



Software for Voice Control Robot: Example of Implementation

Svitlana Maksymova¹, Rami Matarneh^{2*}, Vyacheslav V. Lyashenko³

¹Department of Computer-Integrated Technologies, Automation and Mechatronics, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine

²Department of Computer Science, Prince Sattam Bin Abdulaziz University, Al-Kharj, Saudi Arabia

³Department of Informatics, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine

Email: *ramimatarneh@gmail.com

How to cite this paper: Maksymova, S., Matarneh, R. and Lyashenko, V.V. (2017) Software for Voice Control Robot: Example of Implementation. *Open Access Library Journal*, 4: e3848.

<https://doi.org/10.4236/oalib.1103848>

Received: July 26, 2017

Accepted: August 20, 2017

Published: August 23, 2017

Copyright © 2017 by authors and Open Access Library Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Considering software for voice control robot, the main application window is shown. The examples illustrate various robot controls. The structure of the management team is developed. The grammar of various commands is described. A library of words is proposed, which the operator can enter to achieve the set goals. The main elements of the library are listed. To provide voice control, a sequence of actions is provided, which must be followed by the operator. The commands that may be needed in the case of interactive correction of actions are considered.

Subject Areas

Artificial Intelligence, Computational Robotics

Keywords

Robot, Voice Control, Voice Recognition, Fuzzy Logic, XML-File

1. Introduction

The assembly processes design remains an area in which the use of computer aided design tools is small and depends on subjective, human factors. First of all, this is due to the design patterns informality, the inadequate capabilities of 3D and solid modeling tools, the need to combine computational capabilities with artificial intelligence methods.

Modern technological processes CAD (computer-aided design) systems used in robotic production usually provide a learning mode implemented using a software interface. For example, in [1] software enables to exchange data

through supply chain network; in [2] hexapod work space is analyzed using integration of a CAD/CAE system. However, in this case, such an interface is not always able to accurately convey the requirements of the designer. For example, some values of the coordinates of points, speed parameters and directions of movement may be incorrect. An expedient way to solve the problem is to introduce an interactive mode of control and design of technological processes and, as an option, the introduction of voice control. In [3] mobile assistive robot is designed to support older persons in their private home, and the voice control is useful. In [4] audio information is used for robot motion control.

Using voice control will improve the accuracy of the necessary binding of the robot to the coordinates of the working area, including the situation of its dynamic change. In this case, the operator, using voice control, can orient the manipulator in those positions of articulations and those coordinates that are required for performing technological transitions and operations. Using this method, it is possible to significantly simplify assembly technology process design, to perform robotic system functioning operational control. We can see all these advantages in [3] [4] and in others.

2. Materials and Methods

2.1. Voice in the Control System

Voice control problem is considered by many authors. L. Muda, M. Begam, I. Elamvazuthi [5] propose to use Mel Frequency Cepstral Coefficients (MFCCs) as extraction techniques and Dynamic Time Warping (DTW) as features matching techniques. Y. M. Yu, G. Q. Zhao, T. C. Du [6] implemented voice recognition robot by combining of toy robot and a microcontroller SPCE061. Tasevski Jovica, Nikolić Milutin, Mišković Dragiša [7] report a solution for the integration of the industrial robot ABB IRB140 with the system for automatic speech recognition (ASR) and the system for computer vision. Neng-Sheng Pai, Hua-Jui Kuang, Ting-Yuan Chang, Ying-Che Kuo, and Chun-Yuan Lai [8] propose to use after the speech pre-processing of this speech database, the feature parameters of cepstrum and delta-cepstrum were obtained using linear predictive coefficient (LPC) and then to use Hidden Markov Model (HMM) for model training of the speech database, and the Viterbi algorithm was used to find an optimal state sequence as the reference sample for speech recognition. K. Kannan, Dr. J. Selvakumar [9] in their paper “Arduino Based Voice Controlled Robot” describe their EasyVR module, which they train to recognize commands, but there is no detailed description how they recognize them. N. Manish and B. Jagadeesh Reddy [10] propose to use Microsoft Speech API but they don't take into consideration fuzzy commands. Panek P., Mayer P. [3] propose to use a commercially available automatic speech recognition (ASR) serves as base. Mayank Gupta, Pulkit Verma, Tuhin Bhattacharya, Pradip K. Das [11] propose algorithm that is rather interesting but it is not good for industrial conditions. Stanislav Ondas, Jozef Juhar, Matus Pleva, Anton Cizmar, Roland Holcer [12] use HMM-

based acoustic models trained using the SpeechDatE-SK database. Alberto Poncela and Leticia Gallardo-Estrella [13] use Simon, Sphinx, and Julius/Julian.

In every specific condition we can use different methods. We have to understand the main goal. It may be either the highest working speed or the biggest vocabulary or usefulness for the specific language and so on. According to the goal we can choose different methods and approaches.

2.2. General Ideology of Processing Voice Information

Speech recognition process requires specific procedure. It may be described as next sequence. First of all we have to get entire audio information. Then it is necessary to eliminate noise and silence, so to get clear voice. We distinguish separate words audio information energy. Voice information has greater energy than audio without voice. Then we can use different algorithms to recognize words.

For robot control implementation authors propose different methods for preliminary sound processing such as Mel Frequency Cepstral Coefficients (MFCCs), Dynamic Time Warping (DTW) [5].

After preliminary processing linear predictive coefficient and Hidden Markov Model [8] [12] for model training of the speech database [8] and SpeechDatE-SK database, and the Viterbi algorithm was used to find an optimal state sequence as the reference sample for speech recognition. For recognition a lot of authors propose to use Microsoft Speech API [10], commercially available automatic speech recognition (ASR) [4] [7], Simon, Sphinx, and Julius/Julian [13], API Google [14] but it is not specified, in what way control commands library is organized. Many authors propose working algorithms for toy robots [6], and robots based on Arduino [9], such robots cannot be used in industrial conditions.

Automatic speech recognition methods recognize user's speech and are determined by grammars. If sequence included into grammars is recognized recognition mechanism event, which can be processed by application and planned action is performed.

There are three methods for grammars definition: XML-files using [15] [16], binary configuration files (CFG) using [17] or grammars and data structures compiler methods using [18].

Using XML files is the most convenient way to define grammars [15] [16]. This position may be explained by the fact we can create XML-file that will have specific structure. It may be hierarchical. And recognition system after every word is waiting for one word from limited vocabulary. So first of all it waits for word "robot", then for movement type—"rotate" or "move" and so on.

For command implementation using XML-file next action sequence is used:

- 1) Commands block description begins with a key (pre-control) word, for example "Robot";
- 2) The second command phrase word determines movement type: turn (rotate) or move to a point (move).
- 3) Subsequent words constitute parameters associated with a particular com-

mand.

4) The phrase ends with the word “end”, which indicates that the command has been completed and now it is necessary to start checking it for correct assignment and execution.

3. Results and Discussions

So we took an XML-file for library constructing. And for robot voice control system we have to choose such programming environment that may use developed library. Based on the analysis of automatic speech recognition technologies implemented on personal computers, we chose Microsoft Speech Engine as a technology that integrates well with operating systems and is used for HMI (Human Machine Interface), production cell management and dispatch management.

So let us consider library with specified structure development. Used grammar is composed by the type “TopLevelRule” with a predefined initial state, that is, the automatic speech recognition system looks for the pre-control word “Robot” as a precondition to any recognizable command line. This sequence of words makes rules of the second level, and they are used in “TopLevelRule” and are not directly recognizable. The rule is defined for each planned action. Here is the syntax of the command:

Robot <movement_type><parameters> end

where “Robot”–pre-control word,

<movement_type>–word that determines movement type,

<parameters>–words associated with the command,

“end”–word that indicates command input finish.

Using previous statements and positions a software for robot voice control was developed. The main window is shown in **Figure 1**.

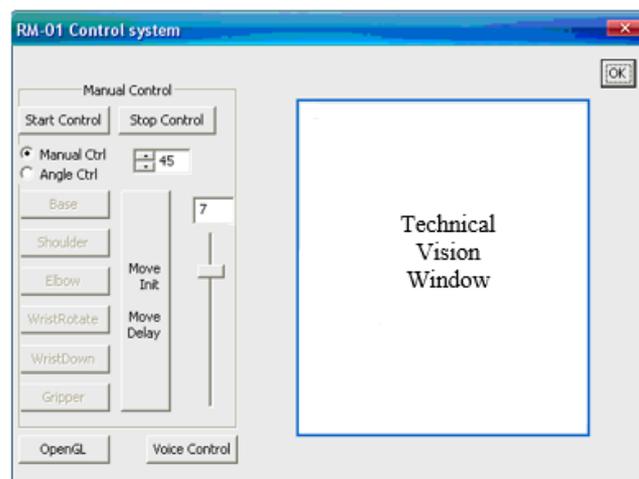


Figure 1. Main Window.

The “StartControl” button connects the robot to control from a personal computer. “StopControl”–disables the robot from the personal computer. The “Manual Ctrl” and “Angle Ctrl” switches allow selecting either a step-by-step moving of the robot, or moving to the corners, respectively. The rotation angle is set by the number in the special control.

The buttons “Base”, “Shoulder”, “Elbow”, “WristRotate”, “WristDown” and “Gripper” allow selecting the link of the robot that needs to be rotated. To initialize the motion or delay, there is a “Move Init/Move Delay”.

In the window, you can select the speed of the robot using the “SliderCtrl” element, the speed value will be indicated in the “Edit” element (in this case, the speed is 7). The “OpenGL” button brings up a window in which the robot’s work is simulated. The “Voice Control” button calls the recognition window, this window, in which the command is directly assigned and checked for compliance with existing tokens, it is shown in **Figure 2**.

When the program is running in recognition mode, XML files are used as a standard grammar description tool for the Microsoft Speech API 5.1 library.

When composing a library of words that can be used in the robot control program text, the following identifiers are also indicated: “words”, “switch”, “joint”, “NUMBERS” (Numbers), “SNUMBERS” (numbers pronounced by one word), “DNUMBERS” (two-digit numbers), “HNUMBERS” (three-digit numbers), “TNUMBERS” (four-digit numbers). The presence of such sections number is explained by English language peculiarities–the language of robot control by means of voice. The numbers from one to twenty are single-word, that is, they can be given by pronouncing one word. Numbers such as “twenty-one”, “twenty-two”, and so on, contain two words in their composition, the first of which denotes tens, and the second–the number from one to nine. Three-digit numbers, for example, “125”, are pronounced in English as “one hundred and twenty-five”. Thus, this number includes in the first place a number from one to nine, indicating the number of hundreds, followed immediately by the very word “hundred” (“hundred”), then either a one-word number (one to twenty),

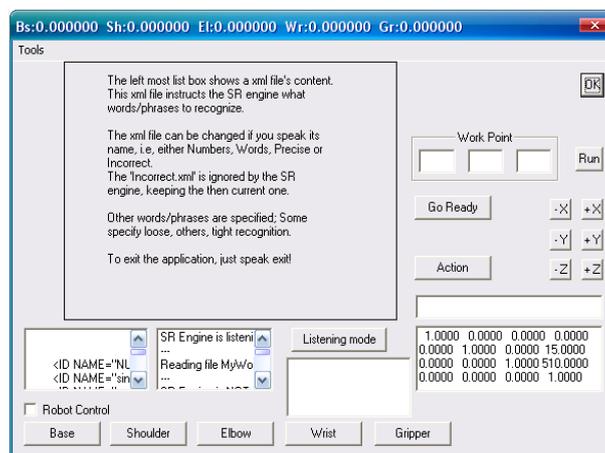


Figure 2. Recognition Window.

or a two-word number. Similarly, you can represent numbers from a thousand to twenty thousand. The number “1125” (four words in Russian) in the English version will sound like “one thousand one hundred and twenty-five”. Here in the first place is the number denoting the number of thousands (from one to twenty), then the word “thousand” (“thousand”) and then everything is analogous to the representation of a three-digit number.

In the “words” section the words “robot” (the beginning of the command), “end” (the end of the command), “exit” (exit from the program) are indicated. In the “switch” section, the words denoting the action type “rotate”, “move” are displayed. The “joint” section includes the names of the links “base”, “shoulder”, “elbow”, “wristrotate”, “wristdown” and “gripper”.

“HNUMBERS”, “TNUMBERS”, each of which has its own structure. “SNUMBERS” includes prime numbers, “DNUMBERS”—tens and prime numbers (“SNUMBERS”), “HNUMBERS”—hundreds and “DNUMBERS”, “TNUMBERS”—thousands and “HNUMBERS”.

$$\begin{aligned} \text{“SNUMBERS”} &= \langle \text{prime numbers} \rangle = 0, 1, \dots, 9 \\ \text{“DNUMBERS”} &= \langle \text{two-digit numbers} \rangle = \langle \text{tens} \rangle + \langle \text{prime numbers} \rangle \langle \text{tens} \rangle = \\ &= 10, 20, \dots, 90 \\ \text{“HNUMBERS”} &= \langle \text{three-digit numbers} \rangle = \langle \text{prime numbers} \rangle + \langle \text{hundred} \rangle + \\ &+ \langle \text{two-digit numbers} \rangle \\ \text{“TNUMBERS”} &= \langle \text{four-digit numbers} \rangle = \langle \text{prime numbers} \rangle + \langle \text{thousand} \rangle + \\ &+ \langle \text{three-digit numbers} \rangle \end{aligned}$$

To specify control commands, they must be pronounced strictly in the order corresponding to existing lexemes. That is, first of all, the pre-command word “robot” is pronounced, followed by the movement type command (“rotate” or “move”). Further, in the case of movement of the “rotate” type, the name of the robot link (“base”, “shoulder”, “elbow”, “wristrotate”, “wristdown”) or “gripper” (grasp).

If you want to rotate in the negative direction (counterclockwise), the word “minus” is pronounced and then the number corresponding to the rotation angle (in the case of the positive direction of rotation (clockwise), the word “minus” is not pronounced).

For example, you need to set the rotation of the robot column to 125 degrees (that is, counter-clockwise). For this, the following command is formed by voice:

robot
move twenty one dot minus nine dot five hundreds forty six
end.

In the case of movement of the “move” type, a triple of numbers is assigned after the given word, corresponding to the values of displacements along the coordinate axes x, y, z. The numbers are separated by a dot-delimiter “dot”,

which denotes the end of the previous number. Similarly to specifying a degree measure of the rotation angle in the “rotate” command, if necessary, in this command, you can set the movement in both the positive direction and the negative direction. If the word “minus” is missing in any coordinate, then moving along this axis should be done in the positive direction. Coordinates are given in strict order: the first number corresponds to the movement along the x-axis, the second—along the ordinate y, the third—along the z-axis. The introduction of only one number means movement only along the x axis, two numbers in the x and y directions. If you need to move along the y and z axes, and there is no x axis, first you set 0, and then move along the necessary axes. Similarly, you can move only along the axis of the applicator: first two zeros, and then the desired number, corresponding to the movement along the z axis. In the same way, any combination of movements of the robot arm grasping along the necessary axes is specified.

Consider an example command:

```
robot
move twenty one dot minus nine dot five hundreds forty six
end.
```

The above example specifies the movement of the manipulator grasp to a point with coordinates (21, -9, 546).

In fact, the keywords “robot” and “end” form a block of functions that provides voice input of information in the robotic production technological processes CAD. Inside this block, you can say any number of “rotate” and “move” commands in any combination. Each command is further tested for correctness of the assignment and feasibility (the ability to perform it from the current position of the robot and under the conditions of the current working area of the robot) and only then performed. Commands are executed sequentially, that is, the verified subcommand is executed, then the next command is started. The execution of commands is terminated in two cases:

- the end of the command (the verification program comes to the word “end”);
- detection of an error in the specified command (if the program finds an error in the command of any command, the execution of the commands is terminated).

Below is an example of setting several subcommands in the same command.

```
robot
rotate shoulder fifty three
move zero dot twelve dot minus seventy eight
rotate elbow minus ninety
end.
```

This command assumes the following actions: rotation of the robot shoulder by 53 degrees, then moving the manipulator to the point with coordinates (0, 12, -78) and the subsequent rotation of the robot's elbow by -90 degrees.

To ensure voice control, follow the sequence below:

- 1) Place the objects in the working area of the robot (as a rule, the component parts are located in the next cell of the working area).
- 2) It is necessary to teach the robot the points of location of the main assembly parts. To do this, use the combination of "rotate", "move" and "fix" commands.
- 3) A number of commands can be compound, they should be combined into meta-commands, for example, "take object A", "place object A at point B".
- 4) Using the control commands of the manipulator and meta commands to set the necessary technological transitions and operations.
- 5) Verify the correctness of the generated process by repeating the assembly process of the product.

In some cases, when designing assembly processes, the operator can not specify the exact commands that would fully ensure the execution of the necessary technological transitions and operations. In such cases it is advisable to use fuzzy commands, for example, "slightly to the left", "even higher", "forward", etc.

Consider fuzzy commands corresponding to the shifts "left", "right", "forward", "back", "up", "down". We will estimate these displacements with the help of the concepts "more", "left/right/forward/back/up/down", "a little to the left/to the right/forward/back" (Slightly left, right, forward, back, up, down) and a "little bit" (a bit). The degree of displacement is shown in **Table 1**.

When using the "rotate" command in this case, the "left" offset is taken as the rotation of the link in the counterclockwise direction, and the "right" shift is the clockwise rotation of the link. If fuzzy shifts are specified in the text of the "move" command, then the "left" offset corresponds to the movement of the robot arm grasp in the negative direction along the abscissa (x), "right"—movement in the positive direction along the x, "forward" and "back" axes (Y), the use of the word "forward" implies a shift in the positive direction, and "back" in the negative direction, respectively, the "up" and "down" shifts determine some movements from the z-axis (z), here "Up" denoting m positive direction of movement, and the "down"—negative.

Table 1. Degrees of displacement.

Direction	Big displacement	Middle displacement	Light displacement	Bit displacement
left	more left	Left	slightly left	a bit left
right	more right	Right	slightly right	a bit right
forward	more forward	Forward	slightly forward	a bit forward
back	more back	Back	slightly back	a bit back
up	more up	Up	slightly up	a bit up
down	more down	Down	slightly down	a bit down

The minimum offset is 10 mm and the maximum offset is 400 mm. Formalization of such a description for a left bias formalization of such a description can be carried out with the help of the following linguistic variable $\langle \beta, T, X, G, M \rangle$, where β is the name of the linguistic variable; T is the set of its values, which are the names of fuzzy variables, the domain of definition of each of which is the set X ; G -syntactic procedure (grammar), allowing to operate with elements of set T ; M is a semantic procedure that allows each new value of a linguistic variable, formed by procedure G , to be transformed into a fuzzy variable. In this case, β is the value of the displacement to the left; $T = \{a_1, a_2, a_3, a_4\} = \{\text{slightly to the left, slightly to the left, to the left, to the left}\}$; $X = [10, 400]$.

Similarly, all linguistic variables can be represented. It should also be noted that quantitative values corresponding to different degrees of bias are determined by experts. In this case, when using the “move” command, “slightly to the left” corresponds to a 10 mm offset in the negative x-direction, “slightly to the left”—100 mm, “left”—200 mm, “even more left”—400 mm. We construct the graphs of the membership of fuzzy sets (**Figure 3, Figure 4**).

Similarly, you can describe all other linguistic variables when using them as part of the “move” command. In the case of the “rotate” command, only the variables “to the left” and “to the right” are applied. For them, the degrees of rotation are assigned the following values: “slightly to the left”—a turn of 10 degrees in the negative direction, “slightly to the left”—to 30 degrees, “left”—to 60 degrees, “more left”—to 90 degrees, all values for The degrees of rotation when

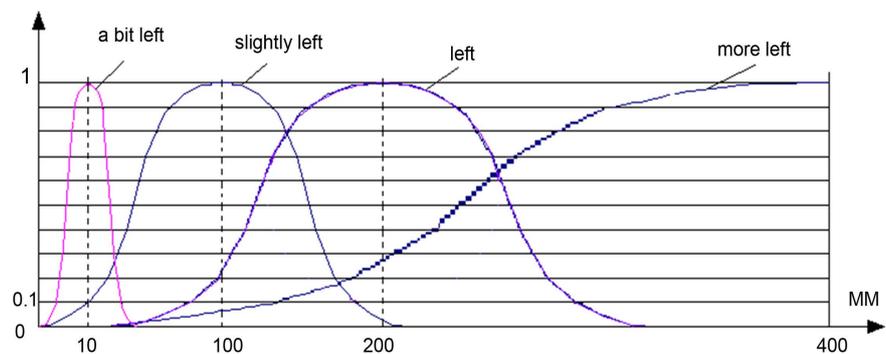


Figure 3. Graphs of belonging for a fuzzy variable “to the left”.

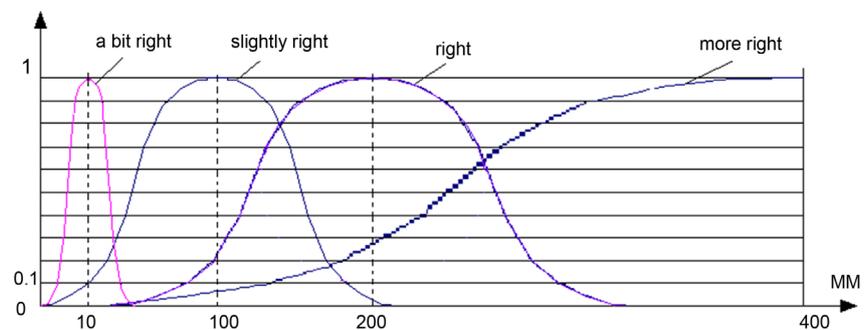


Figure 4. Accessory charts for a fuzzy variable “to the right”.

using the variable “right” are stored with an accuracy to the sign, which corresponds to a rotation in the positive direction.

Thus, the software for voice control of the robot has been developed. The requirements to the working language of the voice control system have been formulated. The library of the words of the working language of the system that make up the working language of the robot is proposed.

The shortcomings of the proposed development include the limited scope of the language of management, as well as the formalization of each phrase that is not natural to the natural language. The advantage is the presence of a key pre-control word, thus it is possible to operate the system in a noisy room.

4. Conclusions

The article considers robot voice control software development. Program main window is displayed, as well as the recognition window, in which the operator can check the correct recognition and interpretation of voice commands. We consider the controls that are present on these windows. Control command structure is developed with a detailed description according to the prevailing grammar. To control the robot, it is necessary to compile a library of words that the operator can input to achieve the set goals. The main elements of this library are discussed and defined through the article. To provide voice control, a sequence of actions is provided, which must be followed by the operator. Also, the commands that may be needed in the case of interactive correction of actions are considered.

Thus, the developed software allows providing industrial robot voice control and can be used in the subsequent design of technological transitions and operations in robotic production technological processes design. We can also combine this software with the ideology of image processing or the methodology of wavelet analysis [19]-[24]. This will create a variety of applications for the use of robots.

References

- [1] Ayağ, Z. (2015) CAD Software Evaluation for Product Design to Exchange Data in a Supply Chain Network. *International Journal of Supply Chain Management*, **4**, 30-38.
- [2] Herbuś, K. and Ociepka, P. (2015) Analysis of the Hexapod Work Space Using Integration of a CAD/CAE System and the LabVIEW Software. *IOP Conference Series: Materials Science and Engineering*, **95**, 012096. <https://doi.org/10.1088/1757-899X/95/1/012096>
- [3] Panek, P. and Mayer, P. (2015) Challenges in Adopting Speech Control for Assistive Robots. In: *Ambient Assisted Living*, Springer International Publishing, Berlin, 3-14. https://doi.org/10.1007/978-3-319-11866-6_1
- [4] Pleshkova, S., Bekiarski, A., Dehkharghani, S.S. and Peeva, K. (2015) Perception of Audio Visual Information for Mobile Robot Motion Control Systems. In: *Computer Vision in Control Systems-2*, Springer International Publishing, New Delh, 135-167. https://doi.org/10.1007/978-3-319-11430-9_6

- [5] Muda, L., Begam, M. and Elamvazuthi, I. (2010) Voice Recognition Algorithms Using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques. arXiv preprint arXiv:1003.4083.
- [6] Yu, Y.M., Zhao, G.Q. and Du, T.C. (2014) Design and Implementation of a Robot Based on Voice Recognition. *Advanced Materials Research*, **875**, 2128-2131. <https://doi.org/10.4028/www.scientific.net/AMR.875-877.2128>
- [7] Tasevski, J., Nikolić, M. and Mišković, D. (2013) Integration of an Industrial Robot with the Systems for Image and Voice Recognition. *Serbian Journal of Electrical Engineering*, **10**, 219-230. <https://doi.org/10.2298/SJEE1301219T>
- [8] Pai, N.S., Kuang, H.J., Chang, T.Y., Kuo, Y.C. and Lai, C.Y. (2014) Implementation of a Tour Guide Robot System Using RFID Technology and Viterbi Algorithm-Based HMM for Speech Recognition. *Mathematical Problems in Engineering*, **2014**, Article ID: 262791. <https://doi.org/10.1155/2014/262791>
- [9] Kannan, K. and Selvakumar, D.J. (2015) Arduino Based Voice Controlled Robot. *International Research Journal of Engineering and Technology (IRJET)*, **2**, 49-55.
- [10] Manish, N. and Reddy, B.J. (2015) Review of Voice Control Robot Applications. *International Journal of Advanced Engineering*, **1**, 671-674.
- [11] Gupta, M., Verma, P., Bhattacharya, T. and Das, P.K. (2015) A Mobile Agents Based Distributed Speech Recognition Engine for Controlling Multiple Robots. *Proceedings of the 2015 Conference on Advances in Robotics*, Goa, 2-4 July 2015, Article No. 28. <https://doi.org/10.1145/2783449.2783477>
- [12] Ondas, S., Juhar, J., Pleva, M., Cizmar, A. and Holcer, R. (2013) Service Robot SCORPIO with Robust Speech Interface. *International Journal of Advanced Robotic Systems*, **10**, 1-11. <https://doi.org/10.5772/54934>
- [13] Poncela, A. and Gallardo-Estrella, L. (2015) Command-Based Voice Teleoperation of a Mobile Robot via a Human-Robot Interface. *Robotica*, **33**, 1-18. <https://doi.org/10.1017/S0263574714000010>
- [14] Lu, C.H. and Hu, Y.L. (2013) Speech Control System for Robot Based on Raspberry Pi. *Advanced Materials Research*, **791**, 663-667. <https://doi.org/10.4028/www.scientific.net/AMR.791-793.663>
- [15] Christodoulides, G. (2014) Praaline: Integrating Tools for Speech Corpus Research. *9th International Conference on Language Resources and Evaluation*, Reykjavik, 28-30 May 2014, 31-34.
- [16] Caines, A., Bentz, C., Graham, C., Polzehl, T. and Buttery, P. (2016) Crowdsourcing a Multilingual Speech Corpus: Recording, Transcription and Annotation of the CROWDED CORPUS. *Proceedings of LREC*, Portoroz, Slovenia, 2016, 23-30.
- [17] Helmke, H., Oualil, Y. and Schulder, M. (2017) Quantifying the Benefits of Speech Recognition for an Air Traffic Management Application. *Konferenz Elektronische Sprachsignalverarbeitung 2017*, Saarbrücken, 15.-17 May 2017, 114-121.
- [18] Sarasa-Cabezuelo, A. and Sierra, J.L. (2013) Grammar-Driven Development of JSON Processing Applications. *2013 Federated Conference on Computer Science and Information Systems (FedCSIS)*, Kraków, 8-11 September 2013, 1557-1564.
- [19] Lyashenko, V., Matarneh, R., Kobylin, O. and Putyatin, Y. (2016) Contour Detection and Allocation for Cytological Images Using Wavelet Analysis Methodology. *International Journal of Advance Research in Computer Science and Management Studies*, **4**, 85-94.
- [20] Lyashenko, V., Matarneh, R. and Kobylin, O. (2016) Contrast Modification as a Tool to Study the Structure of Blood Components. *Journal of Environmental*

Science, Computer Science and Engineering & Technology, **5**, 150-160.

- [21] Lyashenko, V.V., Matarneh, R. and Deineko, Z.V. (2016) Using the Properties of Wavelet Coefficients of Time Series for Image Analysis and Processing. *Journal of Computer Sciences and Applications*, **4**, 27-34.
- [22] Ahmad, M.A., Lyashenko, V.V., Deineko, Z.V., Baker, J.H. and Ahmad, S. (2017) Study of Wavelet Methodology and Chaotic Behavior of Produced Particles in Different Phase Spaces of Relativistic Heavy Ion Collisions. *Journal of Applied Mathematics and Physics*, **5**, 1130. <https://doi.org/10.4236/jamp.2017.55100>
- [23] Lyashenko, V., Babker, A. and Lyubchenko, V. (2017) Wavelet Analysis of Cytological Preparations Image in Different Color Systems. *Open Access Library Journal*, **4**, 1-9.
- [24] Lyashenko, V.V., Ahmad, M.A., Deineko, Z.V. and Rasool, M.H. (2017) Methodology of Wavelets in Relativistic Heavy Ion Collisions in One Dimensional Phase Space. *Journal of High Energy Physics*, **3**, 254-266.



Open Access Library

Submit or recommend next manuscript to OALib Journal and we will provide best service for you:

- Publication frequency: Monthly
- 9 [subject areas](#) of science, technology and medicine
- Fair and rigorous peer-review system
- Fast publication process
- Article promotion in various social networking sites (LinkedIn, Facebook, Twitter, etc.)
- Maximum dissemination of your research work

Submit Your Paper Online: [Click Here to Submit](#)

Or Contact service@oalib.com