Simulator of the navigation equipped with LIDAR of the mobile robot based on the neural network

R V Faizullin

Kalashnikov Izhevsk State Technical University, Faculty of Digital Economics, Studencheskaya St. 42, Izhevsk, 426069, Russia

E-mail rf85@mail.ru

Abstract. The paper considers the issue of using the robot simulators equipped with LIDAR (Light Detection and Ranging). The simulator application allows decreasing the experiment costs, their time and risks or negative aftereffects of accidents, but, at the same time, the simulated environment should identical to the real one. In practice, the simulated environment can have differences and there comes the issue of estimating the robot ability to adapt to new conditions. Neural networks are used to teach the robots, due to which the robots reach the target in the shortest or the fastest way, at the same time, there should be no accident/collision. The initial series represents the multiple time series. To teach the neural networks to control the robots, the paper proposes to introduce the additional index in the form of the minimal distance to the nearest object in the actual time moment. The approach novelty in using the new penalty function for teaching the neural network is the speed index of approaching the nearest object/obstacle. The neural networks were taught on the maps without dynamic objects, but tested in the conditions when several mobile robots were used on the map, thus, they became the dynamic obstacles for each other. The accident risk for robots without the proposed penalty function during testing was 4.5% against 0.3% - with the penalty function.

Keywords: mobile robot, LIDAR, laser sensors, collision, neural network, simulator.

1. Introduction

The robotics development is one the technological development trends in general. The desire to computerize part of the functions is widespread and perspective. Computerization allows decreasing the influence of subjectivity and human factor, and opens the way for improving the productivity of system activity. At the modern development stage robotics allows solving many problems in different knowledge areas.

One of the characteristics of a robotic system is its ability to interact with the environment. Interaction with the environment assumes the ability to identify the surroundings and capability of self-localization in the random environment. Several ways of solving similar problems are known [1, 2, 3], but their success depends directly on the ideal conditions and cost of such systems. The high cost of the sensor part of the known autonomous vehicles (e.g., self-driving cars) is the obstacle to their widespread occurrence. For this reason, great many papers dedicated to the development of autonomous motion algorithms using a relatively inexpensive element base have been recently published [4, 5].

Taking into account the application scale and complexity of robots, the development of applied robotics generates interest in the modern world. In reality, the data collection or system testing can be
expensive, insecure or impracticable. The simulator application allows improving the security and decreasing the cost of robot testing.

The development of simulator to visualize the action spot comes down to three components. The first is the sensor simulation, which describes how the sensor interacts with the action spot (space). The second is the generation (visualization) of the action spot necessary to construct simulated worlds corresponding to the reality. The third is the simulator quality assessment based on the comparison of real and simulated data. The availability of simulated spaces allows imitating the process of robot navigation in this space. The application of computer-aided learning and artificial intelligence for robot control allows teaching it to reach the targets, hurdle obstacles, do optimal routing.

The use of 2D LIDAR by mobile robots and solution of the problem for creating navigation systems based on these data is an urgent task. The attempts are made to accomplish this task, the description of which can be found in the following papers [2, 3, 6, 7].

At present, there is great number of modifications of Simultaneous Localization and Mapping method (SLAM method) adapted for various sensor systems. But the basis of all SLAM modifications is Iterative Closest Point algorithm (ICP algorithm). ICP algorithm and its widespread variations are given in the paper [8].

Let us consider the mobile robot equipped with single beam laser scanner LIDAR. LIDAR is Light Identification Detection and Ranging (“detection, identification and ranging with the help of light”), it is the technology of obtaining and processing information about distant objects with the help of active optical systems using the light absorption and dissipation phenomenon in optically transparent media.

LIDAR data serve as the source for forming the information about the environment. If the sensor transfers the information about the distance from the sensor to the objects around it and there is no other information, the mobile robot action spot is formed only on the available information. The data set is not a perfect method to describe the action spot. The data on the distances “sensor – observation object” can be distorted. The distortion can be caused by the measurement, rounding off and other errors. This fact does not mean that we cannot form the information about the action spot but we should take into account the possible error in the incoming data. The main task of the data collection about the robot action spot is the action strategy formation, i.e. formation of the options of the system reaction to certain characteristics of the environment.

2. Research objective

In order to solve a certain problem of the system, the formation of its mathematical formulation becomes necessary. Let us formulate the problem as follows. A mobile robot (MR) should not approach another object closer than some distance. Two options should be singled out from the point of actual situations. The robot can collide with a stationary (immobile) or mobile object. Since we may not know about the behavior of mobile objects in the surroundings, we should consider the environment as the one with a random component.

The task of preventing the mobile robot collision with objects of the environment can be split into two sub-tasks: identification of the obstacles in front of the robot and movement routing to bypass the obstacles. To accomplish the first sub-task it is necessary to make the following steps:

1. To obtain the data about the distances to the surroundings.
2. To extract the relevant information (possibly being the obstacles).
3. To process the data and obtain information about the object, its danger, trajectory.

The task of object recognition can be significantly simplified, not taking into account the height and shape of the objects in vertical plane. For this, it is sufficient to check the obstacles not by the whole picture in the vertical plane but only along one line located at a certain height from the floor. In our experiments this height was determined by the MR height.

Modern algorithms of MR control based on LIDAR data obtained by computer-aided learning methods allow making the control trouble-free.
Despite the attempts to develop trouble-free control systems, they are still rather urgent. Even with the information from different sensors (radar, LIDAR and cameras), the optimal solution has not been found so far.

The work relevance is connected with broad capabilities of using the results. Let us consider the case of the car trouble-free driving. A human factor is the main reason of car road accidents. The system, which could predict collisions, could significantly decrease a number of road deaths. We can use different sensors but not all of them work ideally. The problems in the sensors operation can be caused by the weather conditions. Even if the sensors obtain the correct information, it should be correctly processed, and the car should have time and possibility to take measures for avoiding an accident.

Apart from the data from the sensors, cars use the wireless automobile communications. There are several types of such communications: vehicle-to-everything (V2X) communications, vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and vehicle-to-pedestrian (V2P). The investigations demonstrated that the combination of V2V and V2I technologies can help to avoid about 80% of all car accidents [9].

In this paper [10], the development of a vehicle collision warning system based on multisensors and V2X communications is presented. On-board sensors including radar, LIDAR, and camera systems that have already been adopted in production vehicles are chosen for this work such that by adding V2X communication devices to the vehicle, it is possible to evaluate the benefits of introducing V2X communications to today’s vehicles in terms of road safety.

However, simple models are also necessary. If the model is simple, it requires less resources and works faster and clearer for the user. In the paper [11] the method to estimate the risk of collision with an obstacle is proposed. The simulated data on the moving object coordinates are added to the LIDAR data. The trajectory prediction allows defining the collision possibility. The collision fact is predicted, the time before the collision is estimated with the help of the analytical method. The use of additional information about the object physical properties improved the accuracy of the results modeling and collision risk estimation, and decreased the computational and memory load on the results processing system.

Neural networks are successfully used to develop the algorithm of robot control [1, 2, 3, 4, 6, 7], however, there are no proofs that the robot route will be ideal. Apart from the optimality, the majority of papers do not consider the possibility of force majeure situations or situations, which were not envisaged at the stage of neural network teaching. Such situations can be connected with the sensors errors; in case of LIDAR, there can be problems with reflecting and transparent objects, the robot can change its behavior by itself, because, for example, its battery is getting down. The robot can collide with the object, which it has not seen before (in the teaching process), or collide with the dynamic object with unpredictable behavior. Those robots, which consider the possibility of dynamic object availability and predict its trajectory [9, 10, 11], do not take into account the random trajectory changes.

Thus, we come across the necessity of developing such neural network and selecting such development criteria for the robot to cope with situations not envisaged at the simulation/teaching stage.

3. Methodology

Let us consider the change in the action spot state as the change in the distances from the mobile robot location to the objects detected by LIDAR sensor. The collision detection system [11] is targeted at determining if and when the collision between the mobile system and environment can occur to notify the motion planning system (or a human operator) that the current motion trajectory needs to be adapted to avoid the collision. Various approaches exist for detecting the presence and range of an obstacle; at present, they are widely used in mobile systems with beam ranging and swept area predictions being the most common.

Beam ranging uses a concept of directing some sort of energy towards the obstacle and evaluating the reflection. Most common are: acoustic waves (sonars), radio waves (radars) and light (LIDAR), with ranging devices providing the information on the distance between the device and an obstacle. Single or multiple sensors can be attached to a mobile system, covering the area of the predicted motion.
As the mobile system is moving along the predicted trajectory, its bounding box sweeps over an area in the environment (also called configuration space). In order to determine the possibility of collision between the mobile system and environment, each measurement must be compared against the predicted path of the system. It is possible to predict this swept area in advance and check whether any of the detected obstacles are within this area. In this case the time to collision can be estimated based on the predicted moment the mobile system’s bounding box touches the obstacle.

A lot of dynamic objects make the task more complicated. A mobile robot has limited resources. This is caused by the limited computational and memory power and limited energy storage. So, the necessity to seek a simplified method of collision prediction arises.

Under collision between objects we understand the situations when the distance between them equals zero. The robot with LIDAR can detect the distance to the obstacle with the accuracy over 99%. The experimental results [12] demonstrate that LIDAR can measure the distances from 0.12 up to 10.5 m with the error around 0.9%. The object color and environment light intensity do not influence the measurements obtained with the help of LIDAR. An autonomous mobile robot can avoid colored objects of different sizes. The robot faced the problems of distance detection only with transparent objects.

So, it follows that we should solve the problem of predicting the multiple time series. Actually, the data flow from LIDAR is the multiple time series.

Let us set the task of multiple time series prediction.

Let us designate \( X = [x^{(1)}, ..., x^{(s)}]^T \) is the given \( s \)-dimensional time series. Now we formulate the design matrix from the series segments:

\[
\begin{pmatrix}
  x_0^{(1)} & x_1^{(1)} & \cdots & x_{n-1}^{(1)} \\
  \vdots & & & \vdots \\
  x_0^{(s)} & x_1^{(s)} & \cdots & x_{n-1}^{(s)}
\end{pmatrix} = X_{0 \cdot (n-1)}
\]  

(1)

Let \( X = [x_0^{(1)}, ..., x_0^{(s)}]^T \) is the value of X series in the time moment \( n \). Now we build up the prediction \( \hat{X} \) of X series in the point \( X_n \). Let us do it \( k \) times for different teaching samples \( X_{t_{train}}^t = X_{(n+i \cdot t-1)}, i = 0, ..., (k - 1) \). We get \( k \) predictions \( \hat{X} = [\hat{X}_n, \hat{X}_{n+1}, ..., \hat{X}_{n+k-1}] \).

The prognostic model looks as follows:

\[
\hat{X}_{t+1} = f(\hat{w}, X_t, X_{t-1}, ..., X_{t-L+2})
\]

(2)

\[
\hat{w} = \text{argmin} S(w, X, \hat{X}_n, \hat{X}_{n+1}, ..., \hat{X}_{n+k-1}) = S(w, X, \hat{X})
\]

(3)

where the loss function:

\[
S(w, X, \hat{X}) = \sum_{i=0}^{k-4} L(X_{n+i}, \hat{X}_{n+i}) = \sum_{i=0}^{k-4} L(X_{n+i}^{(1)}, \hat{X}_{n+i}^{(1)})
\]

(4)

In case of LIDAR, which transfers us the data on the distances of the objects in 360 directions in each time unit, \( s=360 \). Depending on the prediction model, we should also choose the number of time periods, which influences the current value. We select \( n \) periods for matrix (1), then our computations have to be carried out with the matrix with the dimensions \( s^* n \). The complexity of problem solution is caused not only by the larger size of the data analyzed but also by the availability of complex bonds in them. It means that the data will change depending on where to and along which trajectory the mobile robot moves, as well as on the behavior of dynamic objects in the robot vision area.

For dynamic objects not to detain mobile robots, the robots should be able to predict the behavior of the dynamic objects. The given task is being actively accomplished, for example, for self-driving cars. In such tasks other automobiles and pedestrians are considered as dynamic objects [13]. The prediction model should take into account future interactions with an unknown number of dynamic objects. It is necessary to bear in view that dynamic objects can also change trajectories as a result of our behavior or changes in the behavior.

The model teaching based on the simulated data is a widespread practice, though it has a number of shortcomings. It is easier, faster and cheaper to obtain the simulated data.

In our paper, we consider the task of selecting the teaching process of the neural network based on simulated data. The program operation starts from the level loading location of all obstacles, MR starting
point, as well as the area, in which the stop should be made. After that the first generation with random parameters of neurons weight is generated. The neurons are represented in the memory as the samples of class “Neuron”. The class contains the array of numbers: neuron weight, mutation function randomly changing up to 1/3 values from this array, and activation function “Activation()”. One specimen is represented as a sample of AI class. AI class consists of three layers (arrays) of neurons, mutation method “mutate()” calling the mutation function of all neurons in all layers with the chance of 40%, crossing-over function “crossBreed()” creating one new specimen out of two, and decision-making function “makeDecision()”. The decision-making function of the specimen activates all neurons one by one in all layers producing 4 Boolean values at the output. These values indicated what action should be taken at the given time moment based on the environment data, whether it is necessary to speed up or slow down, whether it is necessary to turn left or right.

After the first generation is initialized, the evolution cycle starts. We create some number of specimens. Each specimen is a different set of weights of neurons and, consequently, a separate way of MR reaction to the environment data.

The game cycle starts:

For all specimens of the first generation the environment analysis is initiated by data from the LIDAR, i.e. the data on the distances to the objects by all 360 degrees. After that the function makeDecision()” is called upon for each specimen of the first generation. Based on the decision, the shift of the robot (robot version) fixed to it is calculated; taking into account the current speed, acceleration and its turning angle the specimen will get the scores for the distance covered within one frame. The frame is renewed. The accident risk is calculated for each robot; if it collides with the obstacle, it stops, it is marked with the attributes “crashed” and “finished”, and gets the scores by the adaptability function: for the distance to the target, as well as the penalty (punishment) is the number of scores is decreased by 20%. Further, the car marked as “finished” does not participate in decision-making, miscount of collisions or shift (the game is over for it). If the time of one game cycle exceeds the set time, the car is marked as “finished”, and we calculate the scores with the help of the adaptability function. Then it is estimated if there are still other cars without the attribute “finished”. If there is at least one available, the game cycle is iterated again. The specimen is given scores if it reaches the target. If even none specimen has reached the target and if they have covered different distances, those who covered longer distances will have more scores.

When all the cars are finally marked as “finished”, the time for creating the new generation comes. The new generation is formed by cross-breeding the best specimens of the previous generation. That is, the couple of parents from the best specimens of the previous generation is selected, and the function “crossBreed()” is called upon. The specimen obtained becomes a new member of the next generation, and the method “mutate()” is called upon for it.

Let us teach several neural networks selecting different criteria when awarding the specimens. Let us further consider the capabilities of the neural networks, which were taught on the static data, to get adapted to the situations with the emergence of dynamic objects. This situation will be an example for estimating the capabilities of neural networks to non-standard situations.

4. Results
During the investigation it was revealed that the main aim of teaching the robots is to reach the target, the distance and time criteria are used for optimization. Although the time spent by the robot depends on the distance, the criteria are not identical. If we optimize only the distance covered, the robot stop will not influence the result, therefore, the time is more important. On the other hand, we understand that the distance covered mostly influences the energy spent. The time spent by the robot for covering the distance depends on the robot average speed. Finally, these three indexes are the most widely spread in the process of robot teaching [14, 15, 16].

The collision prediction based on solving the prediction task of zeros emergence in the problem as (1), is a neutral task, however, we propose the model simplification. The task complexity is connected with the fact that, on the one hand, the multiple time series reflects the distance to the obstacles around,
and, on the other hand, this distance changes with time not only based on the location of obstacles but also on the changes in the robot location. From the information capacity point, the information which of 360 points that are described by the LIDAR data will become zero is not that important as the prediction of time (moment) when it happens. What is the difference from what side the hit comes, if we do not know when? Let us consider the possibility of using the information on the proximity of the nearest object instead of the information on the remoteness of the objects from all sides. Thus, it is proposed to transform the multiple series into the one-dimensional one. In our case, we are interested in the nearest point, therefore, the minimal value (min \( x_n \)). Since random factors influence the collision fact and its time, the prediction is only some expected value with some probabilistic characteristics.

Actually, the task comes down to extrapolation. If the schedule describing the change in the minimum distances from MR to the obstacle can be described by some curve (straight line or parabola), it is enough to understand if this curve takes the zero value in the future time moments.

The latest observations of the considered value are more important than the old ones, therefore, we cannot use, for instance, the model of simple moving average. Thus, there comes the task that the newer data are the most important. It was also considered that the expected collision time or expected distance to the nearest object/obstacle can have some error. So, it becomes necessary to modify the neural network processing and introduce the penalty not only for collision but also for the “proximity/approximation” to the object. Here, we should consider two moments. First of all, the danger of approaching changes non-linearly, and, second, the fact of approximation cannot be dangerous in some cases, for instance, when the robot moves along the wall but close to it. We understand that if we give penalty for the approximation to the object/obstacle in each time moment, the robot will not move close along the wall, though, it can be an optimal decision (the shortest way). The penalty was proposed not only for the proximity to the object/obstacle, and for the speed proximate to the object/obstacle, i.e. the speed of changing the minimum distance was considered. So, if the robot moves along the wall, it does not get additional penalties for the wall proximity apart from those it got in the moment of approximating them. Let us repeat that the penalties are increased non-linearly.

Thus, we achieved the sufficient decrease in the potentially dangerous situations.

The neural network was tested as follows. We have trained the neural network using the methods described in the previous section. We found out that after 10,000 iterations, the robot finds the target well and does not cause accidents. If the robot did not make an accident in the last 300 tests, we believe that the robot has been trained to drive safely. The neural networks taught in different ways were used for the robot control simulator in the new environment. If the teaching was carried out only for different maps, but without dynamic objects, the possibility of robot adaptation to new conditions was initiated at the map of several robot simulators. The robots “competitors” played the role of dynamic objects. Dynamic nature of the environment is the reason of the external faults or exogenous events for robot that are not the fault of the system [17]. These faults are usually unpredictable [18].

The neural networks taught to 1) minimize the distance covered to the target, 2) minimize the time to reach the target, 3) with the combined target were compared. Also the comparison was made in parallel with penalty function for the speed to decrease the minimum distance to the obstacles and without the penalties.

During the experiments it was revealed that neural networks, which have penalties for speed slowing down to the minimal distance to the object during teaching, demonstrated the results better than those, which did not have penalties while teaching. Mobile robots, which did not know that the danger is in the fact that the speedily shortened distance to the nearest object is speedily decreasing not to get into accidents, get the target better. Robots controlled by neural networks without a penalty function, collided, especially when a large number of robots were used on the map and their concentration was high. Роботы проверялись на разных картах с разным числом роботов-«конкурентов». The accident rate was calculated based on the results of 300 tests in each condition (by different maps and by different number of robots - "competitors" on maps). The maximum accident rate is mentioned below. The accident occurrence for a robot without the proposed penalty function is 4.5%, against 0.3% - with the penalty one.
5. Conclusion

Thus, the conclusions that neural networks allow controlling robots and are able to learn making the routes to the target are obtained. The optimization of the route target reach can be made by time and distance. The use of the average speed is the combination of previous approaches as the average speed is the ratio of the distance covered to time. During the investigation it was revealed that the robot should consider the minimal distance to the nearest object. To teach the neural networks to control robots, the paper proposes to introduce an additional index as a minimum distance to the nearest object in the given time moment. The approach novelty is in the use of a new penalty function to teach the neural network is the speed index to approach the nearest object/obstacle. The neural networks were taught on the maps without dynamic objects but they were tested in the conditions when several mobile robots acted on the map, thus, they became dynamic obstacles for each other. The accident of the robot accident without the penalty proposition when testing was 4.5%, against 0.3% with the penalty one. The proposed testing approach to root control can be used for other models. The use of the proposed penalty function is also appropriate for teaching with more diverse situations. Since the robot could adapt well to the emergence of dynamic objects, it can be ascertained that the robot can somehow avoid the prediction of forecasting and avoiding emergencies. The proposed penalty function can be used to predict the accidents moments. These suggestions are explained by the fact that the initial data were the ones with LIDAR data, and during teaching there were no dynamic objects on the map, consequently, the program could know how to learn to escape the collisions predicting the dynamic object trajectory. The investigation of other possibilities of using the other penalty function can be directed by future investigates.

Acknowledgment

The work was carried out with the financial support by Kalashnikov Izhevsk State Technical University within the framework of the grant no. FRV/20-55-25.

References

[1] Lozano-Perez T 2012 Autonomous robot vehicles (Springer Science & Business Media)

[2] Csaba G, Somlyai L and Vamosy Z 2018 Mobil robot navigation using 2D LIDAR 2018 IEEE 16th World Symposium on Applied Machine Intelligence and Informatics (SAMI) doi:10.1109/sami.2018.8324002.

[3] Cheng Y and Wang G 2018 Mobile robot navigation based on lidar 2018 Chinese Control and Decision Conference (CCDC) doi:10.1109/ccdc.2018.8407319

[4] Yeon A S A, Kamarudin K, Visanathan R, Mamduh S M, Kamarudin L M, Zakaria A and Shakafl A Y M 2015 Feasibility analysis of 2D-SLAM using combination of Kinect and laser scanner Jurnal Teknologi, 76(12).

[5] Gong Z, Li J and Li W 2016 A low cost indoor mapping robot based on TinySLAM algorithm 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS) pp 4549-4552

[6] Carlos E Aguero, et al. 2015 Inside the virtual robotics challenge: Simulating real-time robotic disaster response. IEEE Transactions on Automation Science and Engineering, 12(2) pp 494–506

[7] Shah Sh, et al, 2018 Airsim: High fidelity visual and physical simulation for autonomous vehicles Field and Service Robotics (Springer) pp 621–635

[8] Chetverikov D, Svirko D, Stepanov D and Krsek P 2002 The trimmed iterative closest point algorithm Object recognition supported by user interaction for service robots (IEEE) vol 3 pp 545-548.

[9] Najm Wassim G, et al. 2010 Frequency of target crashes for intellidrive safety systems (United States. National Highway Traffic Safety Administration)

[10] Baek, Minjin, et al. 2020 Vehicle Trajectory Prediction and Collision Warning via Fusion of Multisensors and Wireless Vehicular Communications Sensors 20.1 288

[11] Bosnak M and Skrjanc I 2017 Efficient Time-To-Collision Estimation for a Braking Supervision System with LIDAR 3rd IEEE International Conference on Cybernetics (CYBCONF) doi:10.1109/cybconf.2017.7985775
[12] Hutabarat D, Rivai M, Purwanto D and Hutomo H 2019 Lidar-based Obstacle Avoidance for the Autonomous Mobile Robot 12th International Conference on Information & Communication Technology and System (ICITS) doi:10.1109/icts.2019.8850952

[13] Rhinehart N, McAllister R, Kitani K and Levine S 2019 Precog: Prediction conditioned on goals in visual multi-agent settings Proceedings of the IEEE International Conference on Computer Vision pp 2821-2830

[14] Gao M and Tian J 2007 Path planning for mobile robot based on improved simulated annealing artificial neural network Third International Conference on Natural Computation (ICNC) vol 3 pp 8-12

[15] Al-Sagban M and Dhaouadi R 2012 Neural-based navigation of a differential-drive mobile robot 12th International Conference on Control Automation Robotics & Vision (ICARCV) pp. 353-358

[16] Algabri M, Mathkour H, Ramdane H and Alsulaiman M 2015 Comparative study of soft computing techniques for mobile robot navigation in an unknown environment Computers in human behavior (Elsevier Ltd)vol 50 pp 42-56

[17] James P Gunderson and Louise F Gunderson 2008 Robots, Reasoning, and Reification. (Springer Science & Business Media).

[18] Bhat P, Thoduka S and Plöger P 2019 A Dependency Detection Method for Sensor-based Fault Detection.