, l^2_5\} = \{\text{to the left; sharply to the left; keep direct course; to the right; sharply to the right}\}$.

The third linguistic variable «class of obstacle» $T_{23}$ has the term set $T_{23} = \{l^3_1, l^3_2, l^3_3, l^3_4, l^3_5\} = \{\text{high degree of need to change the direction of motion; very high degree of need to change the direction of motion; low degree of need to change the direction of motion; change the direction of motion; stop}\}$.

The size of fuzzy rule base of the velocity control module is defined by expression $L_2 = T_{21} \times T_{22} \times T_{23} = 245$ on the basis of cardinality number of the linguistic variable term set.

The second and the third layers of neural network perform the defuzzification operation. These layers consist of neurons with a linear activation function.

Weights of the first neuron from the second layer are interpreted as centers of fuzzy set membership functions and they are modified by the training process. Weights of the second neuron are the constants equal to 1. The last layer of the NFN2 module contains one neuron that outputs the final value of vehicle velocity correction.

IV. THE ALGORITHMS OF OPERATION OF OBSTACLE CLASSIFICATION MODULE

The third stage of the vehicle control system design consists in the obstacle classification module development on the basis of NFN. We suppose the NFN1 having the almost same structure as the NFN2. The differences are the sets of input and output signals and a solved task.

The inputs of NFN3 are the following parameters: $X_3$ is the signal from navigation systems determining distance between obstacles and the vehicle; $Y_1$ is the output of NFN1 controlling the motion direction; $Y_2$ is the output of NFN2 controlling the vehicle velocity.

The output $Y_3$ of NFN3 is the obstacle class defined on the basis of distance to the vehicle, etc.

Let’s verbally define the term set of NFN3 input variables. The first linguistic variable $T_{31}$ «distance from vehicle to obstacle» has the term set $T_{31} = \{l^1_1, l^1_2, l^1_3, \ldots, l^1_7\} = \{\text{in central area; far from center - near to vehicle; very far from center - very close to vehicle; close to center - far from vehicle; very close to center - very far from vehicle; close to area borders; very close to area borders}\}$.

The second linguistic variable $T_{32}$ «motion direction» has the term set $T_{32} = \{l^2_1, l^2_2, l^2_3, l^2_4, l^2_5\} = \{\text{to the left; sharply to the left; smoothly to the left; keep straight course; to the right; sharply to the right; smoothly to the right}\}$.

The third linguistic variable $T_{33}$ «vehicle velocity» has the term set $T_{33} = \{l^3_1, l^3_2, l^3_3, l^3_4, l^3_5\} = \{\text{very high velocity; high velocity; middle velocity; low velocity; very low velocity}\}$.

The first layer of fuzzy module NFN3 consists of $L_3 = T_{31} \times T_{32} \times T_{33} = 245$ fuzzy rules. The second module consists of $L_2 = 2$ neurons. Parameters of the NFN3 are calculated as in NFN2 module. The training algorithm of vehicle state control modules consists in a reduction of previous weight by the value of error derivative. This process continues while the output error of the system is greater than a priori given minimal value.

V. STRUCTURE OF THE INTELLIGENT AHCS

The developed modules NFN1, NFN2, NFN3 are the basis of the control system block-diagram shown in Fig 6.

Let’s consider the common algorithms of system operation. The data formed by sensors about the environment is transmitted to the common control system input. Further the data goes to inputs of corresponding control modules (NFM1, NFN2, NFN3).
Fig. 6. Structure of the intelligent vehicle AHCS.

Operation of the first module NFN₁ begins from the fuzzification of vehicle velocity values and parameters of obstacles. The forming of output signal on the basis of fuzzy rules and fuzzy inference takes place on the next step. Then defuzzification is performed using the neural network consisting of three layers. Control signals correcting the motion direction are formed at the output of neural network. Obtained value comes to input of the second control module. Information about obstacles and the vehicle position also comes to the second input of NFN₂.

Received data is fuzzificated and processed on the base of fuzzy rules. The fuzzy output of module is defuzzificated, and the velocity control signals are formed at the output of NFN₃ module.

The data received from the NFN₁ and the NFN₂ is transmitted to inputs of the third module. Information about the obstacle location comes to the third input of NFN₃. The module output and defuzzification are calculated after the fuzzification procedure on the base of fuzzy rules. The information received at the output of NFN₃ is transmitted to inputs of the NFN₁ and the NFN₂.

The developed system operates in the closed cycle. Operation would continue while sensor information about the parameters of environment, vehicle, and obstacles is received or until the desired location is achieved.

VI. CONCLUSIONS

To solve the autonomous mobile robot control problem in the data incompleteness conditions, we propose the structure of AHCS. This structure is similar to FFNNs by the operation principles. We implemented operational modules of the AHCS on the basis of neuro-fuzzy networks. We developed the algorithms of modules operation for the control of motion direction, velocity, and the obstacle classification. Also we designed the structures of modules in the form of NFNs. The input parameters as linguistic and fuzzy variables were defined.

REFERENCES