

MOBILE ROBOT AUTOMATIC NAVIGATION CONTROL ALGORITHM BASED ON FUZZY NEURAL NETWORK IN INDUSTRIAL INTERNET OF THINGS ENVIRONMENT

P. Pei¹, Yu. N. Petrenko¹

¹ *Belarusian National Technical University, Minsk, The Republic of Belarus*
220013, The Republic of Belarus, Minsk, prospect Nezavisimosti, 65

Abstract

Mobile robot is an important developing direction in the field of robotics, it is widely used in Industrial Internet of Things (IIoT) environment, agriculture, military, transportation, services with the coming of 5G wireless communication technology. Automatic navigation control technology is the core in these research areas, which is also the key technology for mobile robot to achieve intelligentization and autonomation.

The article discusses and researches the neural network technology and its application in mobile robot navigation control. For the characteristics and research of mobile robot navigation problem, it finds the way to improve the mobile robot intelligentization, level of the self-organization, self-learning and adaptive capability. The combination of neural network with other intelligent algorithms solves autonomous navigation problem of the mobile robot in the complex uncertain environments and unknown variable environments. The mobile robot navigation control problem using fuzzy neural network can achieve a more effective real-time navigation control performance through amending the network weights by self-study according to the navigation priori knowledge of human experts.

Keywords: automatic navigation control, 5G, neural network, fuzzy neural network.

DOI: 10.32517/0234-0453-2020-35-1-59-67

For citation:

Pei P., Petrenko Yu. N. Mobile robot automatic navigation control algorithm based on fuzzy neural network in industrial Internet of things environment. *Informatika i obrazovanie — Informatics and Education*, 2020, no. 1, p. 59–67.

Received: November 11, 2019.

Accepted: January 21, 2020.

About the authors

Pei Ping, postgraduate student at the Department “Information Systems and Technologies”, International Institute of Distant Education, Belarusian National Technical University, Minsk, The Republic of Belarus; peiping123456.love@gmail.com; ORCID: 0000-0001-6751-7598

Yury N. Petrenko, IEEE Life Member, Candidate of Sciences (Engineering), Docent, Associate Professor at the Department “Information Systems and Technologies”, International Institute of Distant Education, Belarusian National Technical University, Minsk, The Republic of Belarus; ypetrenko@bntu.by

1. Introduction

Creating the ability of artificial neural network to simulate part of human image thinking is an approach to simulate artificial intelligence. In particular, the artificial neural networks combine to 5G wireless network can be used to solve some of the problems encountered in artificial intelligence research [1]. The application of artificial neural network theory has penetrated into many fields, with gratifying achievement in computer vision, pattern recognition, intelligent control, nonlinear optimization, and adaptive filtering phase information processing and other aspects [2]. Meanwhile, the neural network has also been widely used in the fields of robot kinematics, dynamics and control, etc., due to its adaptive learning function and non-linear mapping capability [3]. In the mobile robot navigation, it was mainly used for the representation of the environment model, local planning, global planning, sensor information fusion, robot control system and so on [4]. In this chapter, the working principle and learning method of neural network was systematically described, several typical neural network models were introduced, and

a robot neural control based on fuzzy neural network was studied. In the neural network framework, the qualitative knowledge was introduced, namely, the neural network was constructed by the fuzzy rule by adding fuzzy layer to the input layer and the output layer of the conventional neural network. The adaptive learning mechanism of the neural network was retained while the network weight was given with a clear physical meaning. Finally, the fuzzy neural network algorithm was applied to the mobile robot navigation control simulation experiment. Based on the priori knowledge summarized by human navigation experts, the network weight was corrected by self-learning to achieve more effective real-time navigation control.

2. Overview of artificial neural network

The artificial neural network is a complex network system that is comprised of numerous simple processing units that reflect nonlinear essential features (such as neurons, processing elements, electronic components, optoelectronic components, etc.). It is proposed based on modern neuroscience research and a reflection of the basic characteristics of human brain function.

However, it is a certain simplification, abstraction and simulation of the human brain, not a true portrayal [4]. The neural network is a very large-scale continuous-time dynamic system with high nonlinearity, which is mainly characterized by distributed storage of information and parallel collaborative processing of information. Although the structure of a single neuron is extremely simple with limited function, the network system composed of a large number of neurons has a rich and colorful behavior. The parallel processing capability of neural network is achieved by a distributed structure in which processes are represented by different numbers of neurons and different interconnection modes and patterns. The operation of the neural network is the transfer process from the input to the output value in which the storage and calculation of the information is completed simultaneously, thus perfectly combining the information access with the calculation. The transfer process of values is similar to that of the current in the resistive network, and the individual neurons in the neural network are working in a parallel way. This intrinsic parallelism differs from the parallelism of the parallel computers in terms of concepts and implementation methods. Compared with digital computers, neural network system has the ability to collectively compute and adaptively learn. In addition, it is excellent at association, synthesis and promotion with strong fault tolerance and robustness.

3. Neural network model

The basic structure of the nervous system is a neuron (neural cell), which is the basic unit for processing information between various parts of the human body. Neurobiologists have found that there were typically 1.4×10^{11} neurons in a human brain. Each neuron consists of a cell body, an axon that connects other neurons, and dendrites which are short and protruding outward. The function of axons is to transmit the output signal (excitement) of this neuron to other neurons. Numerous nerve endings at its ends allow excitement to be transmitted to multiple neurons simultaneously. The function of dendrites is to accept excitement from other neurons. The neuron cell body simply processes all the signals it receives (such as weighted summation which takes into account all input signals and reflects the degree of emphasis on each signal in different weights) before being exported by axons. The part of the neuron dendrites that is interconnected with the nerve endings of another neuron is called a synapse [4]. The brain is able to handle extremely complex analysis and reasoning work since there is a huge number of neurons in the brain on the one hand and the neurons have the ability to nonlinearly process input signal on the other hand. Therefore, a neuron mathematical model that is closer to engineering as Figure 1 can be further established, which is a multi-input single-output nonlinear unit, with weights representing the strength of the connection between neurons, and $f(x)$ as a nonlinear transfer function [4].

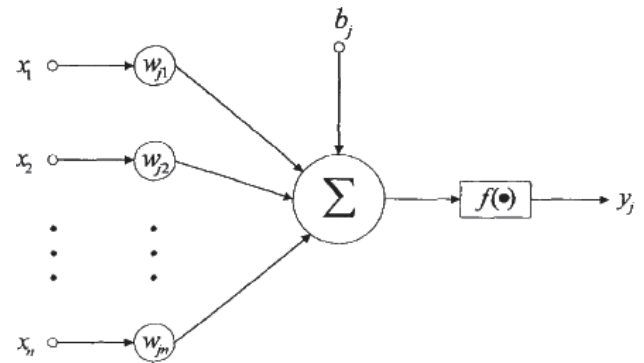


Fig. 1. Mathematical model of neurons

The mathematical expression of the model is as follows:

$$y_j = f\left(\sum_{i=1}^n w_{ji}x_i - b_j\right), \quad (1)$$

where:

x_i is the input signal at the input end;

w_{ji} — connecting weight coefficient;

b_j — the offset of the neuron;

n — the number of input signals;

y_j — the output of the j^{th} neuron;

$f(\bullet)$ — excitation function (or output transform function).

Various complex neural network systems are composed of a large number of interconnected neurons. The common neural network model is the PDP (Parallel Distributed Processing) model proposed by Rumelhart, which covers the following eight aspects [4]. Various complex neural network systems are composed of a large number of interconnected neurons. The common neural network model is the PDP (Parallel Distributed Processing) model proposed by Rumelhart, which covers the following eight aspects [5].

1. The neural network is composed of a group of neurons, which determines that the neural network has parallel processing capability. The MP model is used by each neuron in the neural network uses, but can be divided into input unit that accepts input from the outside world, output unit that sends a signal to the outside world, and hidden unit that is an internal unit of the system without any connection with the outside world. The definition of transfer functions and output functions for different neurons may vary as required.

2. The activation state of the neural network is a state vector composed of the activation states of all neurons, indicating the activation pattern of the system, and characterizing the object represented by the system at time t . The activation value of the unit can be taken as continuous value, such as a value between $[0, 1]$, or a discrete value, such as $\{0, 1\}$.

3. The output of Unit i is calculated with the following equation: $o(t) = f(a(t))$, where $f(\bullet)$ can be identity function, threshold function and so on. The output vector of the neural network, $O(t)$ is constituted by entire unit outputs.

4. The connection pattern means the interconnection between units, which is represented by a connection

weight matrix W . w_{ij} represents the strength and nature of the connection from unit i to unit j . If unit i excites unit j , then $w_{ij} > 0$; if unit i suppresses unit j , then $w_{ij} < 0$. The absolute value of w_{ij} represents the connection strength. The connection pattern constitutes the knowledge of the system, which is distributed in the weight matrix in the form of encoding. A weight matrix is an encoding method that reflects certain knowledge. Thus, the connection pattern embodies the distributed capabilities of the neural network.

5. The passing rule means the combination of the output vector $O(t)$ with the connection matrix, which is the rule that various types of input enter the unit to generate a net input. The net input for unit i is calculated with the following equation: $\text{net}_i = \sum_j w_{ij} o_j(t)$.

6. The activation rule is the rule for updating of the system pattern, which is directly reflected as the activation rule of a single unit, namely $a(t) = F(a(t-1), \text{net}(t) - b)$, represented as $a(t) = F(\text{net}(t))$ under simple pattern.

7. Learning ability is an important feature of the neural network. The learning is the process of changing the system knowledge in order for it to approximate the knowledge to be learned. Since the system knowledge is represented by the connection weight matrix, it is the rule for changing the weight. The learning rules in the PDP model are derived from the Hebb rules to some extent.

8. The working environment is the overall input pattern of the vector set to the external input. Some PDP models require orthogonal sets or linear independent sets, while some PDP models can be used to learn arbitrary input patterns [5].

4. Learning rules in neural network

An appropriate learning method is required for a neural network whose topology has been determined, in order to enable the neural network to exhibit some intelligent characteristics. It can be said that the learning method is the core of researches on artificial neural network [5].

Different functional areas in the brain have their own learning rules. These complete and ingenious learning rules are obtained by the brain in the process of the evolution. For the artificial neural network, the learning method is ultimately the method for adjustment of the network connection weight. The learning algorithms of artificial neural networks can be divided into supervised learning and unsupervised learning. Supervised learning is performed by an external teacher signal, that is, to require a pattern pair that gives both input and correct expected output. When there is any difference between calculated result and expected output, the network will finally generate the correct results by repeating the training through the automatic adjustment mechanism. Unsupervised learning rules do not involve external teacher signals, and their learning behavior is detection rule adaptive to the input space. Since dynamic input signals are

provided to the system in the learning process, and each unit may compete in some way, the winner neurons themselves or their neighborhood are enhanced while other neurons are further suppressed, thus dividing the signal space into multiple effective areas.

The three main rules that are commonly used include [6]:

1) Hebb learning rule for unsupervised learning:

Hebb learning is a kind of related learning based on the following idea: If two neurons are excited simultaneously, the connection strength between them is proportional to the product of their excitation. Let o_i be the activation value (output) of unit i , o_j be the activation value of unit j , and w_{ij} be the connection weight coefficient of unit j to unit i , then the Hebb learning rule can be expressed by the following equation:

$$\Delta w_{ij} = \alpha o_i(t) o_j(t), \quad (2)$$

where α is learning rate.

2) Supervised learning rules or Widrow-Hoff learning rules:

The supervised learning rules are created by introducing the teacher signal into the Hebb learning rule and substituting o_i in the equation (3) with the difference between the expected target output d_i and the actual output o_i of the network.

$$\Delta w_{ij} = \alpha [d_i(t) - o_i(t)] o_j(t). \quad (3)$$

The above equation indicates that the change in the connection strength between the two neurons is proportional to the difference between the teacher signal d_i and the actual output o_i of the network.

3) Supervised Hebb learning rule is created by combining unsupervised Hebb learning rules with supervised learning rules:

$$\Delta w_{ij} = \alpha [d_i(t) - o_i(t)] o_i(t) o_j(t). \quad (4)$$

This learning rule enables neurons to respond to an unknown outside world through an association search, namely, enhancing or weakening the corresponding output under the guidance of the teacher's signal $d_i(t) - o_i(t)$ through relevant learning and self-organization of the environmental information [6].

5. Research on mobile robot navigation control based on fuzzy neural network

5.1. Principle for fuzzy neural network

The fuzzy neural network is a local approximation network that combines the knowledge structure of fuzzy logic reasoning with the self-learning ability of neural network, which is the product of the combination of fuzzy theory and neural network. With the advantages of both the neural network and fuzzy control, it integrates learning, association, recognition, adaption and fuzzy information processing. The combination of fuzzy logic and neural network is one of the key technologies for achieving machine intelligence. By organically combining fuzzy technology with neural

network technology, it has the ability to effectively exert their respective advantages and make up for the shortcomings [7]; the advantage of fuzzy technology lies in the logic reasoning ability which is easy to carry out high-order information processing and introduce fuzzy technology into the neural network, thereby broadening the scope and capabilities of the neural network to process information. In such a way, it can process not only accurate information but also fuzzy or inaccurate information, thus achieving accurate association and mapping as well as inaccurate association and mapping, especially fuzzy association and mapping. Due to its strong advantages in learning and automatic recognition of patterns, the neural network used for fuzzy information processing makes it possible to automatically extract fuzzy rules and automatically generate fuzzy membership functions, thereby allowing the fuzzy system to become an adaptive fuzzy system.

Currently, the fuzzy neural network model is mostly a multi-layer forward network structure, which is attributed to the unidirectional nature of fuzzy reasoning. Different types of fuzzy neural networks have been established based on different neurons and different fuzzy components. Fuzzy neural networks vary mainly depending on membership function, fuzzy weighting operator, fuzzy excitation function and input and output form, as well as structure and parameter setting and adjustment methods [8]. In this thesis, the qualitative knowledge was introduced under the neural network framework, namely, adding fuzzy layer to the input layer and output layer of the conventional neural network. The neural network was established with fuzzy rules in order to retain the learning mechanism of the neural network while providing definite physical meaning for the network weight. By combining the advantages of the fuzzy method with those of the neural network, this method was more targeted than the common neural network. The fuzzy neural network control system is designed basically the same as the fuzzy controller. In this section, the fuzzy neural network was controlled through correction of the self-learning network weight, by establishing the model for fuzzy variable set, the membership function and the fuzzy reasoning control was established with the multi-layer feed-forward BP network [9].

5.2. Fuzzy neural network structure

Figure 2 shows the structure of a fuzzy neural network control model, which is a five-layer network. First, the input and output of the fuzzy neural network neurons is formally described. Each neuron input in the network is defined as $f^{(q)}(x_1^{(q-1)}, \dots, x_n^{(q-1)}; w_{ji}^q, \dots, w_{ji}^q)$ and the output of the node as $x_j^q = g^{(q)}(f^{(q)})$, where the superscript q represents the number of layers. Layer 1: This is the input layer consisting of n input values, and its node is the input node, representing the input linguistic variable [10]. It plays a role in transferring the input value to the next layer, that is $f_i^{(1)} = x_i$, with a connection weight of 1. Layer 2: This is the fuzzification layer which performs a fuzzification of the input values quantitatively, that is, to define some

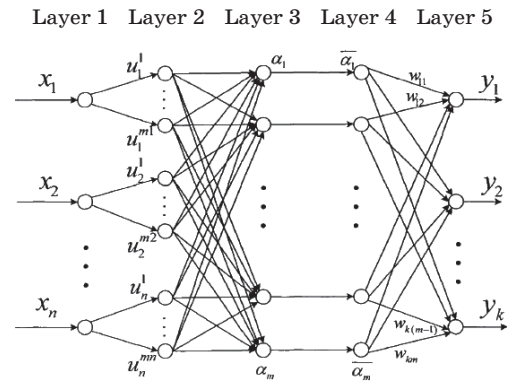


Fig. 2. Fuzzy neural network structure

fuzzy subsets for the input values and then convert these input values into degree of membership on the corresponding fuzzy subsets through the membership function on the corresponding fuzzy subsets. Each node represents a linguistic variable value, such as LB, RS, etc. The radial basis function was used as the excitation function for neurons mentioned here, which generally had different shapes like Gaussian, bell, trapezoidal or triangular. The radial basis function in this thesis was Gaussian function, namely:

$$F = \frac{(x_i^{(2)} - c_{ij})^2}{\sigma_{ij}^2}, \text{ and } \alpha = e^{-F}, \quad (5)$$

where c_{ij} and σ_{ij} in the equation (5) represents the center and width of the Gaussian function in the j^{th} term of the i^{th} input linguistic variable, respectively. When there is an accurate signal input at the input layer, the corresponding membership can be generated in Layer 2 under the excitation of the radial basis function [11]. The nodes in Layer 3 are rule nodes representing fuzzy logic rules. The links in this layer are used to implement precondition or antecedent matching for generating fuzzy logic, that is, to calculate the applicability of each rule. Rule nodes perform the fuzzification and operations:

$$f = \min(x_1^{(2)}, x_2^{(2)}, \dots, x_n^{(2)}). \quad (6)$$

The connection weight of this layer is 1.

The conclusion of the rule node is defined by the network connection in Layer 4, that is, to generate the output of each rule corresponding to the input. It is a normalized calculation, with the number of nodes the same as that in Layer 3, namely:

$$f = \sum_{i=1}^n x_i^{(3)}, \text{ and } \bar{\alpha}_j = \frac{x_j^{(3)}}{f}. \quad (7)$$

Layer 5 is the defuzzified output layer that is used to generate the total output of the control rules, namely:

$$y_k = \sum_{j=1}^m w_{kj} \bar{\alpha}_j. \quad (8)$$

5.3. Learning algorithm of fuzzy neural network

Based on the above structure of the fuzzy neural network, the input component is predetermined, so only the connection weight of the last layer w_{ij}

($i = 1, 2, \dots, n; j = 1, 2, \dots, m$) and the center value c_{ij} and width σ_{ij} ($i = 1, 2, \dots, k; j = 1, 2, \dots, m_i$) of membership function in Layer 2 need to be learned. Since the network structure is essentially also a multi-layer feed-forward network, the learning algorithm for tuning parameters can be designed by using the error back-propagation algorithm like BP network [12]. The error cost function is expressed by $E = \frac{1}{2} \sum_{i=1}^2 (y_{di} - y_i)$. Where

y_{di} and y_i are, respectively, the desired output and the actual output value [13]. The tuning parameters, w_{ij} , c_{ij} and σ_{ij} , are calculated using error back-propagation algorithm given below [14]. For the fuzzy neural network as shown in Figure 2, the function of the node at each layer is given below:

Layer 1:

$$f_j^{(1)} = x_i, x_i^{(1)} = g_i^{(1)} = f_i^{(1)}; i = 1, 2, \dots, n. \quad (9)$$

Layer 2:

$$f_{ij}^{(2)} = \frac{(x_i^{(1)} - c_{ij})^2}{\sigma_{ij}^2}, \quad (10)$$

$$x_{ij}^{(2)} = u_i^j = g_{ij}^{(2)} = \exp(f_{ij}^{(2)}) \exp\left[-\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}\right], \quad (11)$$

where $i = 1, 2, \dots, n; j = 1, 2, \dots, m_i$.

Layer 3:

$$f_j^{(3)} = \min\{x_{1i_1}^{(2)}, x_{2i_2}^{(2)}, \dots, x_{ni_n}^{(2)}\} = \min\{u_1^{j_1}, u_2^{j_2}, \dots, u_n^{j_n}\}, \quad (12)$$

where $j = 1, 2, \dots, m; m = \prod_{i=1}^m m_i$.

Layer 4:

$$f_j^{(4)} = \frac{x_j^{(3)}}{\sum_{i=1}^m x_i} = \frac{\alpha_j}{\sum_{i=1}^m \alpha_i} \quad (13)$$

and

$$x_j^{(4)} = g_j^{(4)} = f_j^{(4)} = \bar{\alpha}_j; j = 1, 2, \dots, m.$$

Layer 5:

$$f_i^{(5)} = \sum_{j=1}^m x_j^{(4)} = \sum_{j=1}^m w_{ij} \bar{\alpha}_j \quad (14)$$

and

$$x_i^{(5)} = y_i^{(5)} = g_i^{(5)} = f_i^{(5)}; i = 1, 2, \dots, k.$$

Firstly, calculate with the following equation:

$$\delta_i^{(5)} = -\frac{\partial E}{\partial f_i^{(5)}} = y_{di} - y_i, \quad (15)$$

$$\frac{\partial E}{\partial w_{ij}} = -\delta_i^{(5)} x_j^{(4)} = -(y_{di} - y_i) \bar{\alpha}_j, \quad (16)$$

$$\delta_j^{(4)} = -\frac{\partial E}{\partial f_j^{(4)}} = \sum_{i=1}^k \delta_i^{(5)} w_{ij}, \quad (17)$$

$$\delta_{ij}^{(3)} = -\frac{\partial E}{\partial f_j^{(3)}} = \frac{1}{\left(\sum_{i=1}^m \alpha_i\right)^2} (\delta_j^{(4)} \sum_{i=1, i \neq j}^m \alpha_i - \sum_{i=1, i \neq j}^m \delta_k^{(4)} \alpha_k), \quad (18)$$

$$\delta_{ij}^{(2)} = \sum_{k=1}^m \delta_k^{(3)} S_{ij} \exp\left[-\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}\right]. \quad (19)$$

Since the minimum operation is used for $f^{(3)}$, when $g_{ij}^{(2)} = u_i^j$ is the minimum value at the k^{th} rule node, $S_{ij} = 1$; otherwise $S_{ij} = 0$. Then:

$$\frac{\partial E}{\partial c_{ij}} = -\delta_{ij}^{(2)} \frac{2(x_i - c_{ij})}{\sigma_{ij}^2}, \quad (20)$$

$$\frac{\partial E}{\partial \sigma_{ij}} = -\delta_{ij}^{(2)} \frac{2(x_i - c_{ij})}{\sigma_{ij}^3}. \quad (21)$$

The resultant parameter adjustment algorithm is as follows:

$$w_{ij}(t+1) = w_{ij}(t) - \beta_1 \frac{\partial E}{\partial \sigma_{ij}}; i = 1, 2, \dots, k; j = 1, 2, \dots, m, \quad (22)$$

$$c_{ij}(t+1) = c_{ij}(t) - \beta_2 \frac{\partial E}{\partial c_{ij}}; i = 1, 2, \dots, n; j = 1, 2, \dots, m, \quad (23)$$

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) - \beta_3 \frac{\partial E}{\partial \sigma_{ij}}; i = 1, 2, \dots, n; j = 1, 2, \dots, m, \quad (24)$$

where β_1, β_2 , and β_3 are learning rates, and $\beta_1 > 0, \beta_2 > 0$, and $\beta_3 > 0$.

6. Simulation experiment

In this section, the real-time navigation control of the mobile robot in a variable environment was completed by applying the above fuzzy neural network algorithm to solve the issues on the navigation control for mobile robots [15]. In the experiment, the control system was used to remember the fuzzy control rules (input-output pattern samples) summarized by human through the neural network, so that the output pattern approximated that of the learning sample only if the input pattern approximated that of a certain learning sample even if the input value deviated from the learning samples. Based on the control requirements, there were four input signals for defining the fuzzy neural network, which were the sonar sensor detection distances, d_r, d_l and d_c , towards three directions in front of the dummy robot, and the angle α between the movement direction of the robot and the targeted connection, in order to realize target location of the robot. The network outputs were the control line speed v and steering angle increment θ of the robot, which controlled the navigation of dummy robots in the external environment. Next, all the parameters were to be fuzzed. Since the Gaussian membership function curve was smoother, all the membership functions in this thesis were Gaussian type in order to improve the accuracy and simplicity of the system [6]. The fuzzy linguistic variables d_r, d_l and d_c that defined the input variables were $\{N, F\} = \{\text{near, far}\}$, with the domain of (0~30). The fuzzy information from the sensor information was mainly used for obstacle avoidance and emergency handling, its membership function should be determined in such a way that it made a more detailed fuzzification presentation of close-range information. When determining the fuzzy control rule of the mobile robot. If the obstacle was detected in the "far" zone, the mobile robot did not perform obstacle avoidance processing but went straight toward the

target. The membership functions of d_r , d_l and d_c are shown in Figure 3:

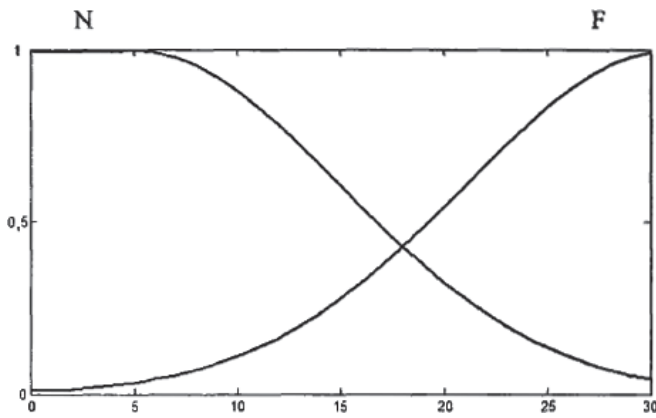


Fig. 3. Degree of membership function of d_r , d_l and d_c

The input angle α were: $\{-60^\circ, -30^\circ, 0^\circ, 30^\circ, 60^\circ\}$, represented by $\{LB, LS, Z, RS, RB\}$, respectively. The increment of the output steering angle θ were: $\{-30^\circ, -15^\circ, 0^\circ, 15^\circ, 30^\circ\}$, represented by $\{TLB, TLS, TZ, TRS, TRB\}$, respectively. The output line speed v were: $\{0.1, 0.25, 0.4\}$, represented by $\{S, M, Q\}$, respectively.

The simulation experiment was implemented on the Microsoft Visual C++ 6.0 software platform based on the following steps: firstly, initialized the weight of the network. According to the possible environment category and target level in the operation of the mobile robot, the training rules of the fuzzy neural network were prepared based on the experience of human navigation experts [16]. The empirical analysis showed that the control factors affecting the steering of the mobile robot were mainly the obstacle distance information in the left, front and right directions and the angle between the robot movement direction and the targeted connection; the main factor affecting the speed of the mobile robot line was the distance information for obstacles. In other words, the closer the distance of the mobile robot to the obstacle, the slower the movement speed of the mobile robot, and the greater the angle of rotation of the mobile robot to avoid obstacles. Otherwise, the mobile robot was able to move toward the target as quickly as possible. Therefore, the values corresponding to the output variables v and θ may be designed depending on the different conditions of the obstacle when establishing the fuzzy control rule [7]. The following 40 rules for control of robot navigation in this experiment have been developed (r1~r40):

1) Obstacles far away or no obstacles at all:

- r1: If the input d_r was F, d_c was F, d_l was F, and α was LB, then the output θ was TLB, and v was Q;
- r2: If the input d_r was F, d_c was F, d_l was F, and α was LS, then the output θ was TLS, and v was Q;
- r3: If the input d_r was F, d_c was F, d_l was F, and α was Z, then the output θ was TZ, and v was Q;
- r4: If the input d_r was F, d_c was F, d_l was F, and α was RS, then the output θ was TRS, and v was Q;
- r5: If the input d_r was F, d_c was F, d_l was F, and α was RB, then the output θ was TRB, and v was Q.

- 2) The left sensor detected an approaching obstacle:
 - r6: If the input d_r was F, d_c was F, d_l was N, and α was LB, then the output θ was TZ, and v was M;
 - r7: If the input d_r was F, d_c was F, d_l was N, and α was LS, then the output θ was TZ, and v was M;
 - r8: If the input d_r was F, d_c was F, d_l was N, and α was Z, then the output θ was TZ, and v was M;
 - r9: If the input d_r was F, d_c was F, d_l was N, and α was RS, then the output θ was TRS, and v was M;
 - r10: If the input d_r was F, d_c was F, d_l was N, and α was RB, then the output θ was TRB, and v was M.

3) Both the left sensor and the front sensor detected an approaching obstacles:

- r11: If the input d_r was F, d_c was N, d_l was N, and α was LB, then the output θ was TRS, and v was S;
- r12: If the input d_r was F, d_c was N, d_l was N, and α was LS, then the output θ was TRS, and v was S;
- r13: If the input d_r was F, d_c was N, d_l was N, and α was Z, then the output θ was TRS, and v was S;
- r14: If the input d_r was F, d_c was N, d_l was N, and α was RS, then the output θ was TRB, and v was S;
- r15: If the input d_r was F, d_c was N, d_l was N, and α was RB, then the output θ was TRB, and v was S.

4) The left sensor, the right sensor, and the front sensor all detected an approaching obstacles:

- r16: If the input d_r was N, d_c was N, d_l was N, and α was LB, then the output θ was TLB, and v was S;
- r17: If the input d_r was N, d_c was N, d_l was N, and α was LS, then the output θ was TLB, and v was S;
- r18: If the input d_r was N, d_c was N, d_l was N, and α was Z, then the output θ was TRB, and v was S;
- r19: If the input d_r was N, d_c was N, d_l was N, and α was RS, then the output θ was TRB, and v was S;
- r20: If the input d_r was N, d_c was N, d_l was N, and α was RB, then the output θ was TRB, and v was S.

5) The front sensor detected an approaching obstacle:

- r21: If the input d_r was F, d_c was N, d_l was F, and α was LB, then the output θ was TLB, and v was S;
- r22: If the input d_r was F, d_c was N, d_l was F, and α was LS, then the output θ was TLS, and v was S;
- r23: If the input d_r was F, d_c was N, d_l was F, and α was Z, then the output θ was TRS, and v was S;
- r24: If the input d_r was F, d_c was N, d_l was F, and α was RS, then the output θ was TRS, and v was S;
- r25: If the input d_r was F, d_c was N, d_l was F, and α was RB, then the output θ was TRB, and v was S.

6) The right sensor detected an approaching obstacle:

- r26: If the input d_r was N, d_c was F, d_l was F, and α was LB, then the output θ was TLB, and v was M;
- r27: If the input d_r was N, d_c was F, d_l was F, and α was LS, then the output θ was TLS, and v was M;
- r28: If the input d_r was N, d_c was F, d_l was F, and α was Z, then the output θ was TZ, and v was M;
- r29: If the input d_r was N, d_c was F, d_l was F, and α was RS, then the output θ was TZ, and v was M;
- r30: If the input d_r was N, d_c was F, d_l was F, and α was RB, then the output θ was TZ, and v was M.

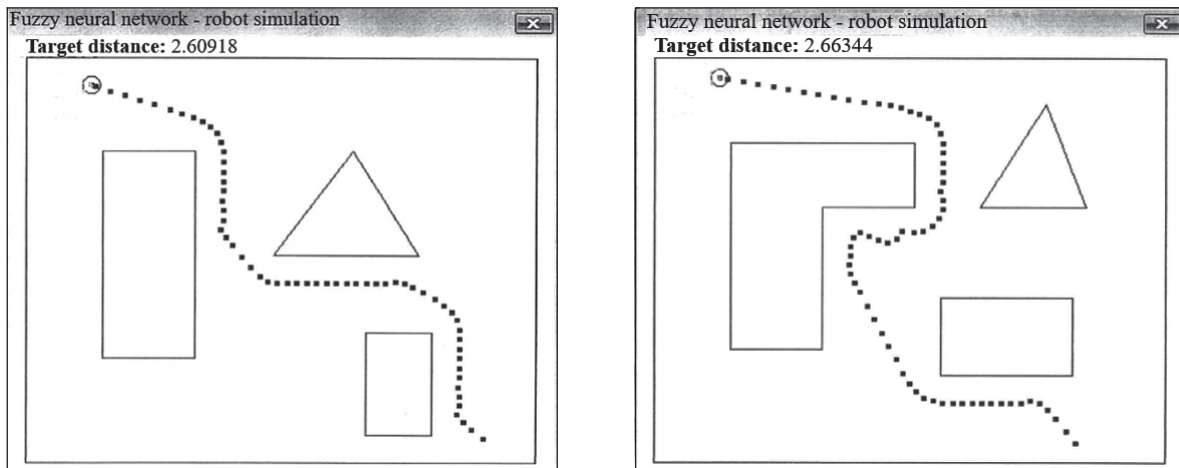


Fig. 4. Diagram of navigation simulation in fuzzy neural network

7) Both the right sensor and the front sensor detected an approaching obstacle:

r31: If the input d_r was N, d_c was N, d_l was F, and α was LB, then the output θ was TLB, and v was S;

r32: If the input d_r was N, d_c was N, d_l was F, and α was LS, then the output θ was TLB, and v was S;

r33: If the input d_r was N, d_c was N, d_l was F, and α was Z, then the output θ was TLS, and v was S;

r34: If the input d_r was N, d_c was N, d_l was F, and α was RS, then the output θ was TLS, and v was S;

r35: If the input d_r was N, d_c was N, d_l was F, and α was RB, then the output θ was TLS, and v was S.

8) Both the right sensor and the left sensor detected an approaching obstacle:

r36: If the input d_r was N, d_c was F, d_l was N, and α was LB, then the output θ was TLB, and v was M;

r37: If the input d_r was N, d_c was F, d_l was N, and α was LS, then the output θ was TLS, and v was M;

r38: If the input d_r was N, d_c was F, d_l was N, and α was Z, then the output θ was TZ, and v was M;

r39: If the input d_r was N, d_c was F, d_l was N, and α was RS, then the output θ was TRS, and v was M;

r40: If the input d_r was N, d_c was F, d_l was N, and α was RB, then the output θ was TRB, and v was M.

The fuzzy neural network might perform learning training based on the navigation control rules upon completion [17]. Considering the convergence speed of network learning, the value was taken as $\beta_1 = \beta_2 = \beta_3 = 0.5$. Upon completion of the network learning, the dummy robot might be directed to perform a navigation and obstacle avoidance in the simulation environment. The dummy robot detected the target and information on the obstacle in the external environment using three analog sonar sensors that were placed in front, while the navigation controller determined the position, distance, direction of movement, etc. of the robot based on the detected information, and then transmitted the information to the internal fuzzy neural network in order to generate corresponding output actions which controlled the dummy robot to avoid obstacles and continuously approach the target area.

The experimental results are shown in Figure 4. After training with the navigation control rule set, dummy robots controlled by fuzzy neural network performed several simulation experiments in different simulation environments. The results indicated that all robots met the requirements of safety since they successfully avoided obstacles in the environment and reached the target area safely without collision. However, the dummy robot sometimes took a long walk since it failed to find the optimal path between the travels in the navigation process. This was mainly attributed to the coverage of the robot's navigation control rule set based on navigation experience summarized by people was not sufficient to cover all possible complex environmental state spaces. In addition, some paths might not be smooth enough due to the insufficient quantity of sensors and lack of comprehensive information during the operation of the robot. However, if the information input was increased, the navigation control rule set might grow rapidly, thus resulting in some defects such as difficulty in preparing the rule base, excessive data calculation, and slow processing speed [18].

7. Conclusion

In this article, the working principle and learning method of neural network was systematically described, several typical neural network models were introduced, and a robot navigation control method based on fuzzy neural network was studied. In the neural network framework, the qualitative knowledge was introduced, namely, the neural network was constructed by the fuzzy rule by adding fuzzy layer to the input layer and the output layer of the conventional neural network. The adaptive learning mechanism of the neural network was retained while the network weight was given with a clear physical meaning. Finally, the fuzzy neural network algorithm was applied to the mobile robot navigation control simulation experiment. Based on the priori knowledge summarized by human navigation experts, the network weight was corrected by self-learning to achieve more effective real-time navigation control.

References

1. Ping P., Petrenko Y. N. Algoritм гибкого управления множественным доступом в мобильной беспроводной системе 5G [The flexible multiple access control algorithm in 5th generation mobile wireless system]. *Informatika — Informatics*, 2018, vol. 15, no. 3, p. 93–101. (In Russian.) Available at: <https://inf.grid.by/jour/article/view/419>
2. Khatib O. Real-time obstacle avoidance for manipulators and mobile robots. *Proc. 1985 IEEE Int. Conf. on Robotics and Automation*. IEEE, 1985, vol. 5, no. 1, p. 500–505. DOI: 10.1109/ROBOT.1985.1087247
3. Khosla P., Volpe R. Superquadric artificial potentials for obstacle avoidance and approach. *Proc. 1988 IEEE Int. Conf. on Robotics and Automation*. IEEE, 1988, vol. 10, no. 2, p. 1778–1784. DOI: 10.1109/ROBOT.1988.12323
4. Glasius R., Komoda A., Gielen S. C. A. M. A biologically inspired neural net for trajectory formation and obstacle avoidance. *Biological Cybernetics*, 1996, vol. 74, p. 511–520. DOI: 10.1007/BF00209422
5. Leon J. A. F., Tosini M., Acosta G. G. Evolutionary reactive behavior for mobile robots navigation. *Proc. IEEE Conf. on Cybernetics and Intelligent Systems*. IEEE, 2004, p. 532–537. DOI: 10.1109/ICCIS.2004.1460471
6. Tuci E., Harvey I., Quinn M. Evolving integrated controllers for autonomous learning robots using dynamic neural networks. *ICSAB: Proc. Seventh Int. Conf. on simulation of adaptive behavior on From animals to animats*. 2002, p. 282–291.
7. Holland J. H. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT Press, 1992. 228 p.
8. Sutton R. S., Barto A. G. Reinforcement learning: an introduction. MIT Press, 1998. 344 p.
9. Tan K. C., Tan K. K., Lee T. H., Zhao S., Chen Y. J. Autonomous robot navigation based on fuzzy sensor fusion and reinforcement learning. *Proc. IEEE Int. Symposium on Intelligent Control*. IEEE, 2002, vol. 28, no. 6, p. 182–187. DOI: 10.1109/ISIC.2002.1157759
10. Carpenter G. A., Grossberg S. ART 2: Self-Organization of stable category recognition codes for analog input patterns. *Applied Optics*, 1987, vol. 26, is. 23, p. 4919–4930. DOI: 10.1364/AO.26.004919
11. Jianhong L., Dezhao C. An enhanced ART2 neural network for clustering analysis. *Proc. First Int. Workshop on Knowledge Discovery and Data Mining*. IEEE, 2008, p. 81–85. DOI: 10.1109/WKDD.2008.117
12. Qian X.-D., Wang Z.-O., Wang Y. A method of data clustering based on improved algorithm of ART2. *Proc. 2005 Int. Conf. on Machine Learning and Cybernetics*. IEEE, 2005, p. 2021–2026. DOI: 10.1109/ICMLC.2005.1527277
13. Leon J. A. F., Tosini M., Acosta G. G. Evolutionary reactive behavior for mobile robots navigation. *IEEE Conference on Cybernetics and Intelligent System*. IEEE, 2004, p. 532–537. DOI: 10.1109/ICCIS.2004.1460471
14. Nelson A. L., Grant E., Galeotti J. M., Rhody S. Maze exploration behaviors using an integrated evolutionary robotic environment. *Robotics and Autonomous Systems*, 2004, vol. 46, is. 3, p. 159–173. DOI: 10.1016/j.robot.2003.11.002
15. Asada M., Noda S., Tawaratsumida S., Hosoda K. Purposive behavior acquisition for a real robot by vision-based reinforcement learning. *Machine Learning*, 1996, vol. 23, p. 279–303. DOI: 10.1023/A:1018237008823
16. Cliff D., Husbands P., Harvey I. Explorations in evolutionary robotics. *Adaptive Behavior*, 1993, vol. 2, is. 1, p. 73–110. DOI: 10.1177/105971239300200104
17. Del R., Millan J. Rapid, safe and incremental learning of navigation strategies. *Proc. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*. IEEE, 1996, vol. 26, is. 3, p. 408–420. DOI: 10.1109/3477.499792
18. Mitchell T. M. Machine learning. New York, McGraw-Hill, 1997. 432 p.

АЛГОРИТМ АВТОМАТИЧЕСКОГО УПРАВЛЕНИЯ НАВИГАЦИЕЙ МОБИЛЬНОГО РОБОТА НА ОСНОВЕ НЕЧЕТКОЙ НЕЙРОННОЙ СЕТИ В ПРОМЫШЛЕННОМ ИНТЕРНЕТЕ ВЕЩЕЙ

П. Пэй¹, Ю. Н. Петренко¹

¹ Белорусский национальный технический университет, г. Минск, Республика Беларусь
220013, Республика Беларусь, г. Минск, пр-т Независимости, д. 65

Аннотация

С появлением технологии беспроводной связи 5G важным и постоянно развивающимся направлением робототехники становятся мобильные роботы, которые широко используются в пространстве промышленного интернета вещей (Industrial Internet of Things — IIoT), в сельском хозяйстве, в армии, на транспорте. Технология автоматического управления навигацией — ядро в этих областях исследований, а также ключевая технология для достижения интеллектуализации и автономизации мобильного робота.

В данной статье подробно рассматриваются и исследуются технологии нейронных сетей и их применение в управлении навигацией, предлагается способ для повышения уровня интеллектуализации мобильного робота с самоорганизацией обучения, придания адаптивных возможностей для решения автономной задачи навигации мобильного робота в сложных неопределенных и неизвестных переменных средах с использованием нейронной сети в сочетании с другими интеллектуальными алгоритмами, такими как обучение с подкреплением и т. п. Решение задачи управления навигацией мобильного робота с использованием нечеткой нейронной сети может обеспечить более эффективную производительность управления в режиме реального времени за счет изменения веса сети путем самообучения в соответствии с априорными знаниями навигации экспертов-операторов.

Ключевые слова: автоматическое управление навигацией, 5G, нейронная сеть, нечеткая нейронная сеть.

DOI: 10.32517/0234-0453-2020-35-1-59-67

Для цитирования:

Пэй П., Петренко Ю. Н. Алгоритм автоматического управления навигацией мобильного робота на основе нечеткой нейронной сети в промышленном интернете вещей // Информатика и образование. 2020. № 1. С. 59–67. (На англ.)

Статья поступила в редакцию: 11 ноября 2019 года.

Статья принята к печати: 21 января 2020 года.

Сведения об авторах

Пэй Пинг, аспирант кафедры «Информационные системы и технологии», Международный институт дистанционного образования, Белорусский национальный технический университет, г. Минск, Республика Беларусь; reiping123456.love@gmail.com; ORCID: 0000-0001-6751-7598

Петренко Юрий Николаевич, пожизненный член IEEE, канд. тех. наук, доцент, доцент кафедры «Информационные системы и технологии», Международный институт дистанционного образования, Белорусский национальный технический университет, г. Минск, Республика Беларусь; ypetrenko@bntu.by

НОВОСТИ**10 технологий, способных кардинально изменить бизнес уже в 2020 году**

CNews Analytics выделил десяток наиболее часто упоминаемых технологий и методологий, которым в 2020 году прочат важную роль в преобразованиях бизнеса.

1. Искусственный интеллект стоит на первых местах практически во всех прогнозах. Несмотря на опасения, что машины отнимут рабочие места у людей, пока благодаря ему компьютеры стали выполнять за людей широкий спектр работ. ИИ-решения способны и автоматически создавать метаданные для звукозаписей с возможностью поиска по ним, и обеспечивать кибербезопасность, анализируя в реальном времени массивы данных о поведении компонентов корпоративной сети и выдавая рекомендации по решению возникающих проблем.

2. Кроме того, ИИ — основа текстовых и голосовых чат-ботов, пользующихся технологиями обработки естественного языка и анализа тональности высказывания.

3. В том же ряду стоят средства роботизации бизнес-процессов (Robotic Process Automation, RPA), которые позволяют автоматизировать повторяющиеся действия пользователей. Причем если первоначально предполагалось, что они будут делать исключительно «механическую» работу — скажем, переносить данные из одной таблицы в другую, — то в последнее время они очень быстро «умнеют» и готовы выполнять контрольные функции и даже раздавать простые, повторяющиеся задания.

4. Методологию Agile, изначально использовавшуюся разработчиками ПО, перенимают и бизнес-структуры. Взаимодействие в географически распределенных командах обеспечивается средствами конферент-связи, которая переживает новый подъем. Не так давно специалисты Массачусетского технологического института провели опрос примерно в трех сотнях компаний, выясняя, какие технические средства помогают создавать наилучшие условия работы сотрудников, и на первое место среди ответов неожиданно вышло видео: инвестиции в видеотехнологии способствуют инновации, слаженному взаимодействию и росту продуктивности. В дальнейшем эффект присутствия при взаимодействии по видео станет еще более глубоким благодаря технологиям дополненной и виртуальной реальности, предсказывают в МТИ.

5. Также среди влиятельных технологических течений эксперты упоминают DataOps. Сам этот термин впервые был предложен всего пять лет тому назад в блоге IBM, и сегодня под ним понимают процессно-ориентированную методологию организации работы групп аналитиков и исследователей данных, направленную на повышение качества аналитических сведений и ускорение их полу-

чения. Помимо принципов организации деятельности, черпающих идеи из Agile-разработки, DataOps опирается на управление данными при помощи средств машинного обучения и искусственного интеллекта. На сегодня, по данным Gartner, DataOps внедрили лишь меньше 1 % участников адресуемого рынка, но аналитики убеждены, что этим организациям гарантировано большое конкурентное преимущество.

6. Вокруг сетей 5G раздута большая шумиха, на них возлагаются большие надежды, хотя работа над стандартами еще не завершена, и до развертывания сетей в общегосударственных масштабах еще могут пройти годы. Тем не менее, в отдельных организациях уже планируют стратегии внедрения систем связи 5G, которые обещают скорость передачи к абоненту и от него на скорости до 20 Гбит/с и до 10 Гбит/с соответственно, а также возможность сохранения соединения при передвижении со скоростью до 500 км/ч.

7. Стандартизация 5G сулит рост применения систем интернета вещей и периферийных вычислений и соответствующее резкое возрастание доступных для анализа потоков данных, к чему предприятиям придется подготовиться путем развертывания надежных архитектур поточной обработки. По информации ассоциации CompTIA, приблизительно в трети компаний уверены, что стратегии в области интернета вещей способны обеспечить рост выручки за счет увеличения объемов производства, монетизации данных и продажи продуктов в виде сервисов.

8. Растет интерес к контейнерам и микросервисам, как к технологиям, позволяющим быстро разрабатывать высокомасштабируемые программные системы, особенно облачные и предназначенные для решений на базе интернета вещей.

9. Опросы, проведенные Gartner в 2019 году, свидетельствуют о начале широкого применения «цифровых двойников» — продуктов, процессов и систем, позволяющих контролировать состояние «исходных» физических объектов и предсказывать возможные проблемы до того, как они возникнут. По общему мнению, в ближайшие годы начнут появляться «цифровые близнецы» организаций, которые позволят вести мониторинг компаний в целом, а не их отдельных систем.

10. И, наконец, что касается блокчейна, вокруг которого тоже немало шума. Возможностей заработать на первичных предложениях монет (ICO) все меньше, но ширится применение распределенных реестров в цепочке поставок, финансовой отрасли и не только, в том числе на базе платформ с открытым кодом, таких как Hyperledger Fabric.

(По материалам CNews)