An intelligent voice recognition system based on fuzzy logic and the bag-of-words technique

O S Stefanenko, L V Lipinskiy, A S Polyakova, J A Khudonogova and E S Semenkin

Reshetnev Siberian State University of Science and Technology, 31 Krasnoyarskii rabochii prospekt, Krasnoyarsk, 660037, Russia

E-mail: polyakova-o.s@mail.ru, lipinskiyl@mail.ru, polyakova_nasty@mail.ru, j.khudonogova@gmail.com, eugeneseumenkin@yandex.ru

Abstract. This paper describes a method for recognizing voice command based on a fuzzy logic system capable of perceiving fuzzy commands, i.e. commands containing fuzzy terms, for example, ‘close’, ‘closer’, ‘close to’, ‘closer than’, ‘further’ and ‘very far’. The developed approach has the ability to be trained for a specific user. The developed fuzzy logic system is used to recognize linguistically inaccurate commands in order to increase the expressiveness of the language for control of a moving robot.

1. Introduction

Robotics is no longer an elitist field of science, and the usage of robots is no longer limited to highly specialized tasks [1]. However, one of the constraints to the development of robotics is the limited capabilities of human-machine communication [2]. Classic computer manipulators, such as a keyboard, mouse, or sensors, are ineffective when working with robots.

In recent times, the number of studies devoted to human-machine interaction has been increasing rapidly [3]. This field of knowledge contains many disciplines in the field of control, machine learning and others, but does not describe one of it. At the same time, the task of voice control is one of the most critical in general human-machine interaction, since the state of the robot must change as a result of the voice command. It is not always possible to switch to an operator, such as in automated information systems. In addition, voice command often need to be recognized and interpreted in real time, which imposes certain restrictions on the voice control system.

When discussing voice control technologies, it is necessary to distinguish between speech recognition systems [4] and voice-based control systems [5], which, unlike the former, not only directly recognize speech, but also interpret the command and translate it into machine signals. At the same time, commands are rarely expressed in single words, and are more often represented as whole phrases. Changing the word order in these phrases can partially or completely change the meaning.

Classification algorithms for solving the voice command recognition problem can be divided into two large groups: statistical and probabilistic.

The most well-known statistical approach to speech recognition, which is well-established in voice identification and voice control tasks, is the hidden Markov model (HMM). HMM is based on the concept of Markov chains, which can represent any random sequential process that undergoes transitions from one state to another [6].
HMMs allow to perform preliminary formation of a reference database of words or phrases and then to identify the presented sound images based on comparison with this database [6, 7]. One of the types of HMMs is continuous HMMs, where transitions between hidden states and the arrival of observations can occur at arbitrary points of time [8, 9].

Artificial neural networks, starting from feed forward and back-propagation neural networks, multi-layer perceptron (MLP), Hopfield networks (HN), and other more complex variants were used as probabilistic classifiers [10, 11, 12, 13]. Convolutional neural networks (CNN) and deep convolutional neural networks (DCNN) are very different from other types of networks, which are usually used for image processing, less often for audio, but they show an improvement compared to classic MLPs [14].

The complexity is in the use of evaluation words with linguistic uncertainty, such as ‘faster’, ‘slower’, ‘stronger’ and ‘louder’, which greatly complicates the interpretation of the command and the generation of machine control signals.

In addition, for command recognition it is important not only to recognize speech, but also to interpret the received words as a command (taking into account linguistically inaccurate expressions). The development of such an interpreter requires a separate study in order to develop an effective interpretation model and develop a procedure for setting numerical parameters.

In view of the foregoing, in this paper we propose the development of algorithmic support to increase the expressiveness of language commands in voice control systems by introducing decision support systems in the fuzzy logic system for the operation of robots moving in space.

2. Process description
Formally, the task of voice control of a given object can be represented as a purposeful change in the state of an object using voice command (figure 1).

![Figure 1. General control scheme.](image)

Figure 1 shows the classic control scheme of a complex object. The control object is that part of the world around us the state of which we want to control. This object has inputs $X(t)$, outputs $Y(t)$ and some control channels $U(t)$. A complex object is controlled using a control device. As a rule, this is an electronic module based on a microcontroller. This module accepts the control target $Z(t)$ as the main input. Using sensors $D(X)$ and $D(Y)$, it collects information about the state of the control object and generates a control action in the form of some machine command $u(t)$. Then, using the executing mechanism (actuator), this command is sent to the control object itself in the form of $U(t)$. Various hardware drivers, servos, stepper motors and other mechanisms can act as an executive mechanism.

Unlike the classical control scheme for voice control, the control device as a target receives not numerical parameters, but human speech or special sound signals. In this case, the control device must be supplemented with a special operator that translates the sound signal (human speech) into a command in the form of a machine command interpreter (figure 2).
At the same time, the task of recognizing voice command consists in constructing an operator for converting an audio signal (human speech) into commands in the form of a machine command interpreter. In a general case, such a command can be represented as follows: \( K(p_1, p_2, ..., p_n) \), where \( K \) is the kind of action or command, and \( p_1, p_2, ..., p_n \) are relevant arguments.

![Figure 2. Convert speech to machine commands.](image)

For example, the \( \text{goto}(x, y) \) command can be interpreted by the system as a change in the current position, while \( (x, y) \) is the vector defining the final desired position.

The main difficulty in recognizing voice command is the semantic interpretation of the words received and the assessment of fuzzy meanings. Let us consider the operation of the ‘Voice command recognizer’ block in more detail (figure 3).

![Figure 3. Module of the ‘Voice command recognizer’.](image)

An audio signal containing an indication of a person regarding the action that must be performed by the object arrives at the input of the module. At the first stage, speech is cleared of various noises using specialized sound filters. At the second stage, text (speech-to-text) is extracted from the received sound signal. The first two stages relate to a more general task, namely speech recognition, and have good industrial solutions in the form of software libraries. The third block is associated with the use of certain models and its interpretation in the form of a machine command. Currently, this task does not have an effective solution. In this paper, we propose a solution based on fuzzy calculations and the principle of expansion.

3. **Proposed approach**

One of the obvious solutions to the problem of forming a machine command based on a voice message is to set a direct correspondence between the received message and the action being performed, for example, in the form of a table. Let us consider the task of controlling the operation of the servo drive. In machine form, the command can be formed as \( \text{go}(p) \), where \( p \) is the parameter of the command, which can be interpreted as the duration of the command, the magnitude of the PWM signal that is supplied to the servo drive or some other interpretation. It is more convenient to set actions...
corresponding to voice messages in the form of a relationship table (table 1). For definiteness, we assume that $p$ takes values in the interval $[0,1]$.

**Table 1.** Table of relationships between voice messages and control parameter values.

| Command / Control parameter | $p = 0$ | $p = 0.5$ | $p = 1$ |
|-----------------------------|---------|-----------|---------|
| go slowly                   | 1       | 0         | 0       |
| go medium                   | 0       | 1         | 0       |
| go fast                     | 0       | 0         | 1       |

The disadvantage of this approach is the low expressiveness of such voice messages, and as a result, only rough control of the object. The expressiveness of correspondence tables can be increased using fuzzy calculations and the principle of expansion.

Table 1 shows the relationship between the set $go = \{"slowly", "medium", "fast"\}$ and the set $p = \{0, 0.5, 1\}$. By introducing a fuzzy relationship, we can significantly expand the expressiveness of the voice command. The fuzzy ratio is calculated in three stages (figure 4).

![Figure 4. Fuzzy ratio calculation.](image)

At the first stage, a voice message arrives. The voice message is decomposed into key terms related to the type of command and its parameters. Based on these key terms, fuzzification is performed, and as a result, the membership function for each command (in our case, for the element of the set ‘go’) is determined. The result is presented as a fuzzy number (1):

$$
\mu_1(x_1) + \mu_2(x_2) + \ldots + \mu_n(x_n)
$$

where $x_i$ is the $i$-th command, and $\mu_i$ is the $i$-th corresponding membership value.

On the basis of a table of crisp relationships, the principle of expansion allows a corresponding fuzzy relationship with more linguistic expressiveness to be built. For instance:

$$
\frac{0.7}{slowly} + \frac{0.1}{medium} + \ldots + \frac{0.05}{fast}
$$

The meaning of this fuzzy value can be described by the term ‘rather slowly’.

At the second stage, the fuzzy ratio is calculated. Let $x \in X, y \in Y$ be distinct sets. A relation $R$ such that $x R y$ is defined on these sets. On the set $X$, a fuzzy set $A$ and the corresponding membership function $\mu_A(x)$ are defined, and on the set $Y$, the fuzzy set $P$ and the membership function $\mu_P(y)$ are defined. Then, using the extension principle, we can construct a fuzzy relation of the sets $A, P$:

$$
\mu_{a R p}(x) = S_{x \in A} [T(\mu_A(x), \mu_P(a, p))]
$$

where $T, S$ are the t-norm and t-conorm, respectively, for example $\max(.)$ and $\min(.)$. $\mu_A(a)$ is the membership function of the set $A$, and $\mu_P(a, p)$ is the membership of ratio $R$. Since a crisp relation is a special case of fuzzy relation, you can choose as a membership function the very crisp relation presented in table form.

Let us consider an example for the $go(p)$ command. As a result of fuzzification, a fuzzy set $A$ is obtained (4):
The fuzzy relation $\mu_R(a, p)$ is given (table 1). It is necessary to define a mapping of the set $A$ to the set $P$.

As a result, we have:

$$B = \max(\min(0.7, 1), \min(0.1, 0), \min(0.05, 0)) + \max(\min(0.7, 0), \min(0.1, 1), \min(0.05, 0)) + \max(\min(0.7, 0), \min(0.1, 0), \min(0.05, 1))$$

Or (6):

$$B = \frac{0.7}{p = 0} + \frac{0.1}{p = 0.5} + \frac{0.05}{p = 1}$$

At the third stage, we form a control command. To retrieve the value of the control parameter, the defuzzification procedure is used. For example, the weighted sum:

$$p^* = \frac{\sum \mu_b(p) \cdot p_i}{\sum \mu_b(p_i)}$$

In our example (8):

$$p^* = \frac{0 \cdot 0.7 + 0.5 \cdot 0.1 + 1 \cdot 0.05}{0.7 + 0.1 + 0.05} = \frac{0.1}{0.85} \approx 0.12$$

The main difficulty arises when performing fuzzification. It is difficult to build a universal procedure that allows you to efficiently extract membership values from a voice command when forming a fuzzy set $A$. One of the solutions to this problem is to set the corresponding memberships by listing (figure 5).
As a universal set, a virtual toggle switch (controller) acts here, the position of which can be set in the interval [0, 1]. The approach using keywords allows you to set some discrete states of the toggle switch. The more such states there are, the more expressive the fuzzification procedure is. However, at the same time, it is more difficult to control the system because you need to keep in mind all the keywords and remember how they relate to each other. Ultimately, a person chooses the control of a system from a limited set. It is impossible to synthesize a new command for which a keyword is not provided.

This problem can be solved by using natural language analysis technologies. One of the most common is the bag-of-words technique. The essence of this approach is that all the normalized words known to the system are represented in the form of a binary vector. Each vector coordinate is associated with a specific word. If the word occurs in a phrase (voice command), then the coordinate takes the value 1, otherwise 0. Thus, each phrase (voice command) is translated into a vector of zeros and ones according to what words were used in the statement (phrase). Next, using the methods of machine learning, a model is constructed that produces a reaction to the received statement (in our case, this is the position of the virtual toggle switch).

Let us consider in more detail the fuzzification procedure using the bag-of-words technique (figure 6).

![Figure 6. Fuzzification using the bag-of-words technique.](image)

The input of the fuzzification procedure receives a certain statement (phrase). It is assumed that this statement will be in the form of an audio signal. At the first stage, the signal is translated into text. Next, using the tokenization procedure, interjections and prepositions are removed from the text. The result of this transformation is a set of keywords that are contained in the statement. At the stage of lemmatization, all keywords are brought into normal form, for example, nouns into the nominative case. Based on the words received and using the vectorization procedure, the text is translated into the feature space. The result of the operation is a vector of zeros and ones. The vector is fed to a model (for example, a neural network) which determines the position of the virtual toggle switch (taking a value from zero to one).

In order for the model to work effectively, it needs to be trained. The main difficulty here is that standard supervised learning is not possible. The loss function is calculated algorithmically, and the neural network affects the loss function through the control object. It is impossible to find the error gradient in this case.

Let us consider the training procedure in more detail. Fuzzification system training can be represented by the following stages:

1. Data collection and preparation
2. Setting the architecture and model parameters
3. Optimization of model parameters

Let us consider these steps in more detail.

**Collection and preparation of data.** In order to train the system, it is necessary to collect a sample where each change in the behaviour of the system will be associated with a user's voice command. To collect data, you must be able to work in a training mode with the system itself, or emulate the system. The system generates changes in its state, and the user comments on this change with the command...
that, in his opinion, describes this change. Once the data is collected, they need to be cleared. Inevitably, there will be inaccurate commands in the sample, outliers and erroneous comments. Data should be cleared and presented in the form of a table convenient for machine learning.

Setting the architecture and model parameters. It is proposed to take a neural network as a model. Neural networks have proven themselves in reinforcement learning and have flexible configuration tools. To train a neural network, it is necessary to choose an architecture and configure weighting coefficients.

Optimization of model parameters. During parameter optimization, the difference between the response of the system to a voice command and the behaviour of the system that has been commented on by the user is minimized. Let the sample \((X, y)\) be collected during data collection, where \(X\) is the set of voice command converted using the bag-of-words technique, and \(y\) is the corresponding behaviour of the system. Then the optimization criterion can be represented as follows (9):

\[
E = \frac{1}{N} \sum_{i=1}^{N} (S(NN(W, X_i)) - y_i)^2 \rightarrow \min_{W}
\]

Here, \(X_i\) is the \(i\)-th voice command converted to a vector using the bag-of-words technique, \(y_i\) is the corresponding system change that the user commented on with the command \(X_i\), \(NN(W, X_i)\) is the reaction of the neural network to the voice command \(X_i\) when weight coefficients \(W\). \(S\) are the response of the control object to the voice command \(X_i\) and its interpretation by the neural network. The difference between the response of the system and the desired response of the system is minimized by choosing the weights of the neural network. To optimize the weight coefficients, the differential evolution algorithm was used.

4. Experimental results

The proposed approach was tested on the next test task. The control object has \(n\) steps. At each step, the object changes its state. At the next step, the user is informed of the desired state of the object in the next step. The user executes a voice command, under the action of which the object changes its state. The task of the object control is to minimize the difference between the desired and actual behaviour of the object. As the control object, a fuzzy logic system was used with membership functions shown in figure 5, fuzzy relations (table 1) and defuzzification by formula (1). The system was controlled in two modes: 1) by keywords, 2) using a neural network and the bag-of-words technique.

In the process of data collection, the control object completely repeats the desired behaviour (figure 7). In this case, the user's voice command corresponding to the system transitions are recorded. In control mode, at each step, the system under the action of a voice command is transferred from one state to another (figure 8).

**Figure 7.** Data collection.

**Figure 8.** Example of object control using a neural network and the bag-of-words technique.
As can be seen from figure 8, at iterations 1-3, there are discrepancies between the desired behaviour of the object and the actual behaviour, and at iterations 3-5 the object practically repeats the desired state. The red line defines the desired behaviour of the object at the next iteration. According to the results of the system, the quality of control is determined in the sense of criterion E (9).

To compare the two approaches to control, a test solution of the problem was performed by each algorithm at 20 control iterations. For training the neural network, a sample of 200 examples was collected. After filtering the data, 191 values remained in the sample. For fuzzification, a neural network with two hidden layers of ten and five neurons, respectively, was used. To train the neural network, we used the differential evolution algorithm with the following parameters: 100 individuals per 100 iterations. The remaining parameters were taken by default. An example of a single solution to the problem is shown in figures 9 and 10.

![Figure 9](image1.png)  ![Figure 10](image2.png)

**Figure 9.** Control of the system using a neural network. The value of the criterion is 0.002523.  **Figure 10.** Control of the system by keywords. The value of the criterion is 0.002632.

Each task was solved by each approach 30 times. The results were averaged and are shown in table 2.

| Approach | FLS+NN | FLS+KW |
|----------|--------|--------|
| E_min    | 0.000677 | 0.001204 |
| E_max    | 0.007213 | 0.008832 |
| E_mean   | 0.003173 | 0.003963 |
| E_std    | 0.001606 | 0.001807 |

As can be seen from table 2, the statistical indicators for both methods are quite close. The main difference between the approaches was that the approach using the neural network is based on the household vocabulary of the operator, and the operator does not need much time to learn how to control the system. In the second approach, the system operator has to learn terms that are unusual for him, and a lot of time is spent on learning to control the system. The total number of keywords for the neural network approach was nine (table 3): ‘medium’, ‘not very’, ‘a bit’, ‘same’, ‘a little’, ‘hard’, ‘very’, ‘confident’, and ‘slightly’.

For control by keywords, it was necessary to apply 11 commands (all the commands in figure 5 and the command ‘same’ or ‘no change’). Moreover, in order to formulate 11 commands, it was required to use about 12 different words. The nature of system control is also different. When controlling the system using a neural network, the real behaviour of the object in many cases differs from the desired behaviour by a small amount. This leads to a deviation of the object from the desired behaviour. When controlling the system by keywords, the object has few cases of deviation from the desired state, but the deviation itself is more significant than when controlling it using a neural network. This can be explained by the fact that the operator controlling the system uses a lexicon that is not natural for him.
and from time to time makes serious mistakes in control. When controlling the system using a neural network, many small deviations can be explained by the low expressiveness of the operator’s language during the initial emulation of the system at the data collection stage.

5. Conclusion
In this paper, an approach to the recognition of voice command using fuzzy logic systems is suggested. Moreover, two fuzzification methods are proposed: using keywords and using a neural network and the bag-of-words technique. It is shown that both of these approaches can be successfully applied to the task of recognizing voice command, while the approaches have differences. The first approach is based on an artificially created conceptual framework and can be difficult for the operator. The second approach is based on the natural language of the operator, but it requires the use of a certain procedure to build the conceptual framework and system training. Both approaches allow the necessary control accuracy to be achieved and should be applied taking into account the specified details.

At the same time, when controlling the system using a neural network, it is possible to increase the accuracy of control by increasing the expressiveness of the operator’s natural language, for example, by repeating the procedure for collecting data and training the neural network. When controlling the system by keywords, the control accuracy can be improved by introducing additional keywords and with better (longer) preparation of the operator in terms of the established vocabulary.

Acknowledgements
This research is supported by the Russian Foundation for the Promotion of small forms of enterprises within project №14356GU/2019

References
[1] Trofimova A, Wiley T, Pagnucco M and Velonaki M 2019 Exploring Human Machine-Mediated Interaction for Applications in Social HRI. In Proceedings of the 31st Australian Conference on Human-Computer-Interaction 347-351
[2] Johannsen G 2009 Human-machine interaction. Control Systems, Robotics and Automation 21 132-62
[3] Clark L, Doyle P, Garaiáde D, Gilmartin E, Schlögl S, Edlund J [et al.] 2019 The State of Speech in HCI: Trends, Themes and Challenges. Interacting with Computers 31(4) 349-371
[4] Paul R, Beniwal R, Kumar R and Saini R 2018 A Review on Speech Recognition Methods. International Journal on Future Revolution in Computer Science & Communication Engineering, 4(2) 292-298
[5] Manish N and Reddy B J 2015 Review of voice control robot applications. International Journal of Advanced Engineering 1(9) 671-674
[6] Rabiner L R 1989 A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE 77(2) 257-286
[7] Seong T W, Ibrahim M Z, Arshad N W B and Mulvaney D J 2018 A comparison of model validation techniques for audio-visual speech recognition. In IT Convergence and Security Springer, Singapore 112-119
[8] Liu Y Y, Li S, Li F, Song L and Rehg J M 2015 Efficient learning of continuous-time hidden markov models for disease progression. In Advances in neural information processing systems 3600-3608
[9] Mustafa M K, Allen T and Appiah K 2019 A comparative review of dynamic neural networks and hidden Markov model methods for mobile on-device speech recognition. Neural Computing and Applications 31(2) 891-899
[10] Wijoyo S 2011 Speech recognition using linear predictive coding and artificial neural network for controlling movement of mobile robot. In: Proceedings of 2011 International Conference on Information and Electronics Engineering (ICIIEE 2011) 28–29
[11] Eng G K and Ahmad A M 2005 Malay speech recognition using self-organizing map and multilayer perceptron. In Proceedings of the postgraduate annual research seminar 233
[12] Park J, Diehl F, Gales M J F, Tomalin M and Woodland P C 2009 Training and adapting MLP features for Arabic speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing* 4461-4464

[13] Hmad N and Allen T 2013 Echo State Networks for Arabic Phoneme Recognition. In *Proceedings of World Academy of Science, Engineering and Technology* 79. World Academy of Science, Engineering and Technology (WASET) 424

[14] Seltzer M L, Yu D and Wang Y 2013 An investigation of deep neural networks for noise robust speech recognition. In *IEEE international conference on acoustics, speech and signal processing* 7398-7402