Journal of Theoretical and Applied Information Technology
15th July 2023. Vol.101. No 13
© 2023 Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

FORMING COMMANDS FOR VOICE CONTROL IN KALDI ENVIRONMENT BASED ON JSPEECH GRAMMAR FORMAT TECHNOLOGY

AMER TAHSEEN ABU JASSAR[1]

[1]College of Computer Science and Information Technology,
Ajloun National University, Jordan

E-mail: [1]amer_abu_jassa@anu.edu.jo

ABSTRACT

The work is devoted to the study of the features of voice control of a mobile robot and the processing of voice commands. For these purposes, standard IT technologies are used. An approach to solving such a problem based on various voice recognition systems is considered. Four main speech recognition systems are described. A system for recognizing voice commands for controlling a robot is considered. A distinctive feature of the proposed system is that the system does not require a permanent connection to the Internet, and the autonomous operation of such a system has the advantage of lower requirements for working with it.

Keywords: *Formation, Utility, Management, Voice Commands, IT Technologies*

1. INTRODUCTION

Robotics has proven itself in many areas of human activity [1-3]. Robots are used in factories to automate production process, during emergencies for prompt and safe assistance.

Robotic systems reduce costs and optimize time parameters, which allow you to optimize production processes. The use of intelligent robots allows you to perform following basic tasks:

- to receive commands in general form;

- navigate and navigate in confined space, possibly with changing environment;

- manipulate objects of complex, and sometimes unknown in advance;

- voice search, etc.

Speech recognition is an important element of modern robot as it increases its ability to interact with people and, above all, use their most natural forms of communication. Voice control systems (VCS) expand capabilities of robots in terms of mastering more and more complex operations. However, an intelligent robot still requires constant human supervision, especially in connection with possibility of emergency and emergency situations.

A robot cannot function autonomously for sufficiently long time in non-stationary environment, since all its actions are strictly formalized, and in these conditions intuition and creative approach are required. To carry out work, information from the knowledge base is entered into the robot interface. First of all, it is a priori information. Operational information is also introduced, which is formed in the process of studying the environment, performing certain actions by the robot.

The information itself includes description of geometric and other physical characteristics of objects in environment and their relationships. This description has hierarchical structure in form of sequential generalization levels of initial information.

A breakthrough in robotics, in addition to robotic vacuum cleaners, etc., of which there are now an inexhaustible set, was the Second Hands project – robot assistant. He knows how to deliver instruments using voice commands and in future will learn to recognize context of employee's actions [4]. Thus, in robotics, level of full-fledged speech interaction with robots, including speech control, is important indicator of robot artificial intelligence level as whole. The relevance of studying problem of forming commands for voice control in robotics determined choice of this research topic.

2. RELATED WORK

The concept of human-robot interaction has been widely studied over past decades. The researchers sought to create human-robot interface that allows users to communicate with robots in simplest and most natural way for humans. As a result, HRIs were created to control robots based on various means [5, 6].

The choice of technology for development of speech recognition module is presented in [7], but specificity is that we are talking about continuous speech. The structure of speech recognition system and various methods of feature extraction for development of such a system are described. The widely used classification methods in development of speech recognition systems were discussed.

The implementation of voice command recognition systems using cloud technologies is presented in many works. The authors of [8] defined requirements for assessment of various cloud services for controlling robots. The robot control architecture was modeled and implemented. VCS includes main functions of cloud speech processing: speech-based intent recognition, slot filling and dialogue-based interaction.

With help of cloud technologies, industrial cloud robotics is presented in [9]. The difference between this work is that attention is paid to ability to control robot from remote location. The authors used Google's speech recognition services to control robotic arm. We used Android Speech API in custom Android app that takes speech signal, decrypts it, and sends it to server.

Numerous works are devoted to implementation of VCS based on local systems [10-14].

An overview of Kaldi design, free and open source toolkit [10]. In [11], using Kaldi toolkit, model of continuous speech recognition in Punjabi language is presented. Russian speech recognition using Kaldi is described in [12]. The study of acoustic modeling was carried out. The models were tested on problem of recognizing continuous Russian speech with very large vocabulary. An open source CMU Sphinx-4 based system is described in [13]. In [14], technological aspects of VCS development based on free SPHINX 4 technology are considered. Construction examples of simplest grammars and acoustic models are considered.

Comparison of speech recognition systems: Microsoft API, Google API and CMU Sphinx is presented in [15]. Over past decades, large number of studies have also been carried out on possibilities and feasibility of using neural networks for speech recognition [16-18]. Recognition of voice commands in intelligent systems using deep neural networks is described by authors in [16]. Features of neural networks for voice transformation are presented in [17]. Speech recognition using deep neural networks is described in [18].

Thus, the fundamental point in the formation of a command for voice control is the choice of the environment and technology for constructing such procedures. At the same time, an important aspect is the ability to control without a permanent connection to the Internet. It is these details that we will pay attention to in the future.

3. OVERVIEW OF VOICE CONTROL SYSTEMS FOR ROBOTS

Systems of speech recognition and understanding for various purposes and for various languages have been created and are being produced.

Most of them are designed for speech of any person. In figure 1 shows generalized diagram of VCS structure.

During analysis of subject area [1-19], it was determined that today there are different voice control systems that differ:

- number of recognized words;

- by designation, for example, there can be command systems or text dictation system "Speech-Text", etc.;

- differ in level of command recognition, for example, recognition of isolated words or continuous speech.

- by type of structural unit, for example, allophone, phoneme, diphon, triphon, word or phrase;

- depending on language of pronunciation;

- by type of dictionary (systems with large dictionary / with small dictionary);

- by type of work (without / with access to Internet);

- depending on speaker (speaker-oriented, speaker-independent, automatic tuning systems for speaker), etc.

Command processing system: analysis of isolated voice commands, words and phrases with clearly defined boundaries of beginning and end and includes following stages: preliminary correction, segmentation into fragments, determination of informative parameters and recognition.

In course of analysis [16-19], it was determined that neural network with following parameters can be used as tool for implementing voice command recognition systems:

- input image – recognized harmonic;

- output signal – function of harmonic belonging to dictionary words;

- desired output signal;

- structure of neural network;

- activation and error function;

- criterion of training quality – minimum error over entire training set;

- learning – back propagation of error.

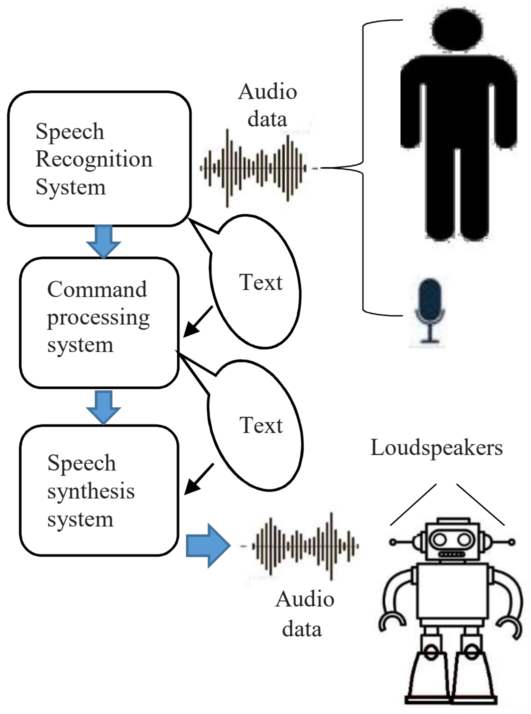


Figure 1: Generalized diagram of VCS structure

Neural networks are best option, as they can be used to recognize continuous speech of increased complexity and to highlight syllables, morphemes and words.

VCS can be implemented on the basis of 4 main speech recognition systems:

CMUSphinx is a speech recognition system that combines a number of tools for tasks of varying complexity;

Kaldi – provides support for FST using the OpenFST open source libraries; linear algebra - BLAS1 (Basic Linear Algebra Subroutines) and LAPACK2 (Linear Algebra PACKage) libraries;

Google Cloud Speech API and Yandex SpeechKit are paid services.

Each of these systems has its own characteristics:

1. CMUSphinx [11, 12, 13]:

- open source;

- work without access to Internet;

- ability to create your own dictionary;

- does not consume a lot of resources with small dictionaries;

- can be used to create complex speech recognition systems;

- low speed and accuracy of recognition;

- difficulties with integration with Russian language.

2. Kaldi [8, 9, 10]:

- open source;

- work without access to Internet;

- system is free;

- integration of Finite State Transducers (FSTs) and extensive use of linear algebra;

- flexibility and extensibility;

- recognizes both continuous speech and single words;

- can be used to create complex speech recognition systems;

- works only with audio files, which will create significant delays when system is working in real time.

3. Google Cloud Speech API [13]:

- cloud system;

- paid service;

- allows you to recognize, voice or translate any text in many languages;

- used as an application system for voice input function;

- ability to use powerful models of neural networks in API;

- need for permanent connection to Internet server and delay in receiving response;

- recognition speed;

- high quality speech recognition and synthesis;

- resistance to noise;

- ability to filter wrong content;

- extensive vocabulary;

- live streaming or pre-recorded audio.

4. Yandex SpeechKit [18, 19]:

- cloud system;

- paid service;

- ease of integration (Yandex products support all major industrial standards and have detailed technical documentation);

- ability to use powerful models of neural networks in API;

- recognition speed;

- high quality speech recognition and synthesis;

- flexibility and extensibility;

- extensive vocabulary;

- system recognizes only Russian, English and Turkish;

- need for permanent connection to Internet server and delay in receiving response;

- to gain access to servers, you need to register in Yandex system, conclude an agreement and only then get an API key.

To improve quality of command recognition, noise filter is used to cut out unnecessary sounds. In addition to 4 main ones, we mention such open source systems as HTK and Julius. HTK (Hidden Markov Model Toolkit) is tool for building and modifying HMM. HTK is mainly used in speech recognition research; this system was also used for speech synthesis [20].

The HTK consists of many modules and tools implemented in C language. These tools provide platform for speech analysis, HMM training and testing, and enable construction of complex systems based on HMM.

The Julius system uses two-pass strategy of speech recognition [20, 21]. Based on n-gram language models. Extremely low memory requirements: for trigram models with dictionary of 20 thousand words, no more than 64 megabytes of memory are used. It was originally developed by Japanese researchers for Japanese language, so at moment there are only simplest (several tens of words) language models for English language.

Separately, we also highlight type for recognizing voice commands – autonomous devices (modules) that have found wide application in robotics, for example, Voice Recognition Module (v 3.1) is compact and easy-to-use speech recognition module. Based on this module, you can create projects with voice control. The memory of recognizer is designed for 7 voice commands, which means module is able to simultaneously compare up to 7 voice commands with incoming sound signal. Speech Recognition Accuracy: 99 % (under ideal conditions). First, you need to train this module - write down a set of commands [20, 21].

The speed of recognition of local speech recognition systems is low. The advantage of cloud systems is that all calculations are performed remotely on the server, and this makes it possible to reduce the load on the local device.

We can see various differences when choosing individual systems for generating voice commands. The significance of such differences is difficult to overestimate and even more so to compare. Such systems may be different [8, 12, 22-25]. This is due to the task facing the developer and what solutions he implements. In our work, we consider voice commands to control the robot. Therefore, the features of such control and such commands are disclosed below.

4. FEATURES OF CONTROL COMMANDS

To begin with, let's describe the main functions that should be implemented with voice control [4]:

- opening port for data transmission to robot;

- initialization of robot;

- waiting for code word;

- isolation of command from text;

- checkingteam for compliance;

- team recognition;

- execution of voice command;

- issuing an answer. If system was unable to recognize command, response is generated informing this user. Then system returns to waiting for code word.

Journal of Theoretical and Applied Information Technology
15th July 2023. Vol.101. No 13
© 2023 Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Voice commands can take following values: rotate, move, move, remember, open, close, take, etc. The rules for constructing control commands are always selected in accordance with technical characteristics of robot.

In order for robot to be able to adequately perceive commands given to it, it is necessary to think over structure of voice request. In order for robot to be able to distinguish voice data that does not concern it, code word is needed from requests directed to it – keyword to activate robot's work, for example, ROBOT. A command is direct order to robot, which is given in one word, simple sentence or question.

The general view of command format example can be represented as in figure 2.

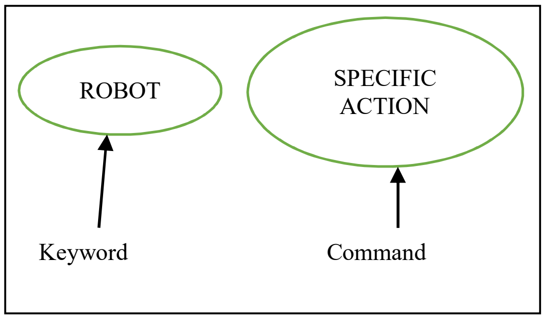
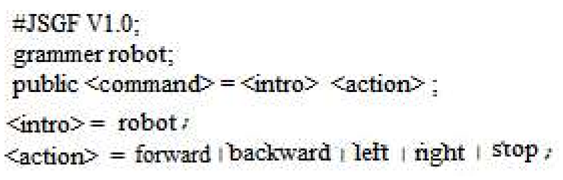


Figure 2: Command format

If command is pronounced immediately after code word and streaming recognition is used, command text is accepted by client while command itself is uttered, and by command time is finished, it starts to process it. Thus, time delay between beginning of user's access to system and beginning of issued command execution is minimized.

The simplest version of such grammar based on four commands: "robot forward", "robot backward", "robot to the left", "robot to the right" and "robot stop" is shown in figure 3 [14].



```
#JSGF V1.0;
grammer robot;
public <command> = <intro> <action> ;
<intro> = robot ;
<action> = forward | backward | left | right | stop ;
```

Figure 3: Grammar of four commands

It used JSpeech Grammar Format (JSGF) technology, platform-independent textual representation of grammars for use in computer speech recognition. JSGF adopts style and conventions of Java language in addition to using traditional grammatical notation.

Command grammars are usually used to describe simple languages designed to issue commands or control something. Usually grammars are written by hand or generated by an algorithm. Most often, grammars do not contain statistical probabilities, but some elements may have weights.

Grammars allow you to very accurately determine structure of recognized speech. For example, certain word can only be repeated two or three times. However, such strictness of rules can be harmful if user, for example, misses a word required by grammar in recognized structure. In this case, entire phrase may not be recognized.

For this reason, grammars are usually made more free than strings of words in strictly defined order. In grammar, it is preferable to use simple rules [14], without many choices, since this slows down recognizer.

This is not to say that voice control system will execute limited number of commands, list will be different for different robots.

The standard functions implemented with help of voice control system are control of robot movement (navigation), take / release, open / close, move objects, and implement individual operations of technological process.

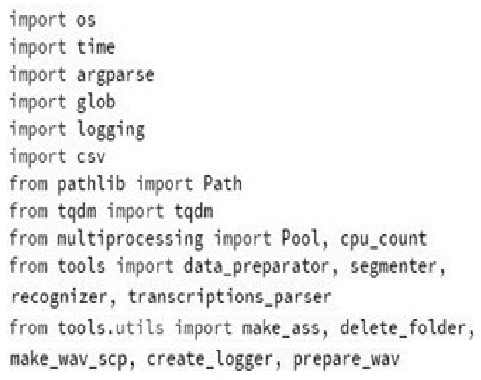
5. FEATURES OF COMMANDS FORMATION FOR VOICE CONTROL IN ROBOTICS

Here we will present our own implementation of recognition and add our version of formation of 4 basic commands for voice control in robotics.

Neural networks were chosen to implement voice command recognition system. A more detailed implementation of voice commands recognition for robot controlling based on neural network is presented by us in [1].

Kaldi was chosen as environment and Python was chosen as programming language. The system was implemented with idea of maximum decomposition to be flexible and expandable.

The first step in program writing is to import libraries from figure 4.



```
import os
import time
import argparse
import glob
import logging
import csv
from pathlib import Path
from tqdm import tqdm
from multiprocessing import Pool, cpu_count
from tools import data_preparator, segmenter,
recognizer, transcriptions_parser
from tools.utils import make_ass, delete_folder,
make_wav_scp, create_logger, prepare_wav
```

Figure 4: Importing libraries and packages

For convenience, logger function was also written, which helps to "identify" errors in program, and if it was not possible to get phoneme to input of neural network. Here is a code fragment for a class for preparing data for speech recognition; initialization of the data preparatory for speech recognition; creating the necessary directories (figure 5).

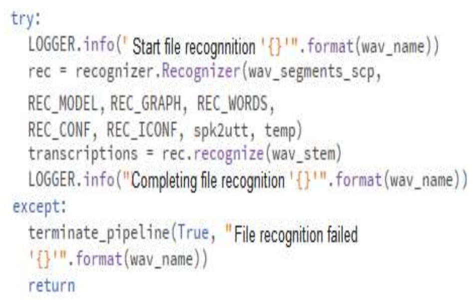
```
class DataPreparator(object):
  """ Class for preparing data for speech recognition"""
  def __init__(self, wav, output, log=False):
    """
    Initialising data preparator for speech recognition
    Arguments:
      wav: path to .WAV audio files
      output: path to prepared files
      log: logging feature
    """
    self.wav = Path(wav)
    self.output = Path(output)
    self.log = log

  def create_directories(self):
    """
    Creating required directories
    Result:
      log_dir: path to directory with log files
      temp_dir: path to directory with temporary files
      ass_dir: path to directory with transcription files
      error_dir: directory path of .WAV files that could
not be recognised
    """
    log_dir = self.output / 'logs'
    temp_dir = self.output / 'temp'
    ass_dir = self.output / 'ass'
    error_dir = self.output / 'error'
    os.makedirs(str(log_dir), exist_ok=True)
    os.makedirs(str(temp_dir), exist_ok=True)
    os.makedirs(str(ass_dir), exist_ok=True)
    os.makedirs(str(error_dir), exist_ok=True)
    return log_dir, temp_dir, ass_dir, error_dir
```

Figure 5: Code fragment for a class for preparing data for speech recognition

The alphacep model is used as acoustic model.

After importing libraries, you need to implement speech recognition start function (figure 6). This function initializes variables for working with phonemes in .wav format. After segmentation, file is recognized using recognizer and transcriptions.

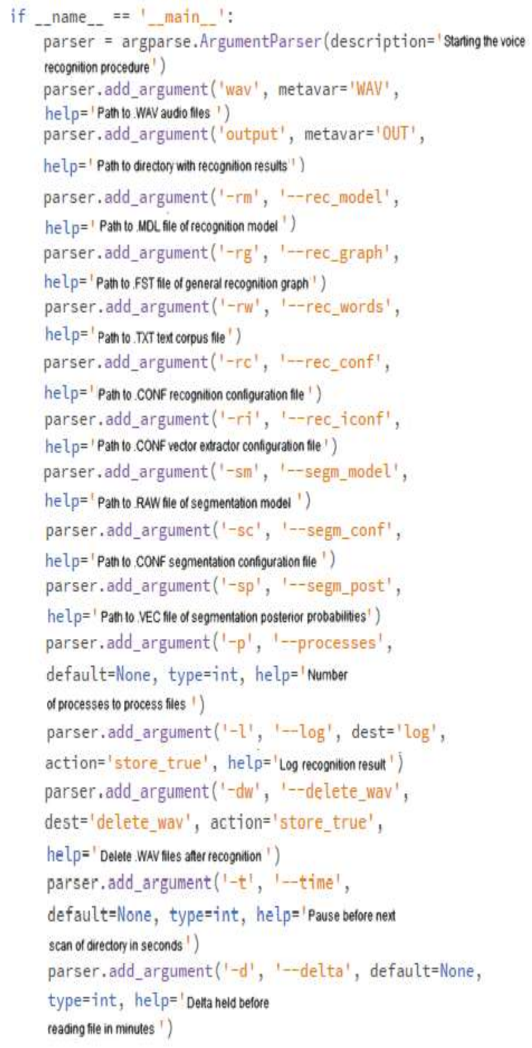


```
try:
  LOGGER.info(' Start file recognnition '{}'".format(wav_name))
  rec = recognizer.Recognizer(wav_segments_scp,

  REC_MODEL, REC_GRAPH, REC_WORDS,
  REC_CONF, REC_ICONF, spk2utt, temp)
  transcriptions = rec.recognize(wav_stem)
  LOGGER.info("Completing file recognition '{}'".format(wav_name))
except:
  terminate_pipeline(True, "File recognition failed
  '{}'".format(wav_name))
  return
```

Figure 6: Code fragment for file recognition

Then code fragment with start of voice command recognition is shown in figure 7.



```
if __name__ == '__main__':
   parser = argparse.ArgumentParser(description='Starting the voice
   recognition procedure')
   parser.add_argument('wav', metavar='WAV',
   help='Path to .WAV audio files ')
   parser.add_argument('output', metavar='OUT',
   help='Path to directory with recognition results')
   parser.add_argument('-rm', '--rec_model',
   help=' Path to .MDL file of recognition model ')
   parser.add_argument('-rg', '--rec_graph',
   help='Path to .FST file of general recognition graph')
   parser.add_argument('-rw', '--rec_words',
   help='Path to .TXT text corpus file')
   parser.add_argument('-rc', '--rec_conf',
   help='Path to .CONF recognition configuration file ')
   parser.add_argument('-ri', '--rec_iconf',
   help='Path to .CONF vector extractor configuration file')
   parser.add_argument('-sm', '--segm_model',
   help='Path to .RAW file of segmentation model ')
   parser.add_argument('-sc', '--segm_conf',
   help='Path to .CONF segmentation configuration file ')
   parser.add_argument('-sp', '--segm_post',
   help=' Path to .VEC file of segmentation posterior probabilities')
   parser.add_argument('-p', '--processes',
   default=None, type=int, help='Number
   of processes to process files ')
   parser.add_argument('-l', '--log', dest='log',
   action='store_true', help='Log recognition result')
   parser.add_argument('-dw', '--delete_wav',
   dest='delete_wav', action='store_true',
   help=' Delete .WAV files after recognition ')
   parser.add_argument('-t', '--time',
   default=None, type=int, help='Pause before next
   scan of directory in seconds')
   parser.add_argument('-d', '--delta', default=None,
   type=int, help='Delta held before
   reading file in minutes ')
```

Figure 7: Code fragment with launch of voice command recognition

Example of implementation of standard commands "move" robot in figure 8.

```
case 1 :
for (pos1 = 20; pos1 <= 60; pos1 += 1) {
robot.write(pos1);
delay(20);}
break;

case 2 :
for (pos1 = 60; pos1 >= 20; pos1 -= 1) {
robot.write(pos1);
delay(20);}
break;

case 3 :
for (pos2 = angle; pos2 >= 0; pos2 -= 1) {
robot.write(pos2);
delay(20);}
angle = 0;

case 4 :
for (pos2 = angle; pos2 >= 0; pos2 -= 1) {
robot.write(pos2);
delay(20);}
angle = 0;
```

Figure 8: Fragment of implementation code for moving robot forward / backward, turning right/left

The proposed system, in which basic commands of robot are implemented (figure 8), is similar in formalization principles to existing analogues, which can also perform similar tasks.

This system is implemented using maximum flexibility of system so that it can be easily adapted to task at hand.

For the experiment, 3 audio files with commands for the robot were taken.

Audio recorded by male and female speakers was used to collect speech data, the results are shown in figure 9. At the same time, the recognition speed is 0.6 sec.

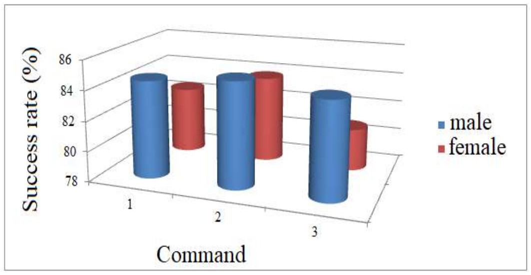


Figure 9: Testing the voice command recognition system

The results obtained are no worse than the systems discussed above. At the same time, it is problematic to directly compare such systems. This is due to the peculiarity of pronunciation, the standardization of the background. However, the proposed implementation is simple and uses standard libraries and technologies. This makes it possible to generate appropriate recognition systems in the field, which expands the possibilities of its application.

In this case, as in [8, 9, 10], we focus on the Kaldi system. This allows us to unify the use of individual commands. In the future, we can build a common interface for the compatibility of such voice control systems.

In comparison with the systems described in [11, 12, 13], our approach allows us to provide a higher accuracy of command recognition and the speed of such recognition.

The use of JSpeech Grammar Format technology further provides the possibility of transferring the developed voice control commands between different systems for which such commands are used. This unifies our development, makes it possible to build various voice control systems on its basis.

6. CONCLUSION

The creation of voice control systems for robot is topical trend in development of information technology. Based on this, paper considers approach to solving problem based on voice recognition systems. For this, generalized diagram of voice control system structure is considered, features of such systems are highlighted, and main speech recognition systems are described. Examples of strategies for working with teams in such VCSs are considered.

As a result of analysis, it was determined that voice control system will execute limited number of commands, and list of such commands will be different for different robots. At the same time, standard functions implemented with help of voice control system are control of robot's movement (navigation), take / release, open / close, move objects, and implement individual operations of technological process. This allows you to unify the creation of various systems that are controlled by voice commands.

The paper shows implementation of author's system for recognizing voice commands for robot controlling based on neural network and implemented using Kaldi and Pyton programming language.

A distinctive feature of proposed system is that system does not require permanent connection to Internet, and autonomous operation of such system has advantage of lower requirements for working with it (less memory). The developed system can be used to create complex speech recognition systems.

As restrictions that are imposed on this development, it is necessary to indicate a limited number of commands to control the robot to perform only the necessary functions. The same applies to the user interface for communication with the robot control system.

REFERENCES:

[1] V. Lyashenko, F. Laariedh, S. Sotnik, and M. A. Ahmad, "Recognition of Voice Commands Based on Neural Network," TEM Journal, Vol. 10, No. 2, 2021, pp. 583-591.

[2] J. H. Baker, V. Lyashenko, S. Sotnik, F. Laariedh, S. K. Mustafa, and M. A. Ahmad, "Some Interesting Features of Semantic Model in Robotic Science," International Journal of Engineering Trends and Technology, Vol. 69, No. 7, 2021, pp. 38-44.

[3] R. Matarneh, S. Maksymova, Zh. Deineko, and V. Lyashenko, "Building Robot Voice Control Training Methodology Using Artificial Neural Net," International Journal of Civil Engineering and Technology, Vol. 8, No. 10, 2017, pp. 523-532.

[4] F. Peller-Konrad, "H²T Projects-Current Projects," Links-https://h2t.anthropomatik.kit.edu/ english/99.php, 2020.

[5] W. Khaewratana, E. S. Veinott, and S. M. Ramkumar, "Development of a Generalized Voice-Controlled Human-Robot Interface: One Automatic Speech Recognition System for All Robots," In 2020 3rd International Conference on Control and Robots (ICCR). 2020, pp. 38-42.

[6] S. Sotnik, S. K. Mustafa, M. A. Ahmad, V. Lyashenko, and O. Zeleniy, "Some Features of Route Planning as the Basis in a Mobile Robot," International Journal of Emerging Trends in Engineering Research, Vol. 8, No. 5, 2020, pp. 2074-2079.

[7] S. Bhatt, A. Jain, and A. Dev, "Continuous Speech Recognition Technologies-A Review," Recent Developments in Acoustics, 2021, pp. 85-94.

[8] C. Deuerlein, M. Langer, J. Seßner, P. Heß, and J. Franke, "Human-robot-interaction using cloud-based speech recognition systems," Procedia CIRP, Vol. 97, 2021, pp. 130-135.

[9] G. Kalaiarassan, C. F. Prashanth, M. Prakash, and S. Phirke, "Speech Recognition based Industrial Cloud Robot for Service-Oriented Sustainable Manufacturing," In IOP Conference Series: Materials Science and Engineering, Vol. 1123, No. 1, 2021, pp. 012047.

[10] D. Povey, and et al., "The Kaldi speech recognition toolkit," In IEEE 2011 workshop on automatic speech recognition and understanding (No. CONF). IEEE Signal Processing Society, 2011.

[11] J. Guglani, and A. N. Mishra, "Continuous Punjabi speech recognition model based on Kaldi ASR toolkit," International Journal of Speech Technology, Vol. 21, No. 2, 2018, pp. 211-216.

[12] I. Kipyatkova, and A. Karpov, "DNN-based acoustic modeling for Russian speech recognition using Kaldi," International Conference on Speech and Computer, 2016, pp. 246-253.

[13] M. Telmem, and Y. Ghanou, "Amazigh speech recognition system based on CMUSphinx," Proceedings of the Mediterranean Symposium on Smart City Applications, 2017, pp. 397-410.

[14] R. Yu. Yuldashev, and et al., "Development of a voice control system for robots based on CMUSphinx toolkit," Alleya nauki, Vol. 3, No. 10, 2017, pp. 827-835.

[15] V. Këpuska, and G. Bohouta, "Comparing speech recognition systems (Microsoft API, Google API and CMUSphinx)," Int. J. Eng. Res. Appl., Vol. 7, No. 3, 2017, pp. 20-24.

[16] A. Sokolov, and A. V. Savchenko, "Voice command recognition in intelligent systems using deep neural networks," IEEE 17th World Symposium on Applied Machine Intelligence and Informatics (SAMI), 2019, pp. 113-116.

[17] S. Desai, and et al., "Spectral mapping using artificial neural networks for voice conversion," IEEE Transactions on Audio, Speech, and Language Processing, Vol. 18, No. 5, 2010, pp. 954-964.

[18] A. B. Nassif, and et al., "Speech recognition using deep neural networks: A systematic review," IEEE access, Vol. 7, 2019, pp. 19143-19165.

[19] J. Tebelskis, "Speech Recognition using Neural Networks," School of Computer Science Carnegie Mellon University Pittsburgh, Pennsylvania, 1996.

[20] F. Barkani, H. Satori, M. Hamidi, O. Zealouk, and N. Laaidi, “Comparative Evaluation of Speech Recognition Systems Based on Different Toolkits,” Embedded Systems and Artificial Intelligence, 2020, pp. 33-41.

[21] S. Kamoliddin Elbobo ugli, K. Shokhrukhmirzo Imomali ugli, and K. Umidjon Komiljon ugli, “Uzbek speech commands recognition and implementation based on HMM,” 2020 IEEE 14th International Conference on Application of Information and Communication Technologies (AICT), 2020, pp. 1-6.

[22] Y. M. Al-Sharo, A. T. Abu-Jassar, S. Sotnik, and V. Lyashenko, “Neural Networks As A Tool For Pattern Recognition of Fasteners,” International Journal of Engineering Trends and Technology, Vol. 69, No. 10, 2021, pp. 151-160.

[23] V. Savanevych, S. Khlamov, V. Vlasenko, Z. Deineko, O. Briukhovetskyi, I. Tabakova, and T. Trunova, “Formation of a Typical Form of An Object Image in a Series of Digital Frames,” Eastern-European Journal of Enterprise Technologies, Vol. 6, No. 2(120), 2022, pp. 51-59.

[24] A. T. Abu-Jassar, H. Attar, V. Yevsieiev, A. Amer, N. Demska, A. K. Luhach, and V. Lyashenko, “Electronic User Authentication Key for Access to HMI/SCADA via Unsecured Internet Networks,” Computational Intelligence and Neuroscience, Vol. 2022, 2022, Article ID 5866922.

[25] S. Khlamov, V. Savanevych, O. Briukhovetskyi, I. Tabakova, and T. Trunova, “Astronomical Knowledge Discovery in Databases by the CoLiTec Software,” In 2022 12th International Conference on Advanced Computer Information Technologies (ACIT), 2022, pp. 583-586.