Positioning algorithm for AGV autonomous driving platform based on artificial neural networks

Patryk Bałazy[1], Paweł Gut[2], Paweł Knap[3]
Faculty of Mechanical Engineering and Robotics, AGH University of Science and Technology,
Kraków, Poland
[1]Corresponding author
E-mail: [1]*balazy@student.agh.edu.pl*, [2]*pgut@student.agh.edu.pl*, [3]*balazy@student.agh.edu.pl*

Received 7 June 2021; received in revised form 21 June 2021; accepted 29 June 2021
DOI https://doi.org/10.21595/rsa.2021.22109

Abstract. This paper presents an artificial intelligence algorithm responsible for the autonomy of a platform. The proposed algorithm allows the platform to move from an initial position to a set one without human intervention and with understanding and response to the dynamic environment. The implementation of such a task is possible by using a combination of a camera identifying the environment with a laser LIDAR sensor and a vision system. The signals from the sensors are analysed through convolutional neural networks. Based on AI inference, the platform makes decisions, including determining the optimal path for itself. A transfer learning method will be used to teach the neural network. This article presents the results of learning the applied neural algorithm.

Keywords: autonomous systems, smart sensors, neural networks, environmental understanding.

1. Introduction

The everyday life faced by people with mobility impairments is a real challenge for them. They have problems getting from A to B. Of course, they are helped by people such as carers or nurses. However, they are also people who sooner or later get tired and who have their own daily needs. Electric wheelchairs are another way of helping people with disabilities to get around. These take the strain off the arm muscles of people who have mobility problems or complete leg inertia. However, the biggest problem is the movement of people who are not even able to steer the previously mentioned electric wheelchair. An AGV autonomous driving platform equipped with a neural positioning algorithm comes to the aid of people with disabilities.

An autonomous AGV equipped with a positioning algorithm is an innovative solution that can make everyday life easier for people with mobility impairments. The AGV is connected to the cloud, where data on the location of a given place is stored, as well as who, when, how and why it was prepared. An algorithm controls the platform in such a way that it moves the disabled person between two points, choosing the fastest possible route.

The issue of AGV control is the subject of extensive university research [1]-[3]. These are based on image analysis networks [4]-[6], position analysis networks [7], [8], and fuzzy logic [9]-[13]. This paper describes the study of a platform autonomy support algorithm using a reinforcement learning technique in combination with the use of artificial neural networks. The aim of this method is to train an agent to perform a task in an uncertain environment. The agent receives observations from the LIDAR sensor and a reward for its actions from the environment and in response sends actions (called actions) to the agent. The reward is a measure of the success of the action with respect to the achievement of the task goal, defined by the reward function. The agent's task is to accumulate the reward, i.e. to learn how to interact with the environment in the most beneficial way.

2. Environment model

The mobile platform for which the algorithm is developed is a four-wheeled mobile robot equipped with a two-dimensional LIDAR environment scanner. Simulink software was used to simulate the operation of the robot and the sensors. An environment model (FIG1) was created containing the following factors: kinematics model, LIDAR sensor operation simulator, collision signalling, adding noise to LIDAR readings and reward function. The mobile robot uses kinematics based on a differential mechanism. The modelled environment was then used in a further scheme to link it to the agent's actions. It uses the output of the environment, these are observations, collision information and reward value. The agent uses this data both during learning and during simulation.

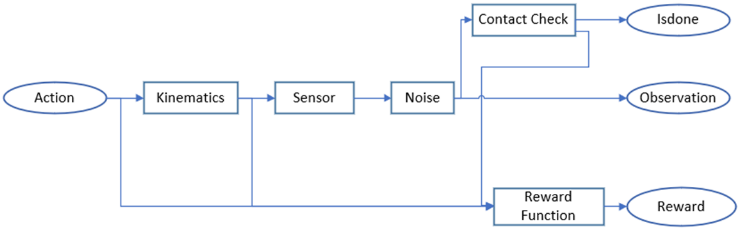


Fig. 1. Block diagram of the environment model

For the model created, the actions are the linear and angular velocity preset and the observations are the readings from the LIDAR scanner. These are normalised and given a range of –1 to 1, where the first value represents clockwise rotation at maximum speed or backward travel at maximum speed, and the second value represents counterclockwise rotation at maximum speed or forward travel at maximum speed.

3. Reward function

A very important element of reinforcement learning is the reward function. It is based on this function that the agent decides which actions to perform and for which it receives the greatest reward. It was used, among others, to ask the final location of our robot. For this purpose, in the course of research work, two reward functions were designed, which allow to reach two different locations on the map. This is defined as a three-dimensional function that has the coordinates of the robot on the x and y axes, while the reward value is defined on the z axis. Each function has one maximum for the coordinates of the robot's final destination. It was also decided that the reward function should only take negative values, this prevents the reward from accumulating while the robot stands still. The designed functions are presented in three-dimensional space (Fig. 2, Fig. 3) and in two-dimensional space superimposed on a room map (Fig. 4, Fig. 5).

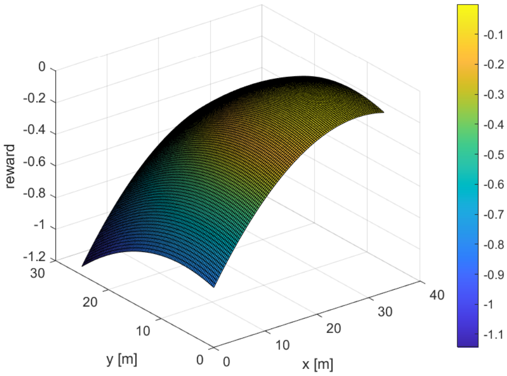


Fig. 2. Reward function for the room in the bottom right corner of the map

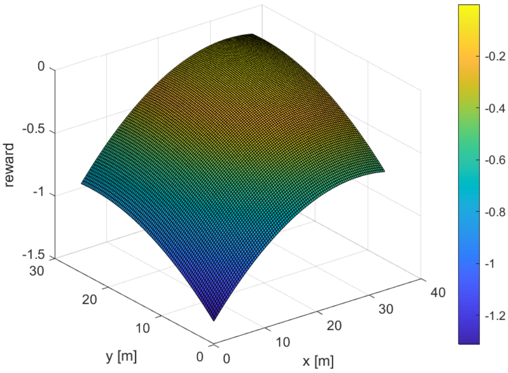


Fig. 3. Reward function for the room in the top right corner of the map

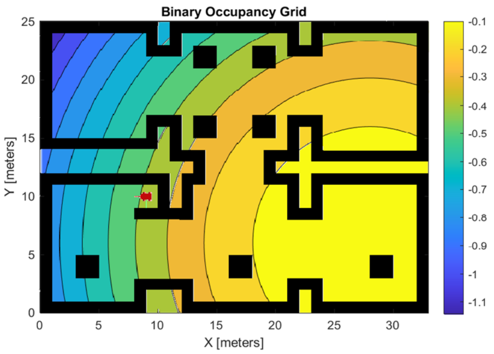


Fig. 4. Room map with superimposed reward function 1

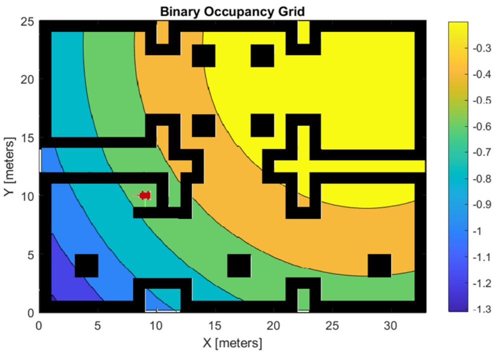
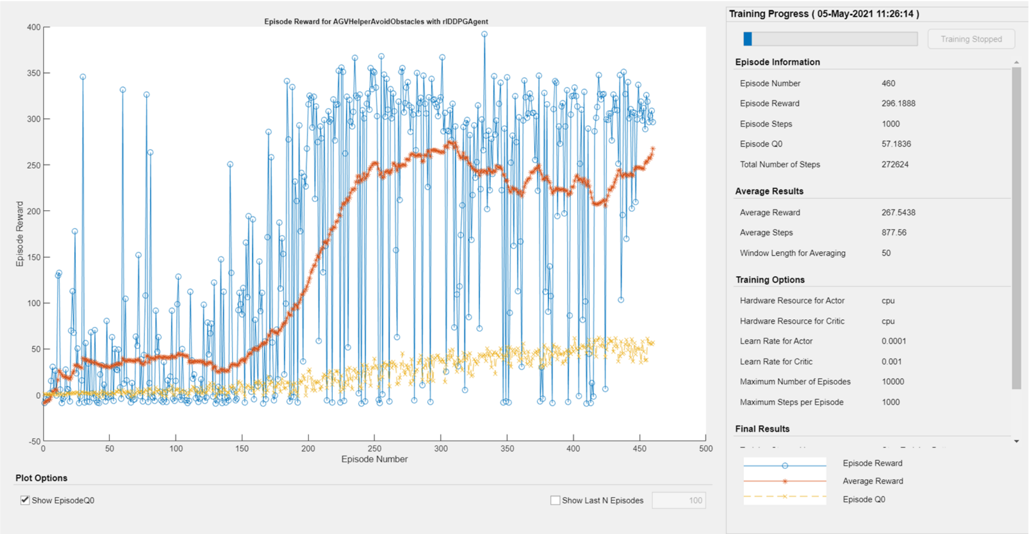


Fig. 5. Room map with superimposed reward function 2

The higher the intensity of the colours in the graph, the higher the value of the reward function, which makes the robot more likely to head towards higher values of the reward function. With the help of the reward function it is possible to define the final destination that the robot should go to. As can be seen in Fig. 4, the most intense colours are in the room at the bottom right of the map and this is where the robot will head. Fig. 5 shows the reward function for which the robot will be directed to the room located in the upper right corner of the map.

4. Training algorithm/result

Once the environment model has been created, and the reward function designed, the process of teaching the agent can proceed. This is a time-consuming process, as each episode involves simulating the agent in the Simulink environment. It also requires a lot of computing power. Learning is completed when the average reward is reached. Fig. 6 shows the progress of agent learning after three hours. The DDPG agent was used because it is often proposed as the first choice in environments with continuous action spaces and observation. The actor and critic functions are deep neural networks. DDPG agent is an actor-critic, model-free algorithm based on the deterministic policy gradient.



Training Progress (05-May-2021 11:26:14)

Episode Information

Episode Number	460
Episode Reward	296.1888
Episode Steps	1000
Episode Q0	57.1836
Total Number of Steps	272624

Average Results

Average Reward	267.5438
Average Steps	877.56
Window Length for Averaging	50

Training Options

Hardware Resource for Actor	cpu
Hardware Resource for Critic	cpu
Learn Rate for Actor	0.0001
Learn Rate for Critic	0.001
Maximum Number of Episodes	10000
Maximum Steps per Episode	1000

Final Results

Fig. 6. Progress of the agent's learning process

During the learning of the agent we received many failures, this was mainly related to an inadequately developed reward function. The platform instead of avoiding obstacles collided with a wall. In the end we managed to train the agent so that the platform drives correctly. The simulation of the trained algorithm is presented in Fig. 7. As can be seen, the platform spontaneously selects a path to the destination, avoiding on its way an obstacle in the form of a pillar. During the training of the agent in this case the reward function was used, directing the platform to a room located in the lower right corner of the map.

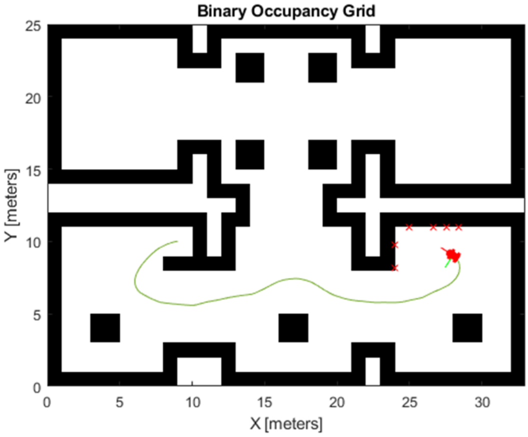


Fig. 7. Driving simulation of an autonomous platform

5. Conclusions

The result of the work is a properly working algorithm that enables the mobile platform to move autonomously to a given destination. It allows for the optimal trajectory to be determined, avoiding obstacles encountered on the way. Ultimately, the algorithm will be implemented in a real AGV robot. The results of work on the hardware implementation of the created algorithm and field tests will be the subject of future publications.

The use of reinforcement learning in system control tasks works very well and significantly facilitates the implementation of such solutions. It also allows to obtain better behaviour of the robot in an unknown environment.

During the work, the learning time of the algorithm, despite the use of parallel learning, proved to be extremely problematic. This meant that even the smallest changes to the structure of the reward, actor or critic functions, or a change in the learning parameters required re-training and waiting for hours for results. This significantly affected the development time of the algorithm.

References

[1] F. Gul, W. Rahiman, and S. S. Nazli Alhady, “A comprehensive study for robot navigation techniques,” *Cogent Engineering*, Vol. 6, No. 1, p. 1632046, Jan. 2019, https://doi.org/10.1080/23311916.2019.1632046

[2] M. Soylu, N. E. Özdemirel, and S. Kayaligil, “A self-organizing neural network approach for the single AGV routing problem,” *European Journal of Operational Research*, Vol. 121, No. 1, pp. 124–137, Feb. 2000, https://doi.org/10.1016/s0377-2217(99)00032-6

[3] J. Keighobadi, K. A. Fazeli, and M. S. Shahidi, “Self-constructing neural network modeling and control of an AGV,” *Positioning*, Vol. 4, No. 2, pp. 160–168, 2013, https://doi.org/10.4236/pos.2013.42016

[4] J. Varagul and T. Ito, “Simulation of detecting function object for AGV using computer vision with neural network,” *Procedia Computer Science*, Vol. 96, pp. 159–168, 2016, https://doi.org/10.1016/j.procs.2016.08.122

[5] Y. Bengio, “Learning deep architectures for AI,” *Foundations and Trends® in Machine Learning*, Vol. 2, No. 1, pp. 1–127, 2009, https://doi.org/10.1561/2200000006

[6] M. Gupta, “Deep learning,” in *International Series in Operations Research and Management Science*, Cham: Springer International Publishing, 2019, pp. 569–595, https://doi.org/10.1007/978-3-319-68837-4_17

[7] S. Ito, M. Soga, S. Hiratsuka, H. Matsubara, and M. Ogawa, “Quality index of supervised data for convolutional neural network-based localization,” *Applied Sciences*, Vol. 9, No. 10, p. 1983, May 2019, https://doi.org/10.3390/app9101983

[8] Z. Yan, B. Ouyang, D. Li, H. Liu, and Y. Wang, “Network intelligence empowered industrial robot control in the F-RAN environment,” *IEEE Wireless Communications*, Vol. 27, No. 2, pp. 58–64, Apr. 2020, https://doi.org/10.1109/mwc.001.1900346

[9] I. Dominik, “Advanced controlling of the prototype of SMA linear actuator,” *Solid State Phenomena*, Vol. 177, No. 1, pp. 93–101, Jul. 2011, https://doi.org/10.4028/www.scientific.net/ssp.177.93

[10] I. Dominik, “Implementation of the Type-2 fuzzy controller in PLC,” *Solid State Phenomena*, Vol. 164, pp. 95–98, Jun. 2010, https://doi.org/10.4028/www.scientific.net/ssp.164.95

[11] W. Huang, C. Yang, J. Ji, N. Chen, and K. Xu, “Study on obstacle avoidance of AGV based on fuzzy neural network,” in *2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, Oct. 2019, https://doi.org/10.1109/imcec46724.2019.8983996

[12] C.-L. Hwang, C.-C. Yang, and J. Y. Hung, “Path tracking of an autonomous ground vehicle with different payloads by hierarchical improved fuzzy dynamic sliding-mode control,” *IEEE Transactions on Fuzzy Systems*, Vol. 26, No. 2, pp. 899–914, Apr. 2018, https://doi.org/10.1109/tfuzz.2017.2698370

[13] V. F. Caridá, O. Morandin, and C. C. M. Tuma, “Approaches of fuzzy systems applied to an AGV dispatching system in a FMS,” *The International Journal of Advanced Manufacturing Technology*, Vol. 79, No. 1-4, pp. 615–625, Jul. 2015, https://doi.org/10.1007/s00170-015-6833-8

Patryk Bałazy received his Degree in engineering automation and robotics from the AGH University of Science and Technology, Krakow, Poland, in 2021. His current research interests include control and machine learning.

Pawel Gut received his engineering Degree in automation and robotics from the AGH University of Science and Technology in Krakow, Poland in 2021. His current research interests include artificial intelligence and the use of field-programmable gate array in engineering solutions.

Pawel Knap received his engineering degree in automation and robotics from the AGH University of Science and Technology in Krakow, Poland in 2021. His current research interests include vibrodiagnostics and the use of artificial intelligence in engineering solutions. He is also passionate about combat sports and an active member of AZS Judo AGH club.