D-Star Algorithm Modification

https://doi.org/10.3991/ijoe.v16i08.14243

Seifedine Kadry (✉)
Beirut Arab University, Beirut, Lebanon
s.kadry@bau.edu.lb

Gennady Alferov, Viktor Fedorov
St. Petersburg State University, St. Petersburg, Russia

Abstract—One of the most effective methods for solving a navigation problem is the method of constructing a navigation system based on the simultaneous localization and mapping algorithm and obstacle avoidance algorithms. One of the most effective obstacles avoidance algorithms is the D-star algorithm, which, despite its effectiveness, has some drawbacks. This modification allows to eliminate some problems arising during the implementation of the navigation system.

Keywords—Mobile robots, navigation, movement, algorithm.

1 Introduction

Nowadays, automation of processes in which a robot is able to replace a human being is widespread. To solve this problem, it is necessary that robots can move independently from one position to another. Known methods on this subject have a number of disadvantages.

2 D-Star Algorithm Modification Principles

The most effective of these is the method of constructing a navigation system based on SLAM and D * algorithms [1-3]. This paper presents an algorithm based on the principles of this system and correcting a number of its shortcomings [2-6].

2.1 Basic principles

To implement the modification of the algorithm D *, it is necessary to present the map of the environment in the form of a grid map. All cells are divided into free cells and cells with obstacles. In addition, the initial position of the robot and the cell corresponding to the target point are marked on it.

For effective work of this method when constructing a trajectory in a deterministic medium, the A-star algorithm is used [7-12].

This method allows you to simplify the navigation system to one algorithm. When using it, there is no need to use localization algorithms.

2.2 D-star algorithm modification

- Set the cell on the map, corresponding to the initial position of the robot and the target point. In the case when we need to move not from any particular point to a specific one, but travel in a certain direction a certain distance (i.e. when we know only the location of the target point relative to the robot) the map is built relative to the current position of the robot so it does not matter which point we take as the starting point.
- Scan the environment. It is advisable to scan for all $360°$ This is done with the help of rangefinders. The optimal set of them is one infrared range finder fixed at the front, one at the back, and another at the servo that can turn $360°$. Other types of sensors can be used, but the infrared rangefinder is optimal because it gives the most accurate results. When trying to use ultrasonic range finders, the algorithm may not work effectively, and it may also collide with an obstacle. Improving the operation of the scanning system contributes to the use of filters obtained scans. The best option is the Kalman filter.
- When an obstacle is detected, we calculate which cell it belongs to and add it to the array of obstacles.
- Using the algorithm A *, we construct the initial trajectory of the robot. This trajectory is searched for from the target point to the initial position, which allows you to know the exact distance from each point considered to the target. Add all considered points to the history list, which stores the following information about each point considered:

  - Is the weight of m equal to the sum of the weight of the previous point and the distance from it to the current one. The weight of the target point is zero, and the weight of each cell containing an obstacle tends to infinity.
  - Previous $(\mathrm{k}-1)$ cell, i.e. the one that will be the next point in the trajectory obtained using the A* algorithm.
  - Path length from target cell $\mathrm{p(k)} = \mathrm{m(k)} + \mathrm{p(k}-1)$.
  - Coordinates x and y.

- We consider the movement along the resulting trajectory from the starting point to the target. Let's take the direction of the robot at the point where the coordinate of the point where we are going to move forward is taken. Let us take as the initial - the direction in which the robot is at the starting point when the algorithm is run (x_1, y_1), where x, $\mathrm{y} = \{-1, 0, 1\}$.
- To find a direction from one point to the next along the path, the following information is necessary:

  - Coordinates of the current point.
  - The direction of the robot at the current point.
  - Coordinates of the next point on the path.

To move the robot you need to know the following values: the direction dir ($\mathrm{dir} = -1$ $-$ backward, $\mathrm{dir} = 0$ - there is no forward movement, $\mathrm{dir} = 1$ - forward), the direction of rotation turn ($\mathrm{turn} = -1$ - to the right, $\mathrm{turn} = 1$ to the left, $\mathrm{turn} = 0$ - without a turn), the angle of rotation φ (from 0 to 90 degrees).

We make two vectors:

- Vector $\vec{a}$ with the beginning at the current point, the end - in the coordinate corresponding to the direction of the robot at this point.
- Vector $\vec{b}$ with the beginning at the current point, the end - at the next point on the trajectory.

The angle between the vectors $\vec{a}$ and $\vec{b}$ is denoted by φ_0. We find this angle by the formula $\cos\varphi_0 = \frac{\vec{a} * \vec{b}}{|\vec{a}| * |\vec{b}|}$, so $\varphi_0 = \arccos\varphi_0$. If $\varphi_0 \leq 90°$, then for this point the state variables are $dir = 1$, and $\varphi = \varphi_0$, otherwise, $dir = -1$, and $\varphi = 180° - \varphi_0$. The direction of rotation is found using the vector product. To do this, we put the coordinate along the z axis of the vectors $\vec{a}$ and $\vec{b}$ equal to zero and consider the sign of the z component of the result of their vector product:

- If $x_a * y_b - y_a * x_b < 0$, then $turn = -1$.
- If $x_a * y_b - y_a * x_b > 0$, then $turn = 1$.

otherwise, $turn = 0$.

At each step, we do a check: if the next cell of this trajectory does not contain an obstacle, then move on, if it contains - stop moving and go to step 7, if it is the target - finish the algorithm.

If at least one of the neighboring cells k_{new} is available in the history list, then it is necessary to study them. The basic algorithm D * works according to the following principle: If at least one of the neighboring cells k_{new} is available in the history list and the updated weight of this cell is $m(k_{new}) \leq m(k) + \alpha$, where α is a coefficient that determines which of the criteria is more important: the length of the path or the running time of the algorithm (the optimal path length is $\alpha = 3$), then go to step 8, otherwise, make the current cell the initial position of the robot and go to paragraph 2. With this approach, a problem may arise when a robot, approaching some obstacles like walls, can make It is unnecessary and inefficient movement.

To solve this problem, it was decided to use the method of estimating points using a method similar to the method of potential fields [13-18].

For each point that satisfies the rule of the basic algorithm D * described above, a set of forces is generated, which are formed when the target point is represented as an attractor, and certain obstacles as a repeller. To determine the obstacles necessary for taking into account in this method, it is necessary to construct a circle whose radius is 3 cell lengths of a given grid, with the center at the intersection of the diagonals of the square forming the considered cell. Through this center we draw a line perpendicular to the vector directed to the target points. From the two semicircles formed by this line and a circle, choose one that partially or completely contains a vector directed to the target point. All cells with obstructions, completely or partially lying in this semicircle, are designated as repellers for the cell currently being considered. The force vector formed by the attractor is constant modulo - $|\overrightarrow{F_{attr}}| = const$ (the optimal choice for a constant is the length of the side of the square forming the cell), and the module of vectors formed by repellers is calculated using the formula $|\overrightarrow{F_i}| = 1/l_i^2$, where i is the number of the corresponding repeller, and l_i is the distance from the center of the investigated cell from the list to the center of the point of the i th repeller.

Find the resultant force vector

$$\overrightarrow{F_{result}} = \overrightarrow{F_{attr}} + \sum_i \overrightarrow{F_i} \qquad (1)$$

Build a vector equal to the resultant (1), with the beginning at the intersection of the diagonals of the current cell. If the distance from the center of the target cell to the end of this vector is less than the distance to its beginning, then moving to this point is permissible if the conditions of the basic algorithm D * specified at the beginning of the step are met.

If there are suitable points to move, go to step 8, otherwise, make the current cell the starting position of the robot and go to step 2.

- From the neighboring cells that satisfy the condition from clause 7, select the distance from the target point to which is the smallest, and proceed to step 9.
- With the help of information about the previous cell known for each cell, we move from each new point considered to the previous one until we reach the target point.

3 Algorithm Comparison

Robots controlled by a navigation system based on the D* algorithm do not need to use sensors such as gyroscope, accelerometer, gps, etc. This significantly reduces the cost of the robot and the time required to configure it [19-24].

The modified algorithm will make decisions faster by eliminating the need for additional calculations associated with receiving and using data from positioning sensors. However, the error of calculating the current location increases slightly, which can cause small deviations from the trajectory.

The results obtained by path length will be similar to the results of the original D* algorithm.

4 Conclusion

The method presented in the paper allows to solve some problems arising in the practical implementation of the navigation network construction method. It is a more efficient method for certain types of tasks. In addition, it is easier to learn, because it is presented not as a composite system, but as an algorithm that solves all the tasks assigned to it.

5 Acknowledgement

This work was supported by Russian Fondation for Basic Research, project № 18-08-00419.

6 References

[1] Koenig, S.; Likhachev, M. (2005),” Fast Replanning for Navigation in Unknown Terrain”, Transactions on Robotics, 21 (3): 354–363, CiteSeerX 10.1.1.65.5979, https://doi.org/10.1109/tro.2004.838026

[2] Koenig, S.; Likhachev, M.; Furcy, D. (2004),” Lifelong Planning A*”, Artificial Intelligence, 155 (1–2): 93–146, https://doi.org/10.1016/j.artint.2003.12.001

[3] Stentz, Anthony (1995),” The Focussed D* Algorithm for Real-Time Replanning”, Proceedings of the International Joint Conference on Artificial Intelligence: 1652–1659, CiteSeerX 10.1.1.41.8257

[4] Kulakov, F., Kadry, S., Alferov, G., Sharlay, A. Bilateral remote control over space manipulators (2018) AIP Conference Proceedings, 2040, P.150015 https://doi.org/10.1063/1.5079218

[5] Kulakov, F., Alferov, G., Sokolov, B., Gorovenko, P., Sharlay, A. Dynamic analysis of space robot remote control system (2018) AIP Conference Proceedings, 1959, P. 080014 https://doi.org/10.1063/1.5034731

[6] Alferov, G., Ivanov, G., Efimova, P., Sharlay, A. Study on the structure of limit invariant sets of stationary control systems with nonlinearity of hysteresis type (2017) AIP Conference Proceedings, 1863, P.080003. https://doi.org/10.1063/1.4992264

[7] Kulakov, F., Sokolov, B., Shalyto, A., Alferov, G. Robot master slave and supervisory control with large time delays of control signals and feedback (2016) Applied Mathematical Sciences, 10 (33-36), pp. 1783-1796. https://doi.org/10.12988/ams.2016.6380

[8] Kulakov, F., Alferov, G.V., Efimova, P., Chernakova, S., Shymanchuk, D. Modeling and control of robot manipulators with the constraints at the moving objects (2015) 2015 International Conference on” Stability and Control Processes” in Memory of V.I. Zubov, SCP 2015 - Proceedings, P. 7342075, pp. 102-105. https://doi.org/10.1109/scp.2015.7342075

[9] Kulakov,F., Kadry,S., Alferov,G., Efimova,P. Remote control of space robots change-adaptive in its external environment, International Journal of Online and Biomedical Engineering, Vol.15,N7, 2019. https://doi.org/10.3991/ijoe.v15i07.10219

[10] Kulakov, F., Alferov, G., Efimova, P. Methods of remote control over space robots (2015) 2015 International Conference on Mechanics - Seventh Polyakhov’s Reading, P.7106742, https://doi.org/10.1109/polyakhov.2015.7106742

[11] Kadry, S., Alferov, G., Ivanov, G., Korolev V. and Selitskaya E. A New Method to Study the Periodic Solutions of the Ordinary Differential Equations Using Functional Analysis. Mathematics 2019, 7(8), 677; https://doi.org/10.3390/math7080677 (registering DOI) https://doi.org/10.3390/math7080677

[12] Ivanov G., Alferov G., Efimova P. Integrability of nonsmooth one-variable functions (2017) 2017 Constructive Nonsmooth Analysis and Related Topics (Dedicated to the Memory of V.F. Demyanov), CNSA 2017 – Proceedings, P. 7973965, https://doi.org/10.1109/cnsa.2017.7973965

[13] S.Kadry, G.Alferov, A.Kondratyuk, V.Kurochkin, S. Zhao. Modeling the motion of a space manipulation robot using position control. (2019) AIP Conference Proceedings, 2116, P.080005. https://doi.org/10.1063/1.5114065

[14] Alferov G.V.,Ivanov G.G., Efimova P.A.. The structural study of limited invariant sets of relay stabilized system (Book Chapter) (2017) Mechanical Systems: Research, Applications and Technology, pp.101-164

[15] Kadry, S., Alferov, G., Ivanov, G., Sharlay, A. About stability of selector linear differential inclusions (2018) AIP Conference Proceedings, 2040, P. 150013, https://doi.org/10.1063/1.5079216

[16] Kadry, S., Alferov, G., Ivanov, G., Sharlay, A. Stabilization of the program motion of control object with elastically connected elements (2018) AIP Conference Proceedings, 2040, P. 150014 https://doi.org/10.1063/1.5079217

[17] Alferov, G.V., Ivanov, G.G., Efimova, P.A., Sharlay, A.S. Stability of linear systems with multitask right-hand member (2018) Stochastic Methods for Estimation and Problem Solving in Engineering, pp. 74-112.Book Chapter https://doi.org/10.4018/978-1-5225-5045-7.ch004

[18] Ivanov, G., Alferov, G., Sharlay, A., Efimova, P. Conditions of asymptotic stability for linear homogeneous switched systems (2017) AIP Conference Proceedings, 1863, P. 080002, https://doi.org/10.1063/1.4992263

[19] Alferov, G.V., Malafeyev, O.A. The robot control strategy in a domain with dynamical obstacles (1996) Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 1093, pp. 211-217. https://doi.org/10.1007/bfb0013961

[20] Ivanov, G., Alferov, G., Gorovenko, P., Sharlay, A. Estimation of periodic solutions number of first-order differential equations (2018) AIP Conference Proceedings, 1959, P. 080006, . https://doi.org/10.1063/1.5034723

[21] Gennady Alferov, Gennady Ivanov, Artem Sharlay, and Viktor Fedorov. Application of derived numbers theory in problem of function extremum (2019) AIP Conference Proceedings, 2116, 080002; https://doi.org/10.1063/1.5114062.

[22] Gennady Alferov, Gennady Ivanov, Artem Sharlay, and Viktor Fedorov. Application of derived numbers theory in convex analysis (2019) AIP Conference Proceedings, 2116, 080003; https://doi.org/10.1063/1.5114063.

[23] Gennady Alferov, Gennady Ivanov, Artem Sharlay, and Viktor Fedorov. Estimation for number of almost periodic solutions of first-order ordinary differential equations (2019) AIP Conference Proceedings, 2116, 080004; https://doi.org/10.1063/1.5114064.

[24] S.Kadry, G.Alferov, G.Ivanov, V.Korolev, A.Sharlay. Almost periodic solutions of first-order ordinary differential equations (2018) Mathematics, 6 (9), P.171. https://doi.org/10.3390/math6090171

7 Authors

Seifedine Kadry is a Professor of Data Science in Beirut Arab University, Beirut, Lebanon. Email: s.kadry@bau.edu.lb

Gennady Alferov works at St. Petersburg State University, St. Petersburg, Russia. Email: alferovgv@gmail.com

Viktor Fedorov works at St. Petersburg State University, St. Petersburg, Russia. Email: victor-fyodorov@ya.ru

Article submitted 2020-03-15. Resubmitted 2020-04-06. Final acceptance 2020-04-06. Final version published as submitted by the authors.