Smoothed A-star Algorithm for Nonholonomic Mobile Robot Path Planning

Syaiful A. Gunawan*, Gilang N. P. Pratama†, Adha I. Cahyadi*, Bondhan Winduratna*,
Yohannes C. H. Yuwono*, Oyas Wahyunggoro*
*Department of Electrical Engineering and Information Technology
Faculty of Engineering, Universitas Gadjah Mada
Yogyakarta, Indonesia
†Department of Electrical Engineering
Sekolah Tinggi Teknologi Adisutjipto
Yogyakarta, Indonesia

***Abstract*—There are various path planning methods for mobile robots and one of them is A-star algorithm. It serves fairly well for looking the path accurately. Unfortunately, the conventional A-star algorithm may have some sharp turns. It causes problems for nonholonomic mobile robots to track the path. In this paper, we proposed a scheme to make A-star algorithm smoother. The smoothed path is easier for nonholonomic mobile robots to be tracked. We conduct simulation in the virtual environment that covers the physical behavior of mobile robot. Simulation and experiment verify that the smoothed A-star algorithm can be implemented for nonholonomic mobile robots.**

***Keywords*—Nonholonomic Mobile Robot, Path Planning, Smoothed A-star Algorithm.**

I. Introduction

Over past decades, mobile robots have been implemented not only for industrial, military, and academic purposes but also in social applications [1]–[3]. Mobile robots are widely employed to help our life more convenient [4]–[7]. One interesting topic regarding mobile robots is path planning. We can define path planning as a scheme to find a collision-free route from the initial to the desired states while considering several conditions such like the distance, consumption of energy, communication delay, and computation time [8]–[10]. In simple words, it offers a strategy for mobile robots to reach the desired coordinate.

Several methods have been introduced such as visibility graph [11], triangulation of the environment [12], and graph-based path planning [13]. Due to its numerous algorithms, it can be said that the graph-based method is the most popular one. Some notable examples of graph-based path planning are Dijkstra and A-star algorithm. The last one can be said as an improvement of Dijkstra algorithm [14]. It can solve the minimum cost path through a graph accurately [15]. Thus, the optimal path is achieved.

Previously, the mobile robot has been modeled as a point of mass seems by [6] and [7]. It is obvious that the model is oversimplified by ignoring the dynamic and kinematic constraints. In real-world experiment, we can not implement the path planning strategy without taking the physical structure of the mobile robot into account.

The problems in path planning are not only lying on to find the path but also to make sure its visibility. In spite of A-star algorithm may give the optimal path, it does not mean that the mobile robots can follow, particularly for nonholonomic one. It has constraints in its movements which makes difficult to take sharp turns [16]–[18].

In this paper, there are two main contributions. First, we modify the conventional A-star algorithm into a smoothed one. It can guarantee fewer sharp turns, which is more convenient for nonholonomic mobile robots. The second contribution is implementing the smoothed A-star algorithm in V-REP, the virtual environment for robotics applications. It is a powerful platform that can emulate the physical world, including the kinematic and dynamic of robots [19]–[21]. Moreover, the mobile robot is not modeled as a point of mass, just like in [6] and [7].

The rests of this paper are organized as follows. Brief explanation of nonholonomic mobile robot is exposed in section II. Later, the smoothed A-star strategy can be found in section III. Simulation, results, and its analysis are presented in section IV. Lastly, the summary of this paper is elaborated in section V.

II. Nonholonomic Mobile Robot

Before any further, we need to distinguish the definition of the holonomic and nonholonomic mobile robot. It is called holonomic if all of the physical constraints can be integrated into positional constraints as

$$f\{q_1, q_2, \ldots, q_n; t\} = 0, \tag{1}$$

where the terms q_i are its coordinates.

In simple words, if the number of degrees of freedom (DoF) which controllable equals to its DoF, thus mobile robot is holonomic. Since all of its DoF are controllable, we can tell that the mobile robot does not have any kinematic constraint [22]. On the contrary, any system with kinematic constraints is considered as nonholonomic.

III. Path Planning Strategy

In this part, we discuss the general A-star algorithm and the method to make it smoother. Brief explanation of the conventional A-star algorithm can be found at subsection III-A. Meanwhile, subsection III-B presents the smoothing scheme for path planning.

A. A-star Algorithm

There are various problems in the navigation of a mobile robot [23]. One problem that has to be faced is the partial knowledge of the environment and its uncertainties. Things getting more complicated if the environment is vast and dynamics. Thus, it is necessary to employ a subtle approach to overcome those problems.

We can handle the problem by introducing a scheme for path planning. The A-star algorithm is one of the most popular methods for determining the optimal path of mobile robots. Due to its reliability and the fact that it can be applied in various systems, not only in robotics, the A-star algorithm has been deployed by many researchers.

The basic idea of the A-star algorithm is creating a specific node of a graph and build a path from it. Thus, the path extends until it reaches the predetermined position. Nonetheless, the conventional A-star algorithm still far from perfect. In our case, it does not consider the kinematic constraints of the mobile robot, which is still quite vulnerable to crashing nearby obstacles. Ever since the modification of A-star is very likely to overcome things like this.

In general, the conventional A-star algorithm chooses the path that minimizes

$$f(i) = g(i) + h(i), \tag{2}$$

where i, $g(i)$, and $h(i)$ are the next position of robot on the path, the current cost, and heuristic function respectively. The current cost $g(i)$ is calculated from the start to the node i. Meanwhile, $h(i)$ estimates the lowest cost from i to the designated goal.

B. Smoothing Scheme

Unfortunately, the generated path by A-star algorithm may contain sharp turn, which is difficult for nonholonomic mobile robot. Here, we propose a scheme to make it smoother. The main idea is to minimize these conditions.

- The distance between two points.
- The distance that the point gets away from the previous position.

Suppose we have the original path as ν and the smoothed one as μ, then the previous conditions can be denoted as

$$\begin{aligned} d_1 &= (\mu_i - \nu_i)^2, \\ d_2 &= (\mu_i - \mu_{i+1})^2, \end{aligned} \tag{3}$$

where d_1 is the deviation of the original and smoothed path. Meanwhile, d_2 is the deviation from one node to the next one. As stated before, we need to minimize d_1 and d_2 in order to accomplish a smooth path.

Our aim is to make a smooth path between the starting point to the designated goal. It implies that those points are not necessary to be changed. We need to retain them as it is. Hence, we have the starting point $\mu_0 = \nu_0$ and the designated goal $\mu_{n-1} = \nu_{n-1}$

It is mandatory to minimize those conditions, but zero values need to be avoided. Let say, if $(\nu_i - \mu_i)^2 = 0$, then we have the original path. There is no smoothing applied. Meanwhile, if $(\mu_i - \mu_{i+1})^2 = 0$, then we stuck with no path. It is obvious that the objective functions are conflicted with each other. We can handle it by weighting its movement as

$$\begin{aligned} d_1 &= (\mu_i - \nu_i)^2, \\ d_2 &= \gamma (\mu_i - \mu_{i+1})^2, \end{aligned} \tag{4}$$

where the weighting, $\gamma \in \mathbb{R}_+$, is a possitive real number.

After elaborating the necessities, we need to optimize those functions. The gradient descent can be utilized to optimize it by taking a small step in the direction of minimizing the error for each iteration. Hence, for the first term, we have

$$\begin{aligned} \mu_i^1 &= \mu_i - \alpha \nabla d_1, \\ &= \mu_i - \alpha (\nu_i - \mu_i), \end{aligned} \tag{5}$$

where μ_i^1 and α are the update of the first term and the weighting data. Here, any changes are proportional to the weighted deviation of μ_i to ν_i. Meanwhile, the second one is given as

$$\begin{aligned} \mu_i^2 &= \mu_i - \beta \nabla d_2, \\ &= \mu_i - \beta (\mu_i - \mu_{i+1}). \end{aligned} \tag{6}$$

where β is the weight smoothing.

Furthermore, we can consider the deviation from the previous state, μ_{i-1} to μ_i. Hence, we have

$$\mu_i^2 = \mu_i - \beta (2\mu_i - \mu_{i+1} - \mu_{i-1}). \tag{7}$$

The smoother path can be achieved by optimizing those conditions defined in (5) and (7). The illustration of the smoothed path can be seen at Figure 1.

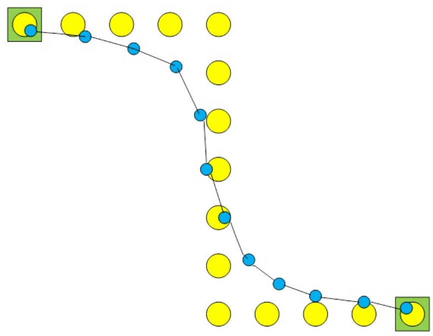

Fig. 1: The smoothed path.

Based on Figure 1, the yellow and blue nodes represent the original and smoothed path. There is an interesting fact if we have $\alpha = 0$, the path will be a direct straight line from the starting point to its goal. On the other hand, we should not accommodate $\beta = 0$, which implies the smoothing is void.

IV. Simulation and Analysis

In this section, we expose how the simulation is conducted, simple kinematics of mobile robot, and the results. First things first, let us explain the experimental setup to this simulation in subsection IV-A. In subsection IV-B, the kinematics of mobile robot is explained briefly. Lastly, the results and analysis are presented in subsection IV-C

A. Experimental Setup

Here we utilize Linux-based computer, V-REP to simulate the environment of mobile robot, while the smoothing scheme is implemented in Python as illustrated in Figure 2.

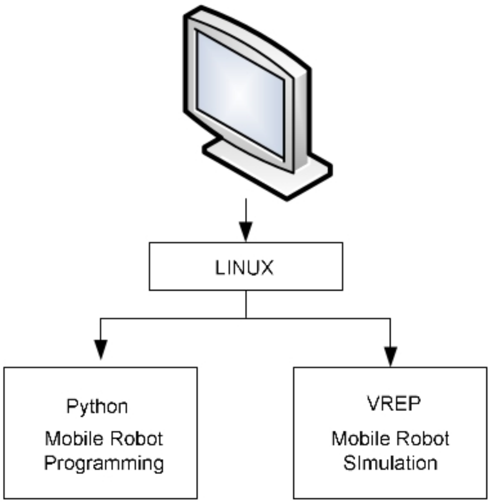

Fig. 2: The experimental setup.

It is more convenient to write the code in Python rather than other programming languages. It can be accomplished since V-REP provides API to convey another programming languages including Python. Moreover, V-REP allows us to design the virtual environment for the robots. Here, we design a maze with narrow-path for mobile robot and the goal as presented in Figure 3.

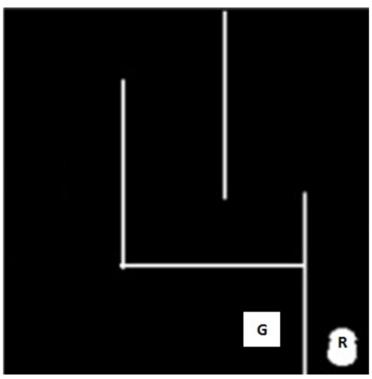

Fig. 3: The maze-based map with narrow-path.

B. Kinematics Model

The mobile robot only has two control inputs, the right and left motor. In addition, it has 3 DoF, they are the orientation (ψ) and position in $x-$ and $y-$axis. The mathematical model of mobile robot can be denoted as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{u} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} u\cos\psi - a\omega\sin\psi \\ u\sin\psi - a\omega\cos\psi \\ \omega \\ \frac{\theta_3}{\theta_1}\omega^2 - \frac{\theta_4}{\theta_1}u \\ -\frac{\theta_5}{\theta_2}u\omega - \frac{\theta_6}{\theta_2}\omega \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{u_{ref}}{\theta_1} \\ \frac{\omega_{ref}}{\theta_2} \end{bmatrix} + \begin{bmatrix} \delta_x \\ \delta_y \\ 0 \\ \delta_u \\ \delta_\omega \end{bmatrix}, \tag{8}$$

where u and ω are the linear and angular velocity with its desired values u_{ref} and ω_{ref}. The detail of parameters in equation (8) can be seen at Figure 4 and [24].

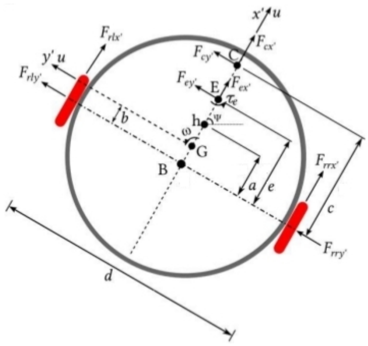

Fig. 4: The model of mobile robot.

Based on Figure 4, we can say that the front face of mobile robot is in $x-$axis, while its left and right side are along $y-$axis.

C. Results and Analysis

In order to verify that the proposed smooting scheme is applicable, we conduct simulation in V-REP. Here we use $\alpha = 0.5$ and $\beta = 0.1$. The mobile robot needs to reach the goal at coordinate $\mathbf{G}$ from the starting point $\mathbf{S}$, where

$$\begin{aligned} \mathbf{S} &= \begin{bmatrix} x_s \\ y_s \end{bmatrix} = \begin{bmatrix} 1.43018 \\ -0.037768 \end{bmatrix}, \\ \mathbf{G} &= \begin{bmatrix} x_g \\ y_g \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \end{aligned}$$

The result, when the mobile robot reaches the designated goal, can be seen at Figure 5.

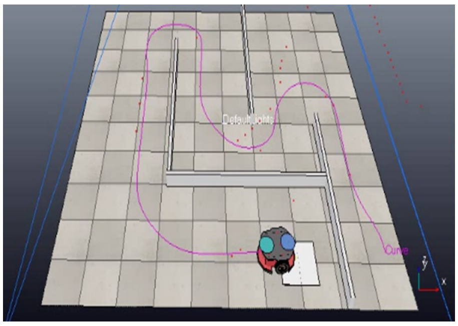

Fig. 5: The mobile robot reaches the goal.

Based on Figure 5, the goal is the white rectangular on the other side of the mobile robot. There are four walls that obstruct the mobile robot to the goal. Despite going through the narrow-path, the mobile robot succeeds in finding a way to reach the goal. In order to make sure that it does not move into the wall, we purposely set higher cost near it. The path that the mobile robot going through is presented in Figure 6.

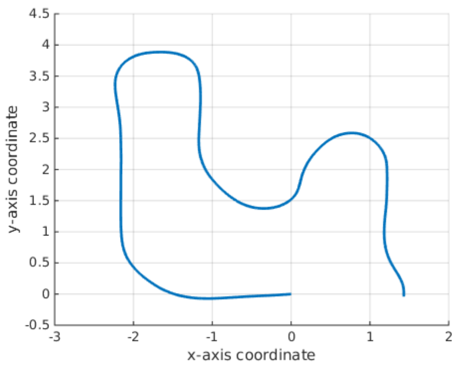

Fig. 6: The path of mobile robot.

The changes in its position along $x-$ and $y-$axis over the time can be seen in Figure 7 and 8.

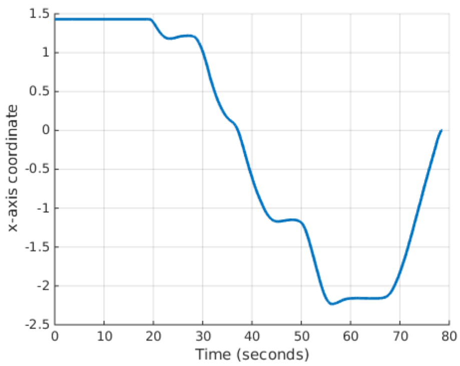

Fig. 7: The coordinate position of the mobile robot along $x-$axis over the time.

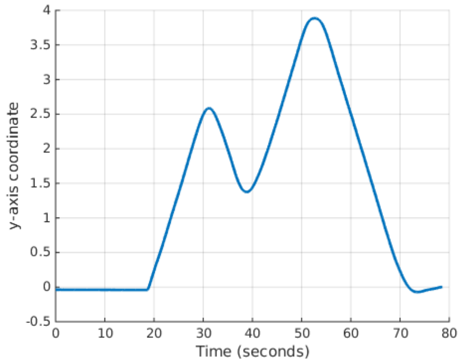

Fig. 8: The coordinate position of the mobile robot along $y-$axis over the time.

Obviously from Figure 7 and 8, we can see that in the first 20 seconds of the simulation, the mobile robot stands still. It can be said that it does not move at all from its initial position. It is due to the fact in the first 20 seconds is needed to determine the path. The computational time itself is not settled, but it depends on the capability of the computer. Thus it may differ to others.

The mobile robot needs around 60 seconds to reach the goal. Hence, the overall time is around 80 seconds. It should be noted that the time is in simulation time, not in real-world time. It means the actual time is longer.

We can explore further the analyze of the behavior of the mobile robot by sampling the data. During the mobile robot turns around, it has to slow down. On the other hand, the mobile robot can move faster in the straight path as shown in Figure 5.

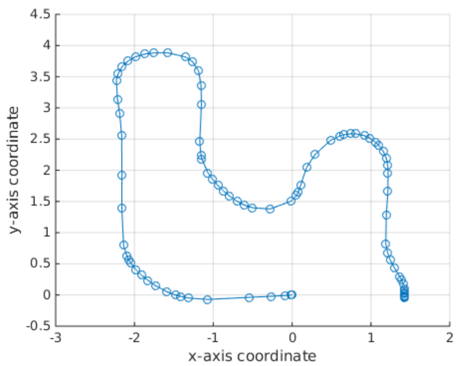

Fig. 9: Velocity analysis of mobile robot.

The closer one point to others means it slack of velocity and vice versa. The points are dense in the beginning since the mobile robot does not move at all. Those points also inform us how the mobile robot is slowing down before stopping at the designated goal.

V. Conclusion

The conventional A-star algorithm can provide the optimum path for the mobile robot. Nonetheless, it does not consider the availability, as an example the path may contain sharp turns. The sharp turns may cause problem for the nonholonomic mobile robot to going through. This paper presents a scheme to make the A-star algorithm smoother by fulfilling the requirements. Furthermore, the smoothed A-star algorithm then being implemented in the virtual environment for nonholonomic mobile robot path planning. Despite being deployed in a maze with narrow-path, the mobile robot succeeds to reach its designated goal. It verifies that the smoothed A-star algorithm is applicable for nonholonomic mobile robot.

Acknowledgment

We would like to thank for research funding from Rekognisi Tugas Akhir (RTA) program of Universitas Gadjah Mada. We are thankful to Direktorat Penelitian Universitas Gadjah Mada for providing guidance and support concerning this final project.

References

[1] Y. Umeda and T. Yakoh, "Configuration and readhesion control for a mobile robot with external sensors," *IEEE Transactions on Industrial Electronics*, vol. 49, no. 1, pp. 241–247, Feb 2002.

[2] H. Takahashi, H. Nishi, and K. Ohnishi, "Autonomous decentralized control for formation of multiple mobile robots considering ability of robot," *IEEE Transactions on Industrial Electronics*, vol. 51, no. 6, pp. 1272–1279, Dec 2004.

[3] N. Uchiyama, T. Hashimoto, S. Sano, and S. Takagi, "Model-reference control approach to obstacle avoidance for a human-operated mobile robot," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 3892–3896, Oct 2009.

[4] A. Gorbenko, M. Mornev, and V. Popov, "Planning a typical working day for indoor service robots," *IAENG International Journal of Computer Science*, vol. 38, no. 3, pp. 176–182, 2011.

[5] Y. Hiroi and A. Ito, "Influence of the height of a robot on comfortableness of verbal interaction," *IAENG International Journal of Computer Science*, vol. 43, no. 4, pp. 447–455, 2016.

[6] R. C. Hidayat, A. R. Rafsanzani, O. Wahyunggoro, and A. I. Cahyadi, "Local arrival time field based path planning using guided waypoints for unknown environment," in *2018 3rd International Conference on Information Technology, Information System and Electrical Engineering (ICITISEE)*, Nov 2018, pp. 325–329.

[7] A. R. Rafsanzani, R. C. Hidayat, A. I. Cahyadi, and S. Herdjunanto, "Omnidirectional sensing for escaping local minimum on potential field mobile robot path planning in corridors environment," in *2018 3rd International Seminar on Sensors, Instrumentation, Measurement ans Metrology (ISSIMM)*, Dec 2018, pp. 79–83.

[8] M. Mendes, A. P. Coimbra, and M. M. Crisostomo, "Assessing the performance of sdm-based robot navigation with different image processing techniques," *IAENG International Journal of Computer Science*, vol. 39, no. 4, pp. 349–356, 2012.

[9] A. Gorbenko and V. Popov, "Visual landmark selection for mobile robot navigation," *IAENG International Journal of Computer Science*, vol. 40, no. 3, pp. 134–142, 2013.

[10] A. M. Sakti, A. I. Cahyadi, and I. Ardiyanto, "Path planning and path following using arrival time field for nonholonomic mobile robot," in *2017 International Conference on Advanced Computing and Applications (ACOMP)*, Nov 2017, pp. 143–148.

[11] D. Wooden and M. Egerstedt, "Oriented visibility graphs: low-complexity planning in real-time environments," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*, May 2006, pp. 2354–2359.

[12] Y. Tarutoko, K. Kobayashi, and K. Watanabe, "Topological map generation based on delaunay triangulation for mobile robot," in *2006 SICE-ICASE International Joint Conference*, Oct 2006, pp. 492–496.

[13] S. Benders and S. Schopferer, "A line-graph path planner for performance constrained fixed-wing uavs in wind fields," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2017, pp. 79–86.

[14] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, Dec 1959.

[15] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, July 1968.

[16] T. Yamamoto and K. Watanabe, "A switching control method for stabilizing a nonholonomic mobile robot using invariant manifold method," in *Proceedings of SICE Annual Conference 2010*, Aug 2010, pp. 3278–3284.

[17] K. Izumi, H. Tanaka, and T. Tsujimura, "Nonholonomic control considering with input saturation for a mobile robot," in *2016 55th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, Sep 2016, pp. 1173–1178.

[18] T. R. Schafle, A. Tokui, and N. Uchiyama, "A hybrid systems approach with input-output linearization for automotive parking control of a nonholonomic mobile robot," in *2018 3rd International Conference on Advanced Robotics and Mechatronics (ICARM)*, July 2018, pp. 1–6.

[19] M. Freese, S. Singh, F. Ozaki, and N. Matsuhira, "Virtual robot experimentation platform v-rep: A versatile 3d robot simulator," in *Simulation, Modeling, and Programming for Autonomous Robots*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 51–62.

[20] E. Rohmer, S. P. N. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 1321–1326.

[21] V. Sharma, S. Yildirim-Yayilgan, and L. V. Gool, "Low-cost scene modeling using a density function improves segmentation performance," in *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Aug 2016, pp. 77–84.

[22] H. H. Triharminto, O. Wahyunggoro, T. B. Adji, and A. I. Cahyadi, "An integrated artificial potential field path planning with kinematic control for nonholonomic mobile robot," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 6, no. 4, pp. 410–418, 2016.

[23] I. Ardiyanto, "Task oriented behavior-based state-adaptive pid (proportional integral derivative) control for low-cost mobile robot," in *2010 Second International Conference on Computer Engineering and Applications*, vol. 1, March 2010, pp. 103–107.

[24] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*. Scituate, MA, USA: Bradford Company, 2004.