Application of an Artificial Neural Network for Planning the Trajectory of a Mobile Robot

Submitted: 08th November 2019; accepted: 30th January 2020

Marcin Białek, Patryk Nowak, Dominik Rybarczyk

DOI: 10.14313/JAMRIS/1-2020/2

Abstract: *This paper presents application of a neural network in the task of planning a mobile robot trajectory. First part contains a review of literature focused on the mobile robots' orientation and overview of artificial neural networks' application in area of robotics. In these sections devices and approaches for collecting data of mobile robots environment have been specified. In addition, the principle of operation and use of artificial neural networks in trajectory planning tasks was also presented. The second part focuses on the mobile robot that was designed in a 3D environment and printed with PLA material. The main onboard logical unit is Arduino Mega. Control system consist of 8-bits microcontrollers and 13 Mpix camera. Discussion in part three describes the system positioning capability using data from the accelerometer and magnetometer with overview of data filtration and the study of the artificial neural network implementation to recognize given trajectories. The last chapter contains a summary with conclusions.*

Keywords: *artificial neural network, mobile robot, machine vision*

1. Introduction

The idea of artificial neural networks (ANN) was taken from natural neurons, which are the basic elements of the nervous system of living organisms, including humans. Neural networks has become the subject of research for various specialties and fields of science, discovering newer and more creative forms of their use. Researchers around the world are developing ANN capabilities, striving to achieve efficiency comparable to living organisms. The number of neurons used is the main comparative scale in these studies. However, unlike real counterparts, they do not transfer signals only, but allow their processing, e.g. by making calculations. The ANN's ability to carry out its tasks is determined by the learning process. The combination of ANN issues and mobile robotics is one of the main currents in the development of modern mechatronics. Planning the trajectory of a mobile robot using SSN requires an approach to two issues. The first is the network itself, which is to recognize the given trajectory. The second is a robot that maps it, which must determine its location in space in a specific way, while avoiding obstacles.

1.1. Orientation of Mobile Robots in Space

Autonomous navigation of mobile robots is one of the main problems faced by their designers, mainly due to the problem of its definition in an unspecified area. To accomplish this task, it is necessary to equip the mobile robot with the sensors to collect data about the environment and its location. Their selection is related to the tasks the robot will perform, and thus the positioning accuracy and type of obstacles, which are mainly characterized by a specific geometry or color are mandatory to be known. Low cost solutions are based on Ultrasonic Sensors (US). The main disadvantages of proposed system was the angle restrictions at which the sound wave falls on the detected surface of the obstacle and the material of which it is made [1]. Nevertheless, they are one of the basic types of sensors used in mobile robots, especially for the implementation of obstacle avoidance tasks [2] and in indoor tasks [3]. It is also possible to track ultrasonic beacons in real time [4] by becoming a transmitter looking for receivers or being a receiver itself. Second popular device is infrared sensor (IR). Characteristic features of obstacles force adjustment of their recognition strategy. The rational approach is to use both ultrasonic and infrared sensors, due to the possibility of mutual complementarity in the detection capabilities [5]. In the case of small mobile robots, the basic environment detection system includes elements such as ultrasonic sensors, infrared sensors, cameras and microphones [6,7]. Global Positioning System (GPS) allows locating receivers based on determining their distance from satellites (at least three) in which they are located, giving data in a quasi-spherical coordinate system (due to the fact that the Earth is not an ideal sphere), which are geodetic coordinates such as: geodetic latitude and longitude and ellipsoidal height or in geographical coordinates: longitude and latitude [8]. The GPS system is designed to locate objects on a larger scale of displacements, expressed in kilometers (square kilometers). That is why it works well with vehicles covering considerable distances in a relatively short time, moving at a much higher speed than mobile robots, being able at the same time to predict its location by projecting the vehicle's direction of movement onto a road map. Robot displacements are much smaller and therefore require greater accuracy [3].

To create maps of the environment, LiDAR sensors are used. They allow to detect objects by measuring laser pulses which are proportional to their distance from the source. For example, determining the position of a robot by measuring distance and angle relative to another robot, using measurements obtained from raw data provided by two laser rangefinders in a 2D plane [9], positioning and orientation of the mobile robot in 3D space, using a laser head measuring the position relative to photoelectric reference points, deployed in the room [10] or the use of an industrial laser navigation system to collect information about the distance between the rotating measuring head and the markers located on the perimeter of the laser beam plane in the area of robot operation [11]. The practical application of LiDAR technology can be autonomous navigation of an agricultural robot in conditions without access to GPS-based solutions [12] or mapping of the environment, through its hybrid representation and robot location [13].

One of the popular devices used to navigate mobile robots has become the Microsoft Kinect system, which is an accessory for the Xbox game console [14]. It's a vision-based system. Kinect is a common tool for navigating a mobile robot, enabling it to avoid obstacles while supplying data, e.g. to an artificial neural network, which deals with environmental recognition and makes decisions about choosing a specific path [14, 15]. An additional advantage of vision system was the creation of algorithms that allow simultaneous localization and building of a map, such as SLAM (Simultaneous Localization and Mapping) [16] and all related solutions such as S-PTAM (SLAM – Parallel Tracking and Mapping) [17]. Solutions for locating robots in a confined space include those based on all kinds of mutual radio communications between mobile robots or reference points. Simple location methods using a local wireless network allow the determination of Euclidean distance between the sample signal vector and references stored in the database [18]. Another use of a wireless network is the use of defined access points to locate an object by measuring signal strength using a compressive sampling theory [19], it enables effective reconstruction of signals from a small amount of data [20]. The considerations to date on the ways of navigating mobile robots are based on different ways of obtaining location data. As previously noted, popular space location and orientation systems such as GPS exhibit lower indoor efficiency. In turn, solutions based on defined reference points limit the robot to operate only in space covered by them. Especially when operating indoor, they give way to other information-based solutions from magnetometers, accelerometers and gyroscopes [21].The data obtained by these sensors can mutually compensate for their errors [22], and in some cases inertial navigation may be valuable [23]. It is based on continuous assessment of the object's position using data from susceptible sensors (such as gyroscope and accelerometer) in relation to the initial position. Not least is also the approach to control the motor drives in accordance with terrain environment [24].

1.2. Artificial Neural Networks

Artificial neural networks have been a field strongly developed over the years among other things in the form of so-called open source projects. Currently, TensorFlow libraries, designed by Google, in November 2015 are one of the most frequently used open source libraries. The vast majority of C ++ were used to write these libraries. Moreover they can use GPU (Graphics Processing Unit) for calculations, what significantly speeds them up. Another advantage of this software is the ability to work on less efficient Raspberry Pi devices or smartphones. ANNs are very often used for all sorts of image classification. Neural networks presented at the International Conference on New Trends in Information Technology[25] is a good example. They can recognize both Arabic script and faces. The architecture of the text recognition network is based on four hidden layers, and its diagram is shown in Fig. 1.

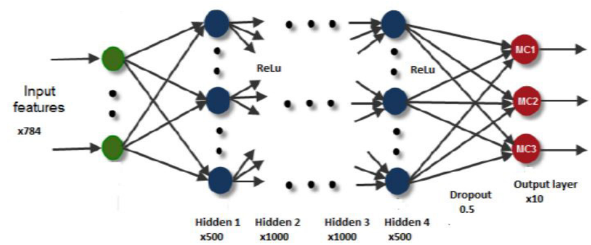

Fig. 1. Network architecture for text recognition [25]

This network was taught based on numbers from zero to ten in 60,000 samples. The image fed at the entrance had a size of 28x28 pixels, which gives a total of 784 pixels for each sample. The value of the number of pixels determines the number of network input neurons shown in Fig. 1. The learning algorithm was based on the back propagation method of the network response error. The network designed in this way has the ability to recognize the images given on its input with an efficiency of 98.46%, which is a high result when it comes to OCR. The second network described in this article is the Convolutional Neural Network (CNN). The architecture of this network is shown in Fig. 2.

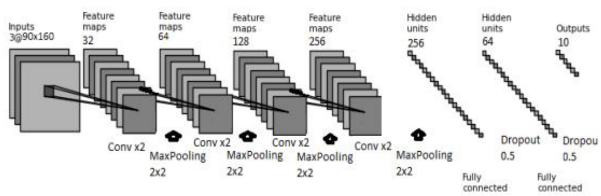

Fig. 2. Convolutional neural network architecture for face recognition [25]

The network shown in Fig. 2 has eight weave layers in four connecting layers, two fully connected layers and an output layer. As the layer activation functions, except for the output layer, where the sigmoidal function was used, the ReLu functions were used. A learning database for the CNN network were photos taken for 10 different students, 50 photos per student. Pictures were taken in different orientations and then subjected to resolution reduction to 90x160 pixels. The photos prepared in this way allowed to start the CNN learning process. The results obtained are shown in Fig.3. The graph shows that for 11 learning epochs the network was able to recognize a given face at about 80%, after 31 epochs it was already about 90%, and for 41 epochs CNN possibilities already reached about 98% and changed further in the learning process slightly.

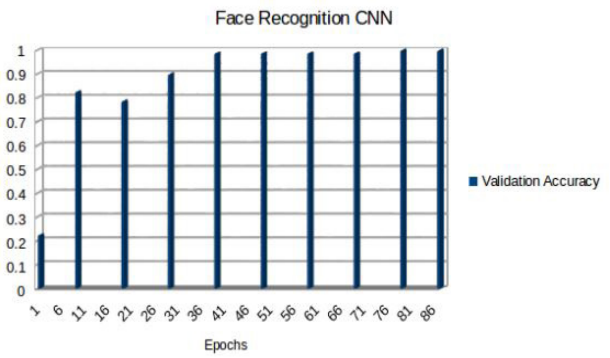

Fig. 3. Graph of CNN recognition capabilities [25]

The MLP deep learning network was used to recognize digits from the MINST database, which contains 60,000 handwritten digits [26]. This network was based on the input layer, with the number of neurons in accordance with the number of pixels present in the image, five hidden layers with the number of neurons, respectively: 2500, 2000, 1500, 1000 and 500, and the output layer consisting of 10 neurons. Backward propagation was used to teach the network and the hyperbolic tangent function was used as an activation function. The network in this configuration achieved very good results, where the error was only 0.35%.

2. Control System

The built-in system allows to choose one of three robot driving modes, by the use of the RC equipment. Remote control, object tracking and ANN pattern recognition. The robot control system is divided into two parts. The first part consists of all the elements in the composition of the mobile robot and includes sensors, a 13 Mpix camera and microcontroller applications for collecting and viewing information (Fig. 4). The second part is the launcher application on a Windows PC (Fig. 5). The application has individual algorithms of an artificial neural network with a multilayer perceptron architecture. Artificial neural network analyses the image sent from a mobile phone camera and on its basis send the tasks to the microcontroller via UART interface.

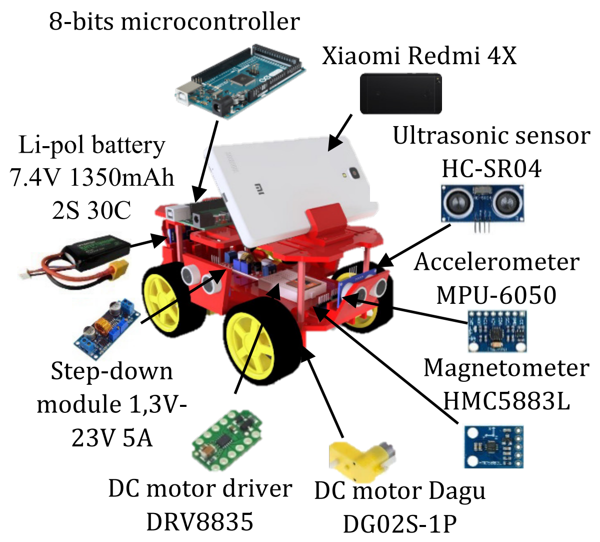

Fig. 4. Placement of robot components

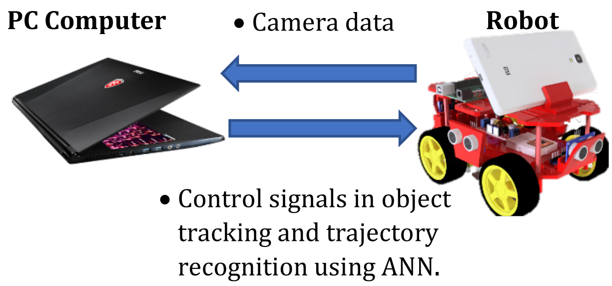

Fig. 5. Block diagram of the overall work of the PC – robot – RC apparatus

3. Mobile Robot Movement

3.1. Data Filtration

In accordance to the information from chapter 2, the robot has been equipped with an accelerometer (MPU6050) and a magnetometer (HMC5883L). The data collected using these two sensors allowed to determine the displacement and direction of robot movement. The device measuring linear or angular acceleration, by measuring it along each axis of the three-dimensional coordinate system: X, Y, Z [27]. In this case accelerometer has measured linear acceleration and, in the process of double integration, it is converted into a displacement [28]. Measurements are recorded only for one axis because the robot's movement is considered in its coordinate system (fig. 6).

Calibration process of the magnetometer bases on data from the MPU6050 accelerometer [29]. This is extremely important because devices of this type are burdened with an error when tilting the system relative to the XY plane. Having regard to presence of the accelerometer it is possible to get an information about angular inclination in the range of 45º. The compensation procedure determines the relative position of the devices to one each other and

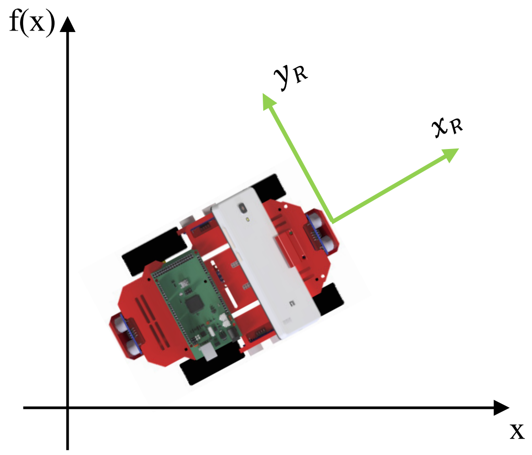

Fig. 6. The robot's coordinate system (xR, yR)

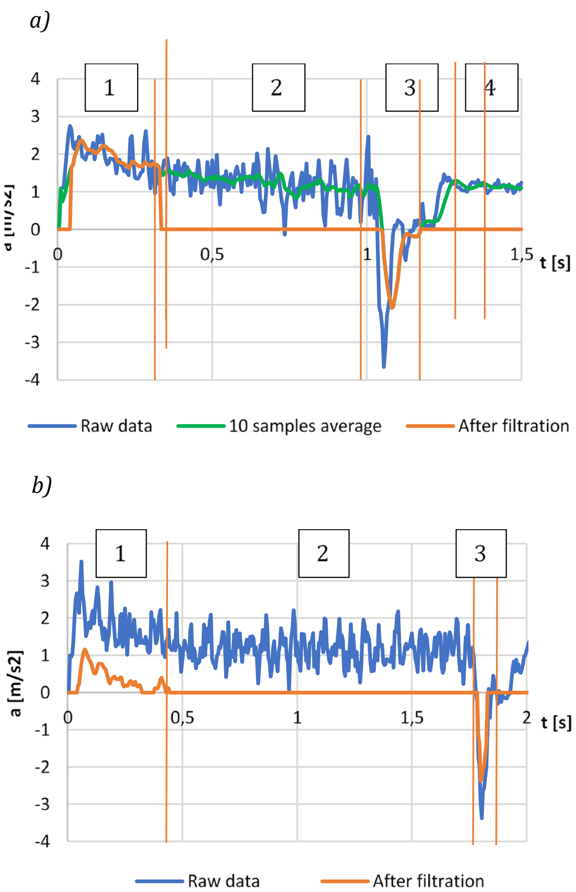

Fig. 7. Implementation of acceleration data filtration: a) course with a zero cut-off factor equal to: 0.18 (1.2 s), b) passage with a factor equal to: 0.24 (2 s)

therefore adjusts the calibration offset. Lack of offset may cause elliptical readings instead of circular thus in some ranges the angle will increase much faster, in others slower. Accelerometer and magnetometer are prone to high noise and external factors such as vibrations, slight tilting. It is therefore necessary to filter the data provided.

Linear acceleration cannot be filtered using Kalman or Complementary Filters that on the other hand can be used in case of angular acceleration. Fig. 7a shows three characteristics. The blue one represents raw data recorded by the accelerometer during 1st driving straight with 0,2s braking. There is a sudden increase in acceleration in the first phase of movement, as a result of powering the engines, overcoming the friction resistance of the wheels against the ground and putting the wheels in motion (Fig. 7a – section 1). At a later stage, its value oscillates in the range of 1-2 m/s^2 to start the braking procedure after a second (Fig. 7a – section 2). Then the acceleration is negative and the robot's movement ends in 1.2-second (Fig. 7a – section 3). In the range of 1.2-1.5 seconds, an acceleration value of approximately 1.4 m/s^2 is visible (Fig. 7a – section 4). This is the zero reference necessary to take into account in the calibration and filtration process due to the geometry of the surface on which the robot is operating. At the beginning, 10 measurement samples are averaged, the table in which these data are stored is each time supplemented with a new acceleration indication and reduced with the oldest. The data filtered at this stage are shown in Fig. 7a in green. The measurement of surface geometry is made by collecting 50 samples of accelerometer indications at a standstill within 8 ms intervals, and then their arithmetic mean is recalculated. The introduction of calibration of the acceleration value at standstill determines the efficiency of the robot displacement calculation. Failure to use calibration would result in excessive acceleration values, and thus erroneous readings when counting the numerical integral.

Known acceleration value at standstill position is set to compensate indications while driving. An additional, experimentally determined zero cut-off value of 0.18 allows to reduce noise by creating a deadband. Thanks to this procedure, acceleration of 0 is clearly noted, and the system is less susceptible to interference (Fig. 7a, orange line). Fig. 7b shows a ride of approximately 2s with a zero cut-off value of 0.24. The blue line characteristic presents the raw data read from the accelerometer and the orange one after filtration. Increasing the value of the zero cutoff factor caused a significant decrease in acceleration compared to the raw data. Reduced values would cause incorrect displacement estimation.

The magnetic meridian direction (magnetic heading) is determined on the basis of the Earth's magnetic field value in two vectors. The magnetometer reads vectors in the X and Y axes. The direction in radians is calculated by calculating the arc tangent of the two variables mentioned above. Magnetometers are very susceptible to the presence of ferromagnetics, which are hidden in the form of pipes arranged in the ground of the room, or other elements present in the laboratory. The global obstacle to the universal use of magnetometers is the need to take into account the diverse position of the Earth's magnetic pole, which clearly does not follow changes in the geographical pole. Magnetometer data is used to orient the robot in a given direction. It is therefore beneficial to keep the indications as real as possible with adequate stability. Averaging

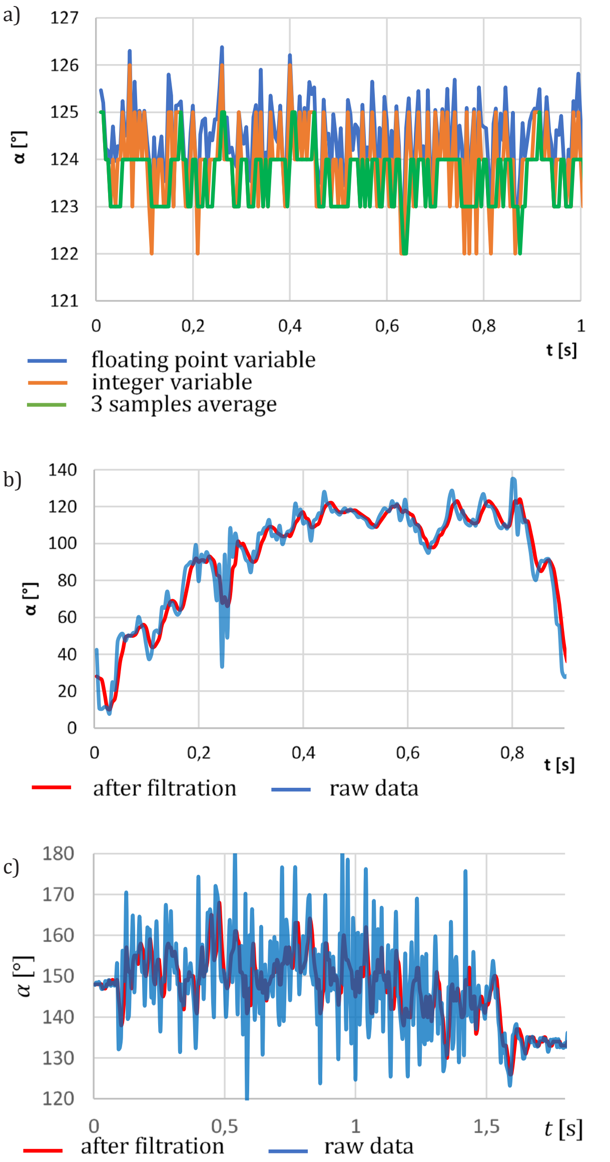

Fig. 8. Measurement of α angle with a magnetometer: a) when the robot is stationary, b) dynamic rotation 10°-120°-10°, c) straight travel – 90 cm

many samples, therefore, becomes fruitless because it would introduce a considerable delay compared to the robot's physical activities. Positioning accuracy does not require decimal or hundredths, because the robot being tested is not able to reach the position with such precision. Initial filtration can therefore be achieved by using a variable to which data from the magnetometer is saved, in integer form, not a floating point (Fig. 8a, blue and orange characteristics). The second stage of filtration is averaging data from three samples and it is a low-pass filter. In this way we get relatively stable indications that affect the robot's operation (Fig. 8a, green course). Fig. 8b shows the course with dynamic rotation of the robot from the indicated position 10° to 120° and back. As can be seen, filtration helps stabilize the noise resulting from the rapid movement of the robot. An example would be smoothing the rapid change of indications in the range of 0.2-0.3 s, and the filtered data are not simultaneously delayed in relation to the original signal. This type of problem could arise if too many averaged samples were used. The second noticeable aspect is that the robot reaches a comparable final angle value with the initial one. Driving over a distance of 90 cm over an uneven surface shows a lot of noise (Fig. 8c). Signal filtration at such indications is very difficult. In addition, the problem of uneven operation of all four engines is illustrated. The initial direction is around 148 °. After traveling 90 cm, the robot is positioned at an angle of about 133°. This gives about 15 cm of discrepancy over a length of less than a meter.

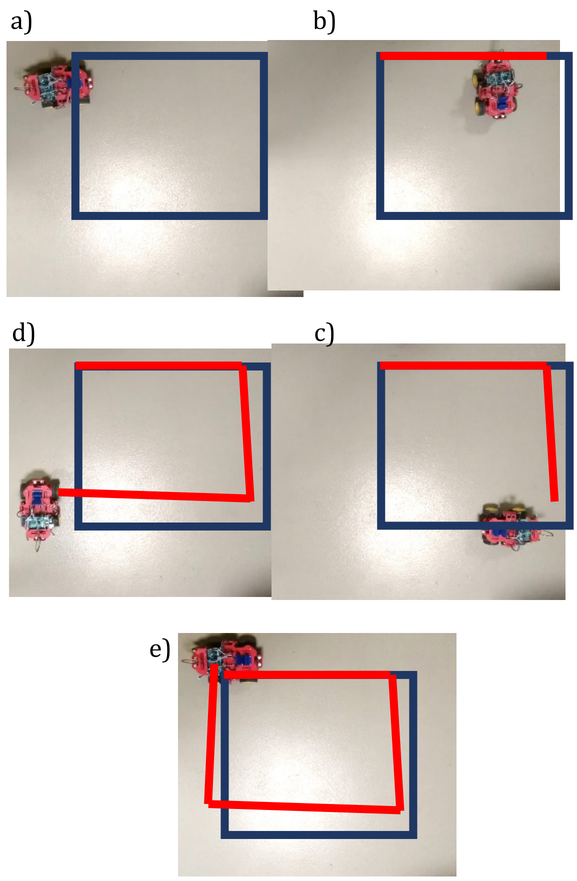

Fig. 9. Implementation of the trajectory of a square with a side length of 50 cm (blue line – the expected shape, red line – the actual shape): a) starting point, b) first 50 cm pass with a 90° turn, c) second 50 cm pass with a 90° turn, d) third pass 50 cm with a 90° turn, e) fourth pass 50 cm with a 90° turn – return to the starting point

Fig. 9 shows the process of implementing a 500 mm square trajectory. The starting point was marked in Fig. 9a. A card with a number was set up in front of the robot for further recognition by the neural network. After the recognition process, the ride begins in front of the specified distance. Then a 90 degree rotation is made (Fig. 9b). These two operations are repeated three more times (Fig. 9c, d), with the robot finishing the journey in the starting position (Fig. 9e).

The shape of the implemented figure is clearly distorted. It is influenced by many factors discussed earlier in this work, with a detailed analysis of specific components. The gradient of the ground has a negative effect on the accelerometer, which calibrates during standstill. Changing it along the robot's motion path causes erroneous readings that are difficult to filter out. The error accumulates due to the necessity of performing the ride and turn four times. Another problem is the stress that occurs when mounting the motors to the frame. Tightly folded elements cause the wheels to lose alignment. Working at constant speed, they can cause the robot to drift to either side as well as error resulting from uneven operation of the motors. Although the Authors compensate it accordingly for each pair of motors (by adjusting the PWM signal fill separately for the motors on the right and left), supplying them with the same voltage does not guarantee repeatability of rotational speeds.

3.2. ANN Recognition

For the task of recognizing the trajectory of the robot's movement, the Authors decided to use a unidirectional neural network. Networks of this type occur in the literature under the name MLP (Multilayer Perceptron), because this network is actually made up of layers called successively the input, hidden and output. The network architecture was shown in Fig. 10.

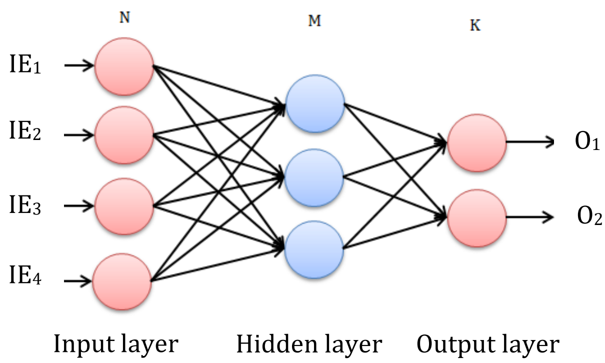

Fig. 10. Artificial neural network architecture [30]

A characteristic feature of this type of ANN is the method of connecting subsequent neurons, which in this case causes that each neuron of the preceding layer is connected to each neuron of the next layer. This method allows the influence of a single neuron on the network input on all neurons of subsequent layers. Such connection causes every neuron at the network input is just as important, which is an undoubted advantage. The disadvantages of this solution include the number needed to calculate and save the weights of the neural network. As we can see from the network architecture, in case of image recognition, a single pixel is the input for the corresponding input layer neuron. For images, i.e. with a resolution of 80x120, the neural network needs 9600 neurons at the input. The Authors of the work decided that numbers from 0 to 9 will be recognized and responsible for planning the appropriate trajectory of the mobile robot. There are 10 such digits, hence the multilayer perceptron needs to work properly with 10 neurons in the output layer. Each neuron of the output is responsible for classifying the corresponding pattern in the form of an image.

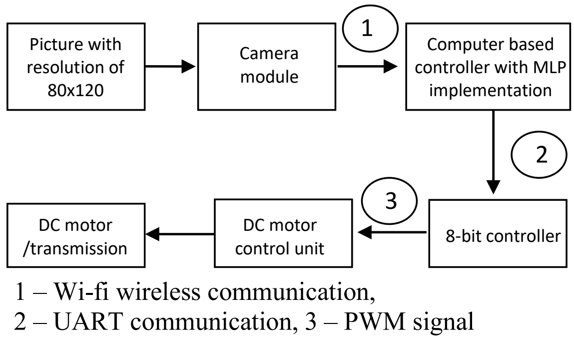

Fig. 11. Blok diagram of system structure

The block diagram in Fig. 11 shows how the planning process of the mobile robot's movement is carried out. The key element in this task is the multilayer perceptron. Due to the relatively large size of the network, and the necessary image analysis, it was decided to use the computer as a calculation unit for the network used. The network that meets the conditions for recognizing digits from 0 to 9 and with a resolution of 80x120 pixels, is a multilayer perceptron with an architecture of 9600-705-10. The number of neurons at the input is determined by the image resolution. The number of neurons at the output results from the number of classified patterns, in our case the digits. However, the number of neurons in the hidden layer was determined on the basis of research, the results of which are presented in chapter 4 of this work. The authors decided to use the sigmoidal function as a function of neuron activation.

4. ANN Examination

Because of artificial neural networks reflected architecture in the operation of human neural cells, they are one of the best tools for solving classification tasks. Classifications of numbers, animals, vegetables and fruits, road signs, faces and many other elements or things are known in the literature. In the work considerations, it was decided to use the neural network to classify the image in the form of a digit, and then, depending on the recognized digit, drive the mobile robot to overcome the appropriate trajectory. The Authors claim that machine vision based on artificial neural network technology is an area that is unrivaled in comparison with other image classification algorithms. The principle of classification and operation of the network, which is referred to at work, i.e. MLP (Multilayer Perceptron), is very similar for all these cases. Multilayer perceptron differ in the number of layers, neurons, activation functions and parameters such as learning coefficient or function parameters determining the shape of the curve. In software design, the most difficult thing is choosing the right learning rate, curve shape and number of neurons in the hidden layer. The most common way of selecting the parameters mentioned above is the test method, relying on existing networks or on own experience in neural network design. The work focuses on architecture networks with one hidden layer. This kind of multilayer perceptron was used due to the fact that in the design of the neural network, we strive not only to reduce the error of the neural network response, but also to the smallest possible number of neurons in the hidden layer. This is because oversizing the number of neurons in this layer leads to an increase in the number of calculations, and thus time to teach MLP. Due to the use of MLP to determine the trajectory of the mobile robot, it was decided to carry out tests to calculate the number of hidden layer neurons, depending on the number of neurons of the input layer and output. The authors did not find in the literature a formula that allows to estimate what number of neurons in the hidden layer allows to complete the task of teaching a network of specific patterns.

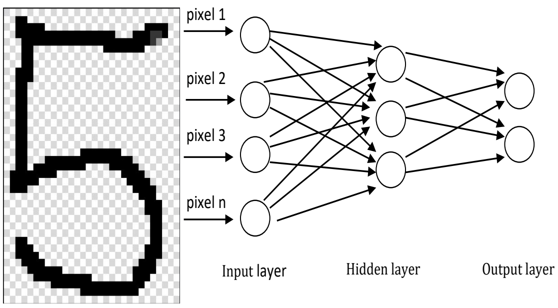

Fig. 12. Architecture of tested networks

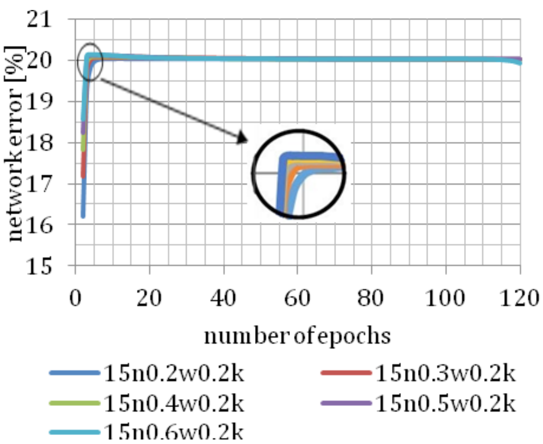

Fig. 13. Architecture 300-15-5, angle parameter of the activation function curve 0.2

The research aims to develop a formula that will determine the number of neurons in the hidden layer. This will allow easy redesign of the neural network in the event of a change in the number of classified patterns that affect the number of output neurons, and in the case of a change in the resolution of the analysed image that affects the number of neurons in the input layer. Images analysed by neural networks usually have low resolutions to limit the time needed for learning. An example is the network trained on the MNIST database, which is a database of 60,000 hand-drawn digits with a resolution of 28x28 pixels, in case of the target network presented in the work this resolution is 120x180 pixels. Due to the fact that the network learning time in case of MLP as shown in the literature example in Fig. 15, for the optimal case is 114 hours for a resolution of 29x29 pixels. Research on networks with a lower resolution than the target 120x180, saved a few hundred or several thousand hours selection of the appropriate number of neurons in the hidden layer. In addition, the research allowed to estimate the value of other parameters, i.e. the shape of the sigmoidal curve as a function of activation, and the learning rate factor. Research was began with small artificial neural networks, while in subsequent iterations the number of neurons in the hidden and input layers were increased by increasing the resolution of the analysed image. Sample results are shown in the following illustrations.

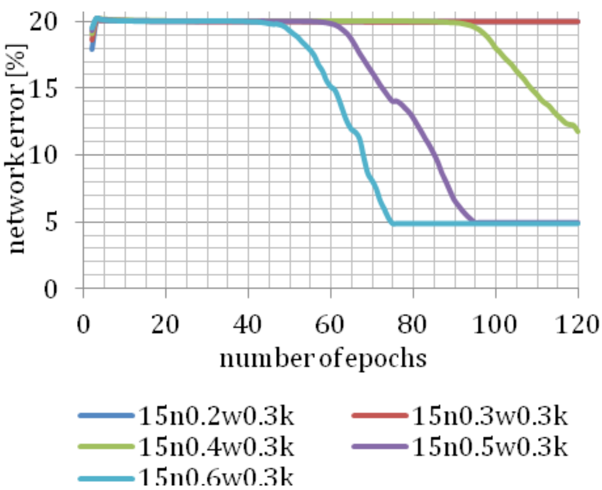

Fig. 14. Architecture 300-15-5, angle parameter of the activation function curve 0.3

Fig. 14 shows to what extent the learning factor affects the learning speed of the artificial neural network. With a learning factor of 0.6, the network learned to the assumed error rate after 75 learning epochs. However, when the value of this coefficient was 0.5, the network already needed 94 learning epochs.

In Fig. 13 to 17 one can observe the learning progress of the neural network with a variable learning coefficient and a variable parameter of the slope of the sigmoidal curve. It was noted that the impact of these parameters is significant. For small values, the artificial neural network did not show learning progress. However, when these values were too large, the network learned chaotically. Analyzing the previous graphs of learning progress, we can see that both the learning rate and the coefficient of the sigmoid curve angle affect the learning speed. Appropriate selection of these parameters allows to optimize the learning process of the artificial neural network.

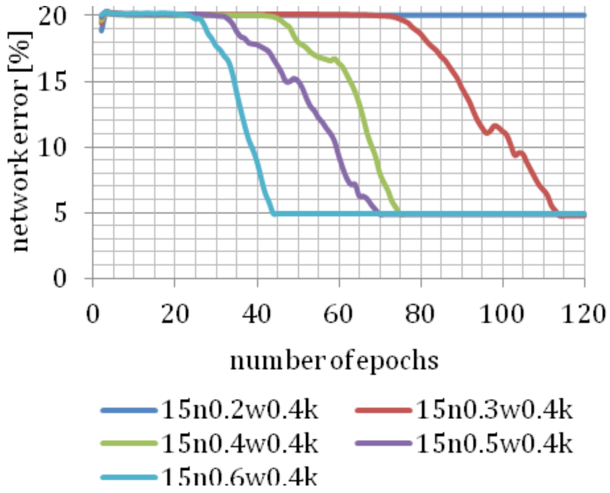

Fig. 15. Architecture 300-15-5, angle parameter of the activation function curve 0.4

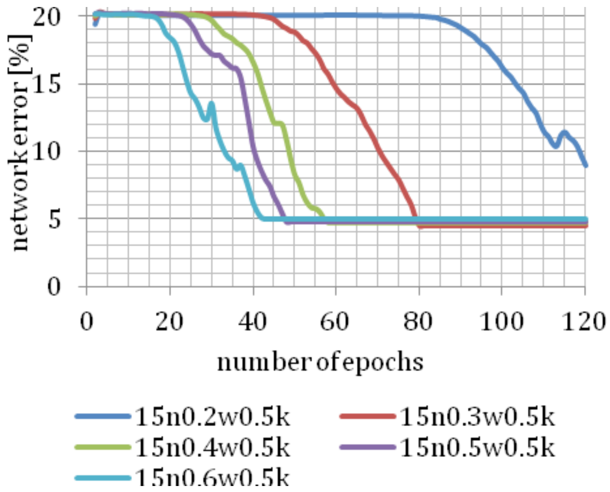

Fig. 16. Architecture 300-15-5, angle parameter of the activation function curve 0.5

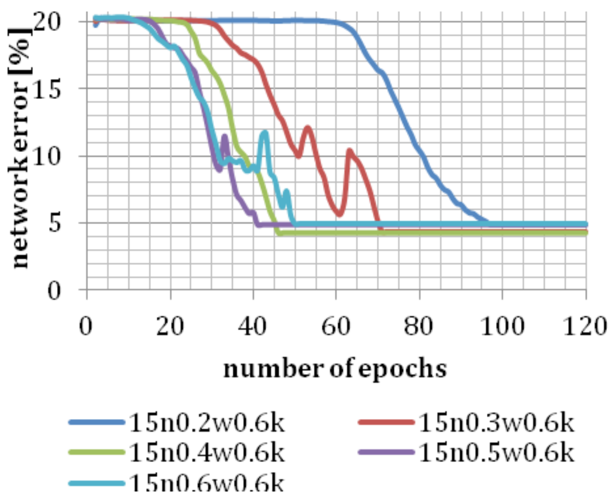

Fig. 17. Architecture 300-15-5, angle parameter of the activation function curve 0.6

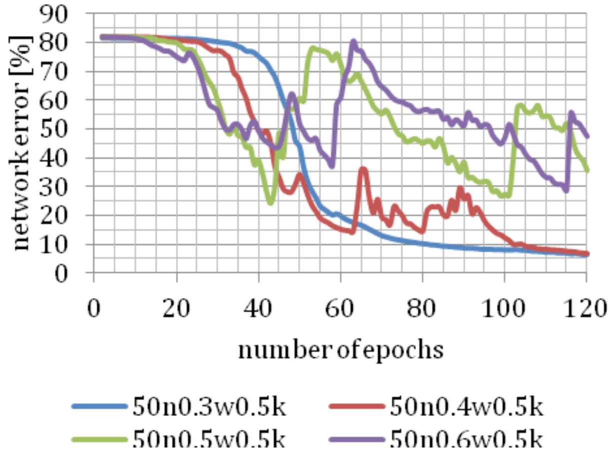

Fig. 18. Architecture 1200-50-5, angle parameter of the activation function curve 0.5

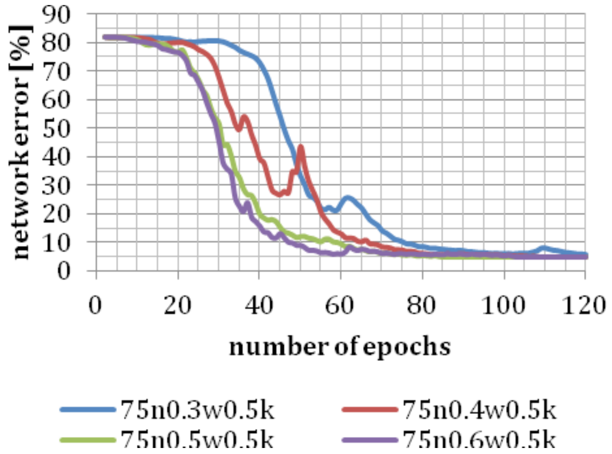

Fig. 19. Architecture 1200-75-5, angle parameter of the activation function curve 0.5

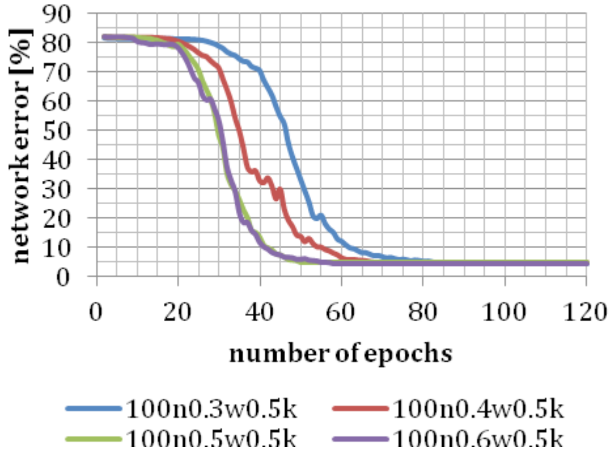

Fig. 20. Architecture 1200-100-5, angle parameter of the activation function curve 0.5

In the first part, Authors have analysed the effect of the learning rate and the slope angle parameter of the sigmoidal curve as a function of the activation of each neurons. In Fig. 18 to 20, however, can be seen how the number of neurons in the hidden layer has an impact on the learning process. During the study, it was also noticed that increasing the number of neurons in the hidden layer caused an increase in learning speed. However, this process was saturated and in case of further increase in the number of neurons in the hidden layer, the multilayer perceptron did not learn faster. Hence, network design is not a simple process.

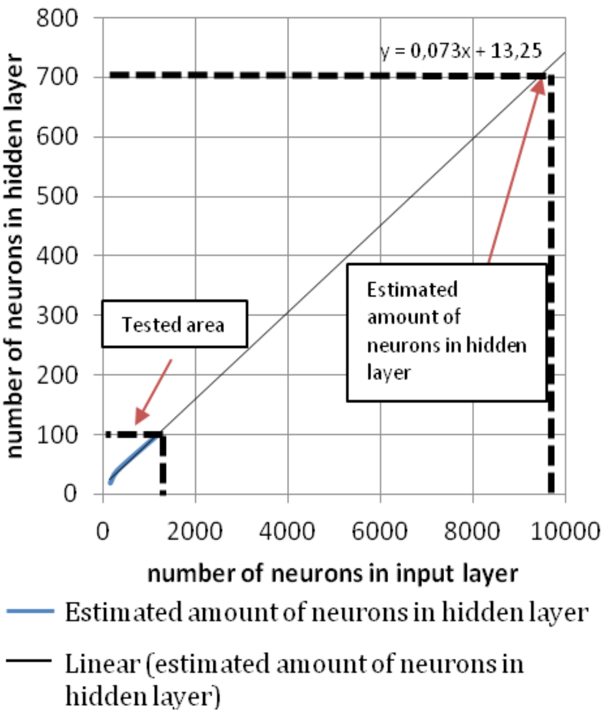

Fig. 21. Trend line graph generated using Excel software for data obtained from networks with architectures with 10 neurons on the output

Basing on performed tests, authors proposed a graph showing the number of hidden layer neurons depending on the examined resolution of the classified image. The results are presented in Fig. 21.

5. Conclusion

Implementation of artificial neural networks in the task of mobile robot navigation as well as machine vision has a crucial value in a way of modern robotics development. Captured image can simultaneously allow robot to avoid obstacle, follow the marker and provide information to navigate. Artificial intelligence methods, including the multilayer perceptron described in the paper, are perfect for this type of tasks.

An example application of the proposed system is an autonomous mobile trolley in warehouses. Robot controller performs scanning of a label placed on a cargo and further proceed an assigned trajectory to deliver it to its destination point.

The designed network allowed to obtain the expected results in terms of path recognition. The algorithm can classify a specific number with certainty above 97% for digits written by a person whose handwriting has been included in the database of learning network. In order to recognize the letter better, it would be necessary to supplement it with samples of a larger number of people.

The use of low cost electronic components for relatively precise robot movement over short distances was a big challenge. For the sensors used, the Authors observed a large discrepancy in the raw results. The proposed filtration methods allowed to obtain satisfactory results. The main sources of interference are the dynamic movements of the entire robot platform and the fact that the sensors are rigidly attached to the structure. Vibrations generated by engines even during standstill meant that the data obtained are affected by errors. Nevertheless, the analysis of accelerometer data while driving allowed defining the nature of the indications depending on the state of the robot (acceleration, driving at a constant speed, braking). To improve the precision of trajectory implementation, it would be necessary to equip the robot in parts with greater accuracy, which would be associated with a higher price for the device.

ACKNOWLEDGEMENTS

The work was supported by the grant of Polish Ministry of Science and Higher Education no. 0614/SBAD/1501.

AUTHORS

Marcin Białek* – Department of Mechatronic Devices, Poznan University of Technology, Poznan, Poland, e-mail: marcin.r.bialek@doctorate.put.poznan.pl.

Patryk Nowak* – Department of Mechatronic Devices, Poznan University of Technology, Poznan, Poland, e-mail: patryk.rob.nowak@doctorate.put.poznan.pl.

Dominik Rybarczyk – Department of Mechatronic Devices, Poznan University of Technology, Poznan, Poland, e-mail: dominik.rybarczyk@put.poznan.pl.

*Corresponding author

REFERENCES

[1] M. Garbacz, "Planowanie ścieżki dla robota mobilnego na podstawie informacji z czujników odległościowych", *Automatyka / Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie*, vol. 10, no. 3, 2006, 135–141.

[2] K. Bhagat, S. Deshmukh, S. Dhonde, S. Ghag and V. Waghmare, "Obstacle Avoidance Robot", *International Journal of Science, Engineering and Technology Research*, vol. 5, no. 2, 2016, 439–442.

[3] C. Randell and H. Muller, "Low Cost Indoor Positioning System". In: G. D. Abowd, B. Brumitt and S. Shafer (eds.), *Ubicomp 2001: Ubiquitous Computing*, 2001, 42–48, DOI: 10.1007/3-540-45427-6_5.

[4] Z. Tan, S. Bi, H. Wang and Z. Wang, "Target Tracking Control of Mobile Robot Based on Ultrasonic Sensor". In: *Proceedings of the 6th Interna-*

tional Conference on Information Engineering for Mechanics and Materials, 2016, 60–64, DOI: 10.2991/icimm-16.2016.13.

[5] S. Adarsh, S. M. Kaleemuddin, D. Bose and K. I. Ramachandran, "Performance comparison of Infrared and Ultrasonic sensors for obstacles of different materials in vehicle/ robot navigation applications", *IOP Conference Series: Materials Science and Engineering*, vol. 149, 2016, DOI: 10.1088/1757-899X/149/1/012141.

[6] J. M. Soares, I. Navarro and A. Martinoli, "The Khepera IV Mobile Robot: Performance Evaluation, Sensory Data and Software Toolbox". In: L. P. Reis, A. P. Moreira, P. U. Lima, L. Montano and V. Muñoz-Martinez (eds.), *Robot 2015: Second Iberian Robotics Conference*, vol. 417, 2016, 767–781, DOI: 10.1007/978-3-319-27146-0_59.

[7] M. Januszka, M. Adamczyk and W. Moczulski, "Nieholonomiczny autonomiczny robot mobilny do inspekcji obiektów technicznych", *Prace Naukowe Politechniki Warszawskiej. Elektronika*, vol. 166, no. 1, 2008, 143–152.

[8] J. B.-Y. Tsui, *Fundamentals of Global Positioning System Receivers: A Software Approach*, John Wiley & Sons, Inc., 2004.

[9] A. Wąsik, R. Ventura, J. N. Pereira, P. U. Lima and A. Martinoli, "Lidar-Based Relative Position Estimation and Tracking for Multi-robot Systems". In: L. P. Reis, A. P. Moreira, P. U. Lima, L. Montano and V. Muñoz-Martinez (eds.), *Robot 2015: Second Iberian Robotics Conference*, vol. 417, 2016, 03–16, DOI: 10.1007/978-3-319-27146-0_1.

[10] Z. Huang, J. Zhu, L. Yang, B. Xue, J. Wu and Z. Zhao, "Accurate 3-D Position and Orientation Method for Indoor Mobile Robot Navigation Based on Photoelectric Scanning", *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 9, 2015, 2518–2529, DOI: 10.1109/TIM.2015.2415031.

[11] T. Więk, "Laserowy system nawigacji platformy mobilnej na przykładzie skanera NAV300", *Pomiary Automatyka Robotyka*, vol. 15, no. 2, 2011, 374–381.

[12] F. B. P. Malavazi, R. Guyonneau, J.-B. Fasquel, S. Lagrange and F. Mercier, "LiDAR-only based navigation algorithm for an autonomous agricultural robot", *Computers and Electronics in Agriculture*, vol. 154, 2018, 71–79, DOI: 10.1016/j.compag.2018.08.034.

[13] B. Siemiątkowska, "Hybrydowa reprezentacja otoczenia robota mobilnego", *Pomiary Automatyka Robotyka*, vol. 11, no. 2, 2007.

[14] D. S. O. Correa, D. F. Sciotti, M. G. Prado, D. O. Sales, D. F. Wolf and F. S. Osorio, "Mobile Robots Navigation in Indoor Environments Using Kinect Sensor". In: *2012 Second Brazilian Conference on Critical Embedded Systems*, 2012, 36–41, DOI: 10.1109/CBSEC.2012.18.

[15] P. Fankhauser, M. Bloesch, D. Rodriguez, R. Kaestner, M. Hutter and R. Siegwart, "Kinect v2 for mobile robot navigation: Evaluation and modeling". In: *2015 International Conference on Advanced Robotics (ICAR)*, 2015, 388–394, DOI: 10.1109/ICAR.2015.7251485.

[16] A. Oliver, S. Kang, B. C. Wünsche and B. MacDonald, "Using the Kinect as a navigation sensor for mobile robotics". In: *Proceedings of the 27th Conference on Image and Vision Computing New Zealand*, 2012, 509–514, DOI: 10.1145/2425836.2425932.

[17] T. Pire, T. Fischer, J. Civera, P. De Cristoforis and J. J. Berlles, "Stereo parallel tracking and mapping for robot localization". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, 1373–1378, DOI: 10.1109/IROS.2015.7353546.

[18] K. Kaemarungsi and P. Krishnamurthy, "Modeling of indoor positioning systems based on location fingerprinting". In: *IEEE INFOCOM 2004*, vol. 2, 2004, 1012–1022, DOI: 10.1109/INFCOM.2004.1356988.

[19] C. Feng, W. S. A. Au, S. Valaee and Z. Tan, "Received-Signal-Strength-Based Indoor Positioning Using Compressive Sensing", *IEEE Transactions on Mobile Computing*, vol. 11, no. 12, 2012, 1983–1993, DOI: 10.1109/TMC.2011.216.

[20] Ł. Błaszczyk, "Podstawy Teorii Oszczędnego Próbkowania (Compressed Sensing – Theoretical Preliminaries)", B.S. thesis, Faculty of Mathematics and Information Science, Warsaw University of Technology, 2014 (in Polish).

[21] W. Kang, S. Nam, Y. Han and S. Lee, "Improved heading estimation for smartphone-based indoor positioning systems". In: *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications – (PIMRC)*, 2012, 2449–2453, DOI: 10.1109/PIMRC.2012.6362768.

[22] B. Muset and S. Emerich, "Distance Measuring using Accelerometer and Gyroscope Sensors", *Carpathian Journal of Electronic and Computer Engineering*, vol. 5, no. 1, 2012, 83–86.

[23] M. E. Qazizada and E. Pivarčiová, "Mobile Robot Controlling Possibilities of Inertial Navigation System", *Procedia Engineering*, vol. 149, 2016, 404–413, DOI: 10.1016/j.proeng.2016.06.685.

[24] Y. Pei and L. Kleeman, "Mobile robot floor classification using motor current and accelerometer measurements". In: *2016 IEEE 14th International Workshop on Advanced Motion Control (AMC)*, 2016, 545–552, DOI: 10.1109/AMC.2016.7496407.

[25] K. S. Younis and A. A. Alkhateeb, "A New Implementation of Deep Neural Networks for Optical Character Recognition and Face Recognition". In: *Proceedings of the new trends in information technology*, 2017, 157–162.

[26] D. C. Cireşan, U. Meier, L. M. Gambardella and J. Schmidhuber, "Deep Big Multilayer Perceptrons for Digit Recognition". In: G. Montavon, G. B. Orr and K.-R. Müller (eds.), *Neural Networks: Tricks of the Trade*, vol. 7700, 2012, 581–598,
DOI: 10.1007/978-3-642-35289-8_31.

[27] M. Dobrowolski, M. Dobrowolski and P. Kopniak, "Analiza możliwości wykorzystania czujników urządzeń mobilnych pod kontrolą zmodyfikowanych systemów operacyjnych (Analysis of the use of sensors in mobile devices with modified operating systems)", *Journal of Computer Sciences Institute*, vol. 5, 2017, 193-199 (in Polish).

[28] K. Seifert and O. Camacho, *Implementing Positioning Algorithms Using Accelerometers*, Application Note AN3397, Freescale Semiconductor, 2007.

[29] "jarzebski/Arduino-HMC5883L: HMC5883L Triple Axis Digital Compass Arduino Library". https://github.com/jarzebski/Arduino-HMC5883L. Accessed on: 2020-05-28.

[30] https://petrospsyllos.com/images/ssn-kurs-2/Obraz5.png. Accessed on: 17.06.2020.