Romanian Journal of Petroleum & Gas Technology
VOL. IV (LXXV) • No. 2/2023

SOFTWARE IMPLEMENTATION OF THE SPIDER4LEGS MOBILE ROBOT

Constantin Daniel Stanciu [1]

Mihai Vlagea [1]

Gabriela Bucur [1]

[1] Petroleum-Gas University of Ploiesti, Romania
e-mail: gbucur@upg-ploiesti.ro

DOI: 10.51865/JPGT.2023.02.01

ABSTRACT

The objective of this paper is to develop the control system for driving a mobile robot using the Arduino board, controlled by an ESP 32 logic processor, named SPIDER4LEGS. In order to be used in the oil and gas industry, for example for the internal inspection of oil product transport pipelines, the four-legged spider type robot version was chosen, especially due to the special mobility it offers. The created control system must ensure the control of the movement of the kinematic couplings in the construction of the legs, the monitoring and maintenance of stability as well as the establishment of the optimal stepping sequences. The control system is simple and it use the Wi-Fi communication, because it is the best way to transfer data between the robot and the operator, offering a lot of advantages.

Keywords: mobile robot, servomotor, ESP 32

INTRODUCTION

The mobile robot is an automatic, complex system that can perform different activities in a variety of real-world situations. It is a combination of devices equipped with servomotors and sensors under the control of a hierarchical computing system, operating in a real space, marked by a series of physical properties and planning movements so that they can perform a task according to the initial state of the system and depending on the existing a priori information, related to the work environment. In a very broad and consequently comprehensive sense, a system represents a set of elements that interact with each other and with the environment in order to achieve a certain goal [1]. Mobility allows the robot much greater flexibility in the execution of complex tasks.

The specific problems that appear in mobile robots are the following: avoiding impact with stationary or moving objects, determining the position and orientation of the robot on the ground, as well as planning an optimal movement trajectory. In the case of a distributed automatic robotic system, the spatial positions are of extreme importance and the fulfilment of the desired goals as well as the functioning of the entire system depends on them. The robot must be able to plan its movements, automatically decide which movements to perform to complete a task, depending on the current arrangement of objects in the workspace. Movement planning does not consist of a single and well-defined problem, but of a set of problems, some of which are more or less variants of the others. Avoiding collisions with fixed or moving obstacles, located in the robot's workspace can be done by several methods: the creation of a mechanical guard that stops the robot by deformation, the use of sensors that measure the distance to obstacles in the direction of movement, the use of proximity sensors, the use of correlated information from more many types of sensors. Locating objects can also be done by physical contact, but this procedure imposes restrictions on the speed of movement. The physical contact between the robot and the obstacles in the environment generates reaction forces that change the state of the robot. High working speeds make the dynamic effects of a physical contact with obstacles or handled objects risky and can lead to damage to the object or the robot.

Among the more frequently used navigation methods can be mentioned: measuring the number of rotations made by the motor wheels, using accelerators and gyroscopes, electromagnetic buoys installed in the field, optical or magnetic passive or semi-passive beacons. Information about the workspace can be obtained independently of any action of the robot and can be organized on navigation maps. The map provides a configuration of the initial structure of the workspace [1].

The robot must operate safely, for example it must avoid obstacles or stay away from dangerous operating conditions and not endanger the operator in the vicinity of the robot [2]. The necessary configuration is obtained by updating the initial map with information obtained from the navigation system of the autonomous robot. Based on the initial model, a possible trajectory can be established to achieve the goal, a trajectory that is as close to the real one as the information about the working environment is closer to the reality on the ground. The initially existing information can contribute to a division of the space into accessible and forbidden areas. In the process of modelling the workspace, it is important to take into account the dimensions and the physical-mechanical possibilities of approaching these obstacles by the robot, depending on their dimensions [2].

One field of application of the pasting robots is the internal inspection of pipelines. But these systems have problems with going over obstacles or moving on inclined surfaces. One such robot is the one built by Siemens in 1995, the Pipe Climbing Robot, with 8 legs. Another example is the MORIZ robot, with 8 legs, developed by the Technical University of Munich, capable of climbing the inner surface of pipes with different slopes.

Wired communication it is the simplest way to transfer data between the robot and the operator. A series of problems associated with wireless transmission disappear in the case of wired data communication, at the cost of restricting the robot's mobility. The main advantage of wired communication systems is that, in addition to the data transmitted between the robot and the human operator, it can also be supplied with electricity, thus eliminating the need for a battery on the robot and increasing the autonomy of the robot [2]. *Wireless communication systems* they use a series of technologies for transmitting data through the air (eng. wireless), among which we mention: transmission through waves in the infrared spectrum, bluetooth, radio modems, WiFi (eng. wireless fidelity) and others [2].

SPIDER4LEGS MOBILE ROBOT STRUCTURE

The components for the robot are the following: the main body with 4 legs; ESP32 development board, Wi-Fi and Bluetooth BLE, Dual Core; PCA9685 module, I2C interface, 16 CH, servo motor; 12 Servo motors SG90 180 degrees. The mechanical design of the robot and the implementation of the actuation system are presented in Figure 1 and Figure 2.

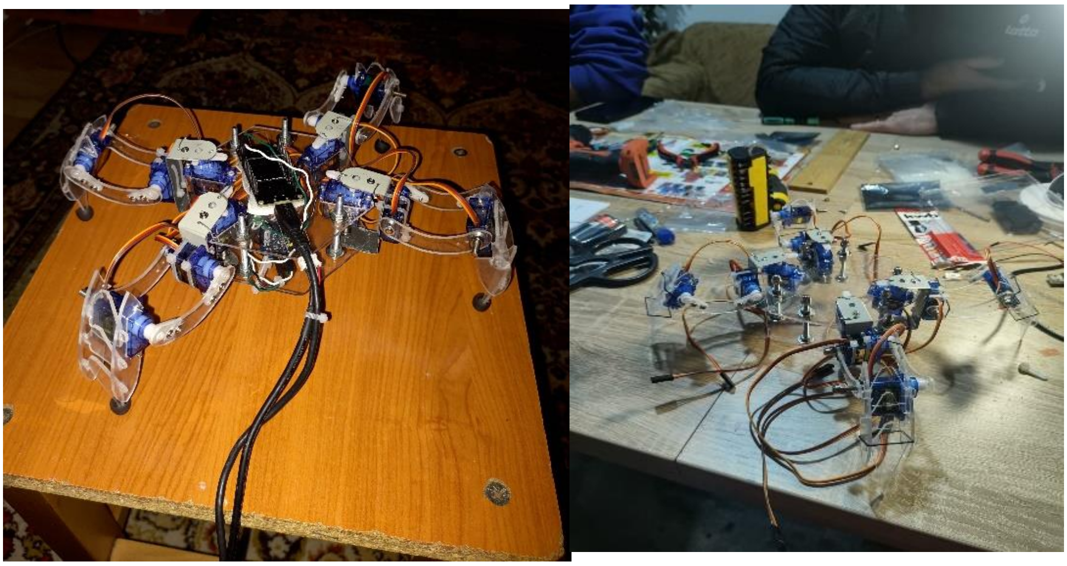

***Figure 1**. SPIDER4LEGS mobile robot*

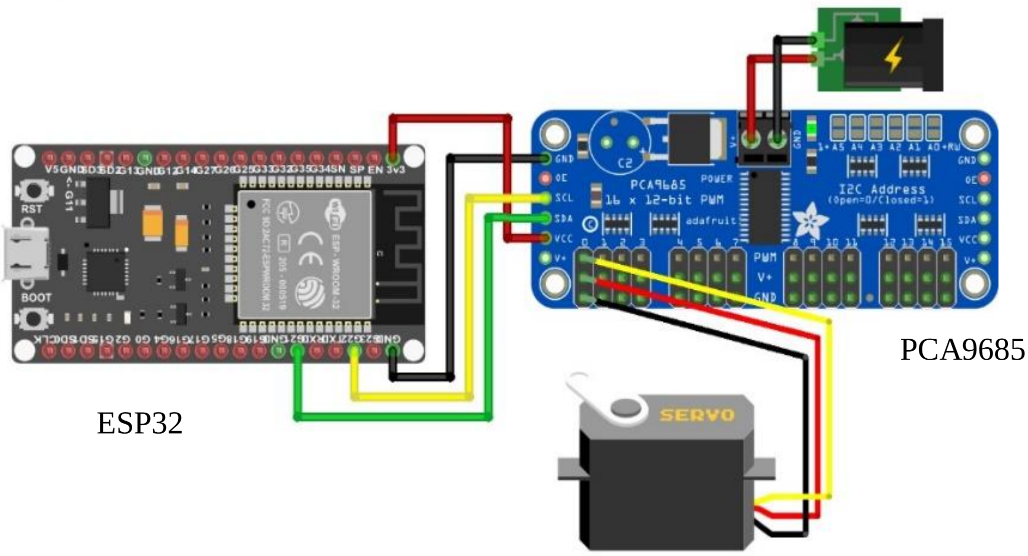

***Figure 2**. Electrical connection diagram*

SOFTWARE DEVELOPMENT FOR ROBOT CONTROL

Developing software for this robot means, in fact, creating the program for the microcontroller. So, all the servo motors are controlled using Pulse with Modulation (PWM) received from the microcontroller. The direction and position of the servos can be controlled by varying the PWM signal. For that objective, each servomotor is connected to the servo driver board PCA9685. For this application, the values of the PWM signal was calculated in the design part, which was presented in another work.

The difficulty of controlling a quadruped mobile robot consists in ensuring static stability while walking. *That's why we should always have three feet in contact with the ground*, but its center of mass should always be kept inside the support polygon. Moreover, four is the minimum number of legs for a robot, if no additional dynamic balancing elements are used [7].

Figure 3 shows the software development logic for controlling the four-legged robot. After initializing the program, if there is a command from the user, the Wi-Fi module will send a command for the microcontroller board ESP32. Then, the board will send a signal to the PCA9685 servo driver board. After that, the servo driver board will send the signal to the involved servos to make that movement and it will run in loop.

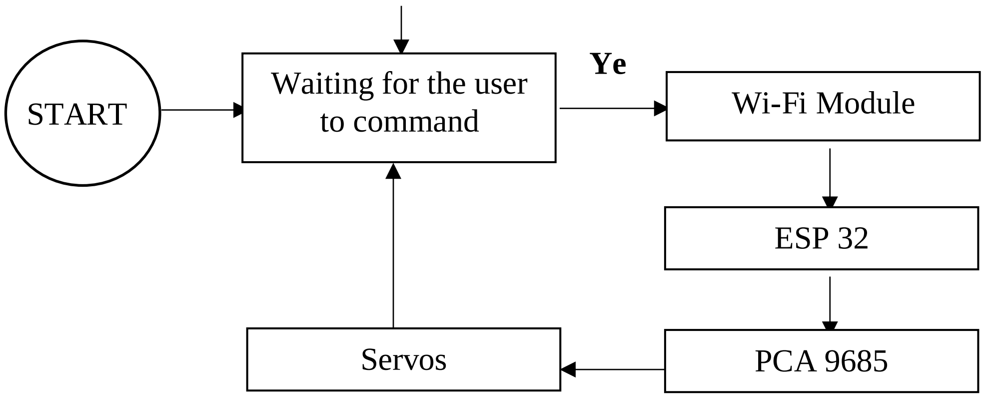

Figure 3. *Schematic for the software process*

In order to control the robot, we have to establish the method of its movement. Walking is a sequence of movements of the legs in space and time. In quantitative terms, it can be defined as follows: a) each leg goes through two repetitive phases, support (contact with the ground) and transfer; b) the duration of a movement cycle (support and transfer) is the same for all four legs. Within the same cycle, the relative duration of the support and transfer phases may vary; c) during a cycle of revolution, the movements of the four legs follow each other in the order of their entry into action; d) the movement is diagonal, this means in a half cycle of revolution, two diagonal legs succeed each other [7].

The control strategy has two hierarchical levels, an upper level that ensures the control of the cooperation of the arms in order to carry out the desired movement, on the trajectory, of the object-task, and a lower level that determines the individual movement of each leg.

The logical schemes of the functions will be presented in the following.

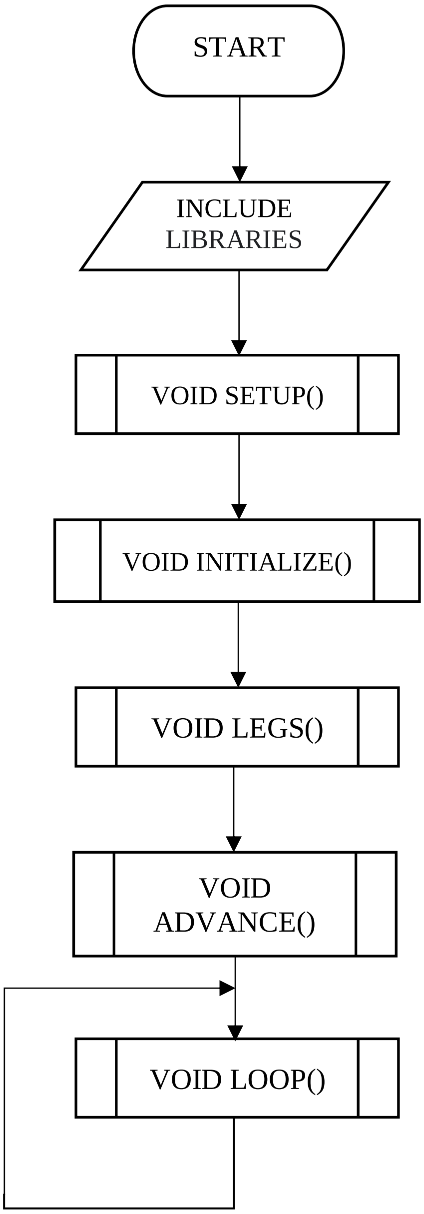

***Figure 4**. Main logic diagram*

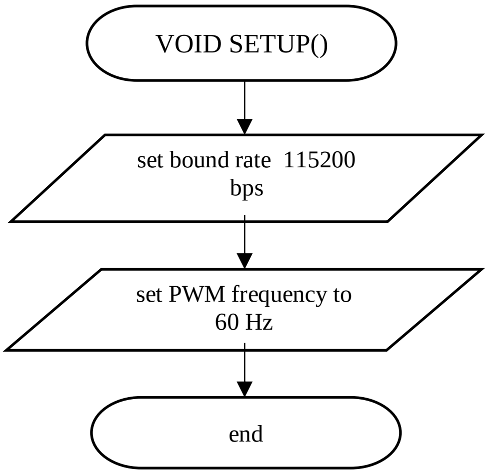

***Figure 5**. Logic scheme of setup function*

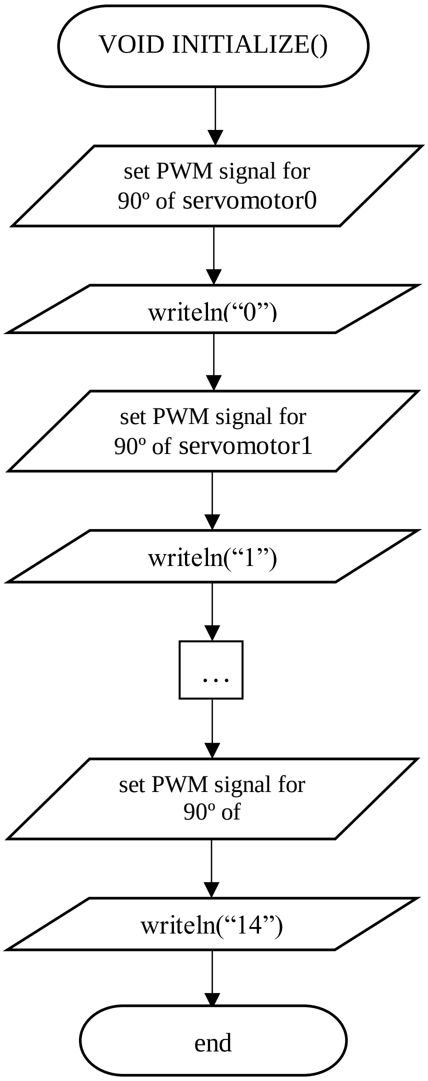

***Figure 6**. Logic scheme of initialize function*

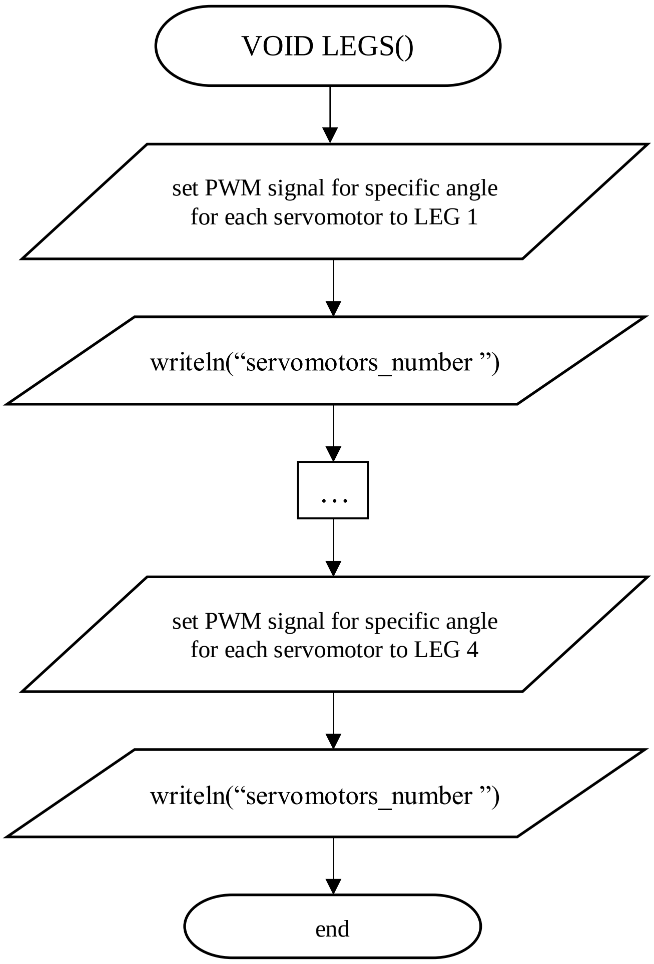

Figure 7*. Logic scheme of legs function*

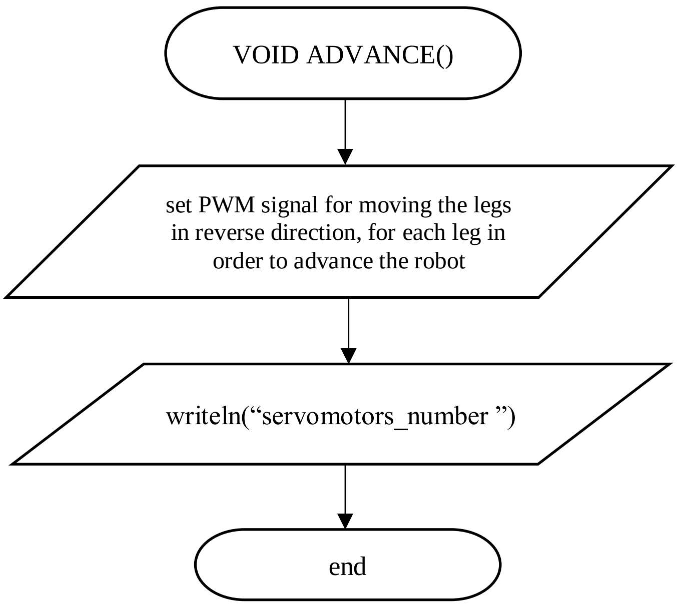

Figure 8*. Logic scheme of advanced function*

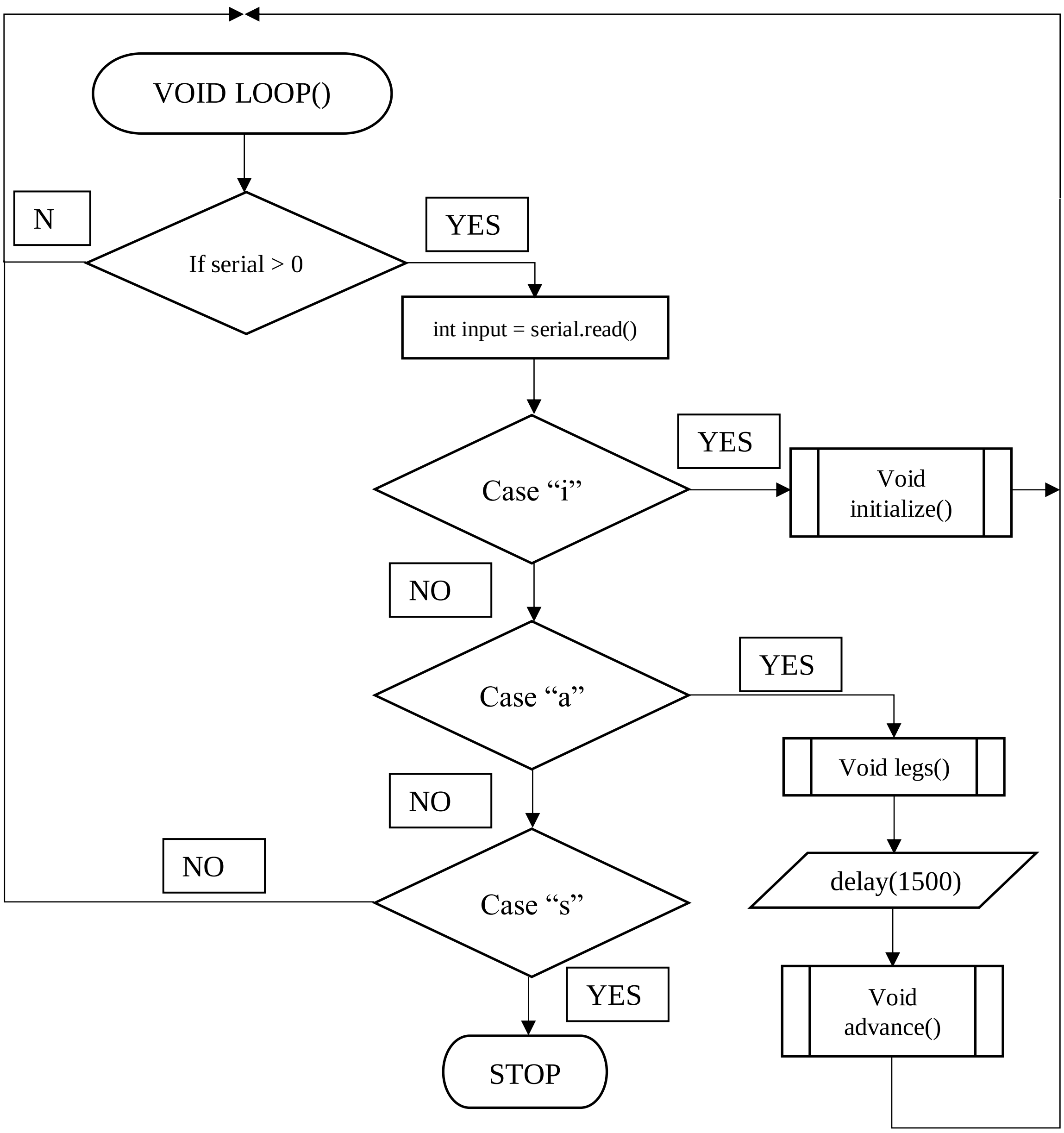

***Figure 9**. Logic scheme of loop function*

The gait chosen for this robot is a simple one, so that we always have three feet in contact with the ground. For forward movement, rotation and backward movement, the sequence of movements is as follows: front leg-left, rear leg-right, front leg-right, rear leg-left.

The actuators are connected to the PCA board through a serial connection. Also, a serial protocol is used to connect PCA9685 servo board and ESP32 board.

CONCLUSIONS

The organization of the driving programs represents a very important aspect in the driving systems of the robots, which greatly depends on the adopted performances. Programs can be created either based on the sequential exploitation of the component blocks within a fixed structure, or through variable programming structures, in this case the transition from one command to another is carried out in an adaptive manner, based on the latest results obtained from the space operation of the robot.

The complexity of the driving system and the degree of difficulty of the performed operations determine the adoption of specific technologies for the implementation of driving laws. Conventional driving methods are based on a proper modeling of the system.

In contrast to conventional techniques, fuzzy management methods offer much more pragmatic solutions, with obvious application facilities in various fields, requiring an acceptable level of knowledge in the field of systems, control logic and technology. The use of fuzzy logic in the implementation of control systems in the absence of total information on the system was also one of the factors that determined the wide spread of fuzzy control in the most diverse fields.

Another method of approaching the unconventional control of robots is formed by neural networks. Recently, fuzzy logic and neural networks form neuro-fuzzy driving systems.

A possible field of application of these robots is the internal inspection of pipelines transporting petroleum products. Another application can be an inspection task on an offshore gas platform.

As future trends, the program can be improved by creating another step structure to design the movement of the mobile robot.

REFERENCES

[1] Bucur, G., Linii flexibile și roboti, Editura UPG Ploieşti, 2021, pp 142-152.

[2] Popescu, C., Conducerea neuro-fuzzy a roboţilor mobili, Editura UPG Ploieşti, 2015

[3] https://www.proconsilgrup.ro/servomotor-tipuri-si-principiul-de-functionare

[4] https://www.youtube.com/watch?v=LXURLvga8bQ&t=302s

[5] https://cdn.instructables.com/ORIG/F0L/Y9R2/ITS40PI2/F0LY9R2ITS40PI2.pdf

[6] https://www.youtube.com/watch?v=JFdXB8Za5Os&t=213s

[7] https://pdfcoffee.com/roboti-pasitori-curs--pdf-free.html

[8] https://www.evansville.edu/majors/computerengineering/downloads/projects-2019/spider-robot-report.pdf

Received: April 2023; Revised: May 2023; Accepted: July 2023; Published: July 2023