Acta Mechanica Slovaca **27 (1): 66 - 70,** March **2023**
https://doi.org/10.21496/ams.2022.020

Acta Mechanica Slovaca
ISSN 1335-2393
www.actamechanica.sk

Improving the Simulation of a Mobile Robot by Imitations of Ultrasonic Sensors

Maksym Grytsiv [1,*], Marek Sukop [1], Martin Kočan [1], Dávid Kovaľuk [1], Dušan Mlinarček [1]

[1] Technical University of Košice, Faculty of Mechanical Engineering, Letná 9, 042 00 Košice, Slovakia

Abstract: This article presents imitating the operation of an ultrasonic sensor in a 3D simulation of the Flowcode program to improve the performance and localization of a mobile robot. The main task will be to compare different methods of measuring distance using a simulation model of the sensor and determine the most profitable placement of these sensors to ensure trouble-free passage of obstacles without collisions.

Keywords: mobile robot; simulation; movement; sensor; localization

1. Introduction

With the beginning of the practice of distance learning, there was a great need to provide the opportunity to teach and test their programs online on a computer without the use of real robots, both for students and employees, and teachers. With the software Flowcode [15], such an opportunity appears. Flowcode is a perfect flowchart language based on graphical programming IDE [13]. In this publication, we will consider the possibilities of using sensors in a 3D model of the movement of a mobile robot in various ways. Mobile robots have started to appear more and more in most companies and industries [11]. From warehouse robots used to load, unload, and move goods from arrival to storage, robots in various forms are becoming more and more useful in everyday life, such as kitchen robots, vacuum cleaners, and lawn mowers [1]. Mobile robots performing targeted tasks in an aggressive environment are currently widely used in industry, anti-terrorist operations, elimination of the consequences of technological disasters, the military sphere, etc. [4-6]. Therefore, this kind of simulation allows you to touch more on the number of areas of activity in which it can be used. When performing navigation tasks, robotic systems, such as autonomous vehicles, take advantage of features that include modelling the environment and locating the system in the environment, controlling movement, detecting, and avoiding obstacles, and moving in dynamic contexts ranging from simple to extremely complex. environments. The four common problems of navigation are perception, localization, motion control, and trajectory planning [7-9].

2. Program with built-in sensors

Robots perceive the world with sensors such as cameras, laser rangers, and RGB-D sensors, which allow them to determine the location and shape of objects in their environment [10]. This kind of robot uses Infrared Ray sensors to find the path and direction [1] Three sensors were combined to ensure basic needs of the mobile robot when avoiding obstacles. The first and main sensor is the Ultrasonic Sensor (Fig. 1), which will measure the distance to the target in front. The next two IR sensors (Fig. 1) will be placed at the front corners of the robot, angled to prevent side impact.

*** Corresponding author:** Maksym Grytsiv, ***E-mail address:*** *maksym.grytsiv@tuke.sk*

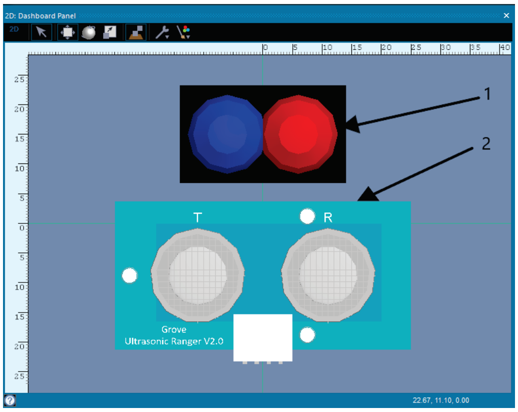

Figure 1: Using Sensors in Flowcode, (1) IR sensor, (2) Ultrasonic Sensor.

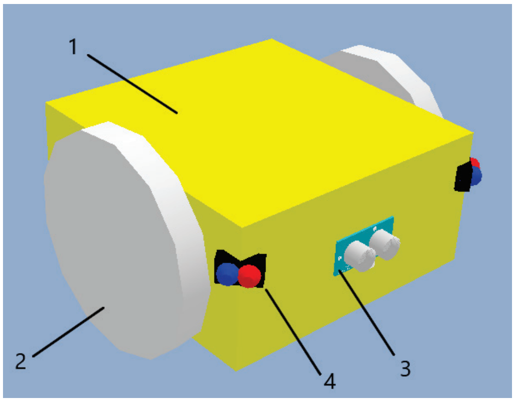

Figure 2: Mobile robot in 3D panel, (1) Body, (2) Wheel, (3) Ultrasonic Sensor, (4) IR sensor.

These sensors are placed on the robot only approximately, in the area of possible attachment, and do not affect the measurement result, making only the visual part. (Fig.2).

To begin with, we took the basis for the reproduction of the simulation in the 3D Flowcode panel from my previous work [3] and then replaced the software for controlling the robot's motion and sensors. Each IR sensor is needed to do initialization.

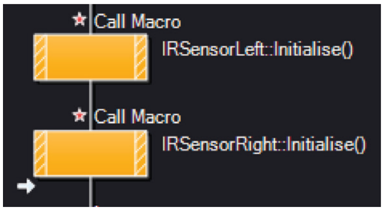

Figure 3: Initialization IR sensors.

After that, let's set up the receiving and sending of a signal from the Ultrasonic sensor and get data. So first, we get them in the millisecond's format - the time delay from the sensor to the object, and then we translate this data using the component block into a distance in millimetres (Fig.4). It is also possible to add component blocks such as "Wait for Echo" and "Ping" (Fig.5). Their main task is to stabilize the signal, remove delays and wait for an echo. But in practice, these blocks create even greater latency, which is incompatible with our motion simulation conditions. Therefore, these blocks were not used.

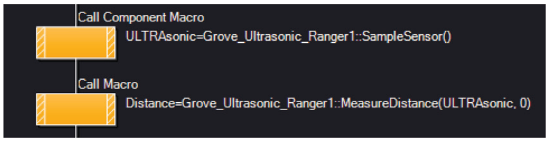

Figure 4: Ultrasonic component blocks.

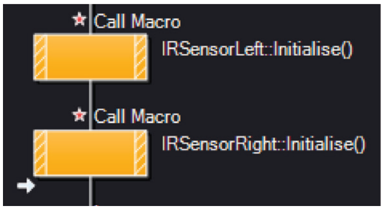

Figure 5: None used components «Ping» and «Wait for Echo».

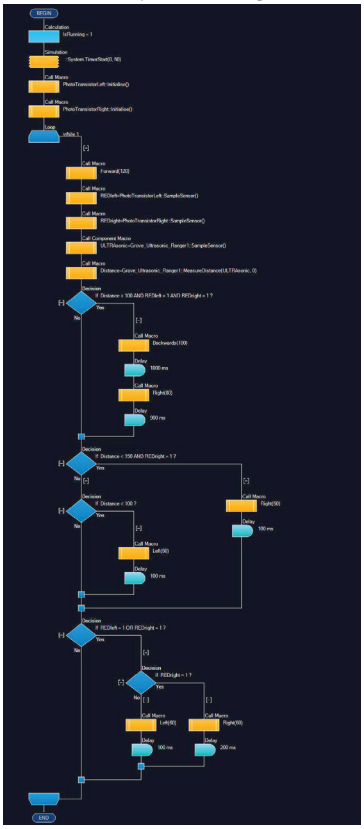

Figure 6: Main Flowchart.

The next part of the Flowcode program is variable and the ability to add new conditions. It serves as the main logic of the robot's movement using the input parameters of the sensors. The conditions are made in such a way as to avoid possible situations of a collision or stuck the robot in a dead end and to minimize contact with the environment. (Fig.6) shows the entire main Flowchart.

We built a territory simulating an interior room (with all kinds of obstacles) to test the performance of the sensors and the entire program (Fig.7). The distance measurement results of the ultrasonic sensor will be shown in the next chapter.

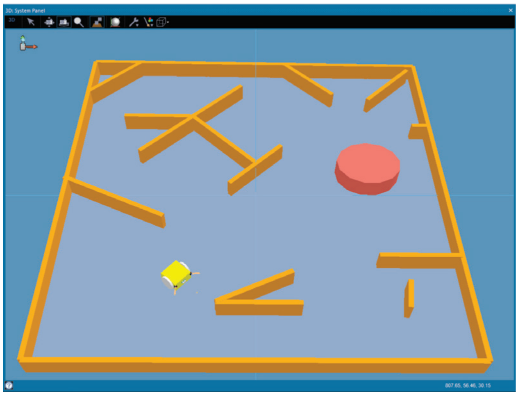

Figure 7: Imitation of an interior room.

3. Program with imitation sensors

The robot's body was created similarly as in the previous chapter, and the sensors were simulated by objects visually similar in shape to real devices. The main distinguishing feature is the independent writing of the functionality of their work. In this case, each was an imitation of an ultrasonic sensor (Fig.8).

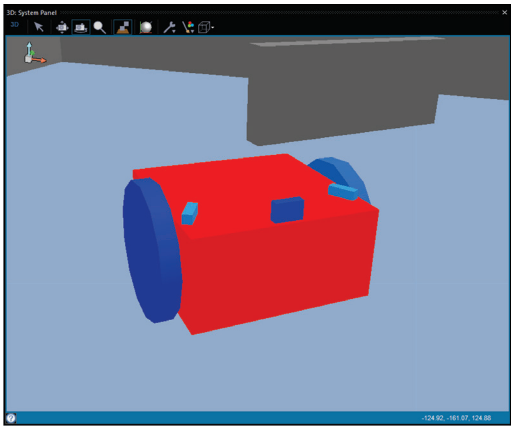

Figure 8: Robot with imitation Sensors.

The structure of the sensor was taken from the previous work [3] and modernized (Fig.9). During testing, it was found that the number of signals is insufficient for ideal operation and control without collisions. For this task, 3 output signals from each simulation module were created to achieve full coverage of the entire space in front of the robot to avoid unforeseen and specific situations.

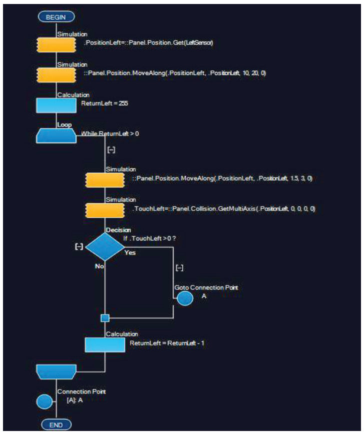

Figure 9: Flowchart imitation model [3].

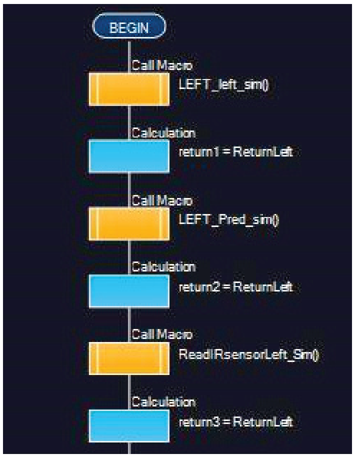

Figure 10: Combined signals part 1.

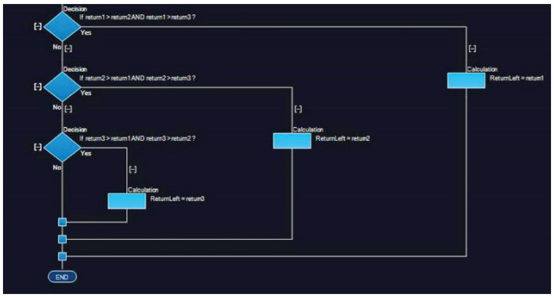

Figure 11: Combined signals part 1.

And then combined all the signals into 1 without changing the main part of the program. Thus, it turned out to create an area of sensor sensitivity, adjusting each parameter separately, depending on its needs (Fig.10).

4. Results

The ultrasonic sensor measures the distance to the target, constantly changing the data coming out of it. Their working principle is based on wavelength [12]. Infrared sensors already have a pre-set distance sensitivity, and their output parameters can be 1 or 0. This is very clearly seen in (Fig.11) where the lateral IR sensors have a constant obstacle measurement value, and the ultrasonic sensor constantly sends and receives a signal back, thereby calculating the distance to the object [14].

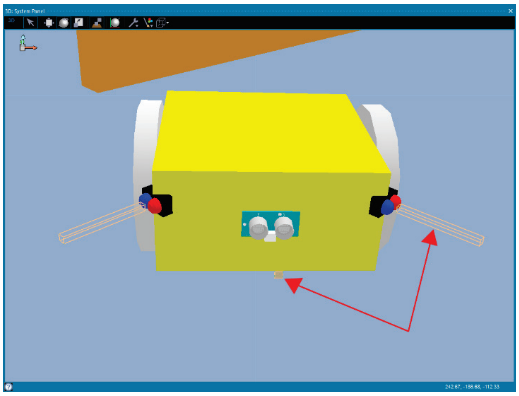

Figure 12: Working Sensors.

The basic window "Simulation debugger" was used to control the distance measurements, which makes it possible to monitor the changes of all variables in real time. (Fig 12).

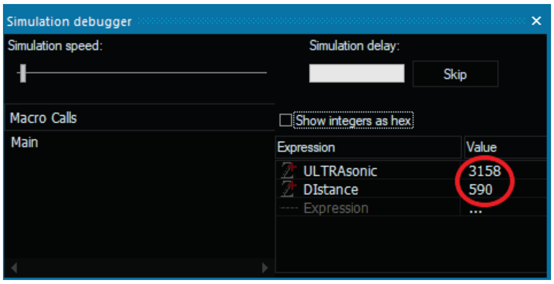

Figure 13: Simulation debugger 1.

The implementation of this idea brought success, during testing it was stated that this modification greatly improves the parameters of the simulation of the movement of a mobile robot bypassing all obstacles in its path (Fig.13). Each of the circled yellow circles represents the location of the sensor signals (small yellow dots inside).

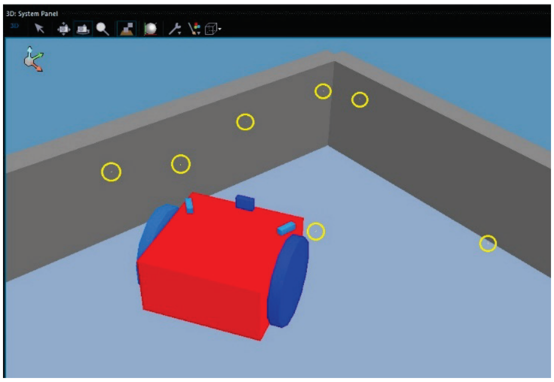

Figure 14: Working Imitation sensors.

Also, as in the first situation, we are monitoring results through the "Simulation debugger" window. Results (Fig.14) in this situation are given not in millimetres, but only in the sensor sensitivity, which is measured from 0 to 255, where each point corresponds to one cycle of motion signal, the distance of which is set manually and depends on the need for sensitivity.

Figure 15: Simulation debugger 2.

The graph shows the result recorded during the movement of robots in their given space and displays changes in the distance from the object to the sensor over a period of time, where large peaks indicate the moment, an obstacle is detected in the danger zone, followed by a change in the direction of its movement. This was implemented using the built-in display in the Flowcode program.

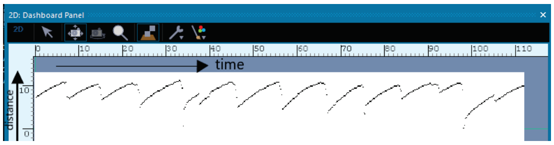

Figure 16: Build-in Sensor results.

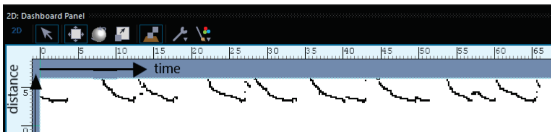

Figure 17: Imitation Sensor results.

5. Conclusions

It takes a lot of time and blocks to create a fully functioning imitating sensor software. At the same time, such detailed elaboration allows well-optimized customization of all the necessary functions as needed. Most autonomous navigation systems are based on different types of sensors such as infrared [7]. Using the program capabilities, it was managed to combine three sensors into one system so that it covers a larger radius around.

The use of ready-made, built-in sensors is much easier in terms of programming, you just need to understand the capabilities of component blocks. This allows you to show what a given sensor looks like in reality and show its functionality. An advantage also is the possibility of quickly using only one component of the transfer function data from the sensor in [mm]. The downside of this option is that it is very poorly customizable and quite limited in capabilities. In my research, it was found that the response of such a device is much less than, where the speed of the ultrasound signal is affected by humidity and temperature of the environment, but in a vacuum is 331.4 ms^{-1} [2]. That creates large delays in situations where the distance to the obstacle increases greatly, Because of problems with the signal response, the performance of the entire robot is reduced.

Program code used to implement and control simulation cannot be used to upload to a real mobile robot. Simulation is used to test and visualize different types of programs and explore ideas.

Acknowledgments

The article was written in the framework of Grant Project: 004TUKE-4/2021 "Development of innovative teaching materials for learning multi-agent robotics"

References and Notes

1. D M Anton et al 2021 IOP Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on Volume: 1 Sci. Eng. DOI: 10.1109/ICCAE.2010.5451881
2. Juruš O. Ferenčik P., 2021, Ultrasonic Sensors used to Detect the Position of the Mobile Service Robot, Acta Mechanica Slovaca 25 (1): 28 – 32,
3. Grytsiv M. Sukop M., 2021 Simulation of Real Conditions in the Movement of Robots, Novus Scientia 2021, ISBN 978-80-553-3798-2
4. Tzafestas S. G. Introduction to Mobile Robot Control. Elsevier, 2014. 750 p. https://doi.org/10.1016/C2013-0-01365-5
5. Landau I. D., Zito G. Digital Control Systems: Design, Identification and Implementation. Springer-Verlag London, 2006. 484 p. https://doi.org/10.1007/978-1-84628-056-6
6. Äström J., Wittenmark B. Computer-Controlled Systems: Theory and Design. (Dover Books on Electrical Engineering). Third Edition. Dover Publ., 2011. 576 p.
7. Zheng, C.; Green, R. Vision-based autonomous navigation in indoor environments. In Proceedings of the 2010 25th International Conference of Image and Vision Computing New Zealand, Queenstown, New Zealand, 8–9 November 2010; pp. 1–7. DOI: 10.1109/IVCNZ.2010.6148876
8. Sariff, N.; Buniyamin, N. Ant colony system for robot path planning in global static environment. In Proceedings of the 9th WSEAS International Conference on System Science and Simulation in Engineering (ICOSSSE'10), Takizawa, Japan, 4–6 October 2010; pp. 192–197.
9. Karur K., Sharma N., A Survey of Path Planning Algorithms for Mobile Robots, Vehicles 2021, 3, 448–468. https://doi.org/10.3390/vehicles3030027
10. Missura M., Roychoudhury A., 2020, Polygonal Perception for Mobile Robots, Conference: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), DOI: 10.1109/IROS45743.2020.9341742
11. Vagaš M., Semjon J., Hajdik M., 2011 Proceedings of 2011 International Conference on Optimization of the Robots and Manipulators, ISBN 978-1-84626-xxx-x
12. Kadëna E., Nguyen P D H, 2021 Mobile Robots: An Overview of Data and Security, Conference: 7th International Conference on Information Systems Security and Privacy DOI: 10.5220/0010174602910299
13. Daoud A., 2021, An Arduino-based Low-Cost Hardware for Temperature Control, WSEAS TRANSACTIONS on SYSTEMS Volume 20, DOI: 10.37394/23202.2021.20.8
14. Ng F., 2020, Ultrasonic Sensors DOI: 10.13140/RG.2.2.33638.78404
15. Flowcode. Matrix Technology Solutions Ltd [online]. [cit. 2022-10-11]. Dostupné z: https://www.matrixtsl.com/flowcode/