2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)
Sheraton Arabella Park Hotel, Munich, Germany, July 3-7, 2017

Improved Follow the Gap Method for Obstacle Avoidance

Mustafa Demir[1] and Volkan Sezer[2]

***Abstract*— Follow the Gap Method (FGM) is a safety-focused geometric obstacle avoidance algorithm for local navigation. In this method, the largest gap around the robot is selected and the robot moves towards a goal point considering the largest gap and minimum distance to obstacle. One of the drawbacks of the method is the extension of the path which sometimes happens unnecessarily. Another drawback comes from small differences between the gap sizes. This sometimes makes the robot to change the selected gaps instantly which causes zigzag trajectories. In this paper, Improved Follow the Gap Method (FGM-I) is presented to eliminate these two drawbacks. In FGM-I, the gaps are selected according to a new utility function and unnecessary gap changes are penalized. A differential drive powerchair is used in simulations. The simulation results show that the FGM-I reduces the effects of the mentioned problems seriously.**

I. INTRODUCTION

Robot navigation is a significant topic for reaching a given target autonomously without collision. Different methods are used in robot navigation depending on whether the map is known or unknown and whether the obstacles are dynamic or static. If the map is known and the obstacles are static, classical path planning methods work perfectly to reach a target position. The trajectory is calculated from the actual point of the robot to a given target. Rapidly Exploring Random Trees (RRTs) [1], Probabilistic Roadmaps (PRMs) [2], cell decomposition methods [3] and search-based methods [4], [5] are among the most common path planning methods.

Obstacle avoidance methods are utilized to reach a target position, if the environment is unknown and/or the obstacles are dynamic. While the robot moves towards the target, the main aim of the obstacle avoidance methods is to avoid from obstacles dynamically. Various methods have been put forward for obstacle avoidance. In bug algorithms [6] the robot starts moving to the target directly using the shortest path, until the it encounters an obstacle. After this, the robot moves around the obstacle, until the obstacle is not between the robot and the target. The paths that are being followed by the robot in bug algorithms are sometimes very long and unsafe.

Artificial potential field (APF) [7] is another commonly used obstacle avoidance method. In APF, the target position is defined as artificial attractive potential field while the obstacles are defined as artificial repulsive potential fields. Both of them affect the robot. The attractive potential field of the target generates a force pulling the robot towards the target and the repulsive potential fields of obstacles generate forces which push away the robot from them. Through these repulsive and attractive forces, the robot is able to avoid obstacles as it moves towards the target. However, the local minima [8] is a very critical problem of this method. When the total forces on the robot are zero, the robot loses its ability to move. There are some studies [8], [9] aiming to solve the local minima problem of APF method.

Another important method used for obstacle avoidance is the Vector Field Histogram (VFH) [10]. In VFH, a two-dimensional Cartesian histogram grid is utilized to show the obstacles. Each cell in the histogram grids has information about whether there is an obstacle or not. After that, a one-dimensional polar histogram is created around the current location of the robot. The feasible field which has low polar obstacle density is selected from between all polar histogram fields and the robot starts to travel using the selected field. There are some studies implemented to improve this method which can be found in [11], [12].

Dynamic Window Approach (DWA) [13] is a velocity-based obstacle avoidance method that determines the acceptable collision-free velocity set for the robot in order to reach the target by maximizing its objective function. The optimum velocity pair depends on the final heading angle, the minimum distance to the obstacles and the admissible velocity values. Some other studies for the improvement of the DWA can be found in [14], [15].

Follow the Gap method (FGM) [16], is a safety-focused geometric obstacle avoidance algorithm. In this method, the largest gap's center angle, the goal angle and the minimum distances to obstacles are used to avoid from obstacles, while the robot moves towards the goal point. Intelligent Follow the Gap Method [17] is proposed to solve the dead-end scenario by combining the FGM with Intelligent Bug Algorithm [18]. Besides these methods, Nearness Diagram (ND) [19], Obstacle Restriction Method (ORM) [20] and search-based methods [21]–[23] are used for obstacle avoidance.

This paper is based on the improvement of FGM. In FGM algorithm, two drawbacks are observed. First, selecting the largest gap may cause the path to be long. Second, small differences between the gap sizes cause to change the selected gap instantly which causes zigzag trajectories. In this paper, an extension of FGM which is named as improved FGM or FGM-I is presented to improve the gap selection of the FGM and to prevent instant changes of the selected gap.

*This work was supported by Turkish Scientific and Technological Research Council (TUBITAK) under project no 215E140.

[1]Mustafa Demir is with Graduate School of Science Engineering and Technology, Mechatronics Engineering Department, Istanbul Technical University, Istanbul, TURKEY demirmust@itu.edu.tr

[2]Volkan Sezer is with Control and Automation Engineering Department, Istanbul Technical University, Istanbul, TURKEY sezerv@itu.edu.tr

A utility function is added to the gap selection process of the FGM which shortens the trajectory without significant lose of safety. In addition to the utility function, zigzag trajectories are eliminated by penalizing the unnecessary largest gap decision.

Outline of the paper is as follows: Section II explains the FGM and the drawbacks of it. Section III presents the FGM-I to eliminate the mentioned problems. After this, the simulation results are illustrated in Section IV and finally the conclusions are given in Section V.

II. FOLLOW THE GAP METHOD

The FGM is a geometric obstacle avoidance algorithm in which the safety is the most important factor. The FGM determines the final heading angle using the largest gap center angle, the goal angle and the minimum distances to the obstacles. A safety coefficient used in the method which affects the distance between the robot and the obstacles during the trajectory. While calculating the final heading angle, firstly, the gaps in the field of view are determined using the border angle value of the obstacles [16]. Then the largest gap is calculated from border angle values.

In the second step, the angle, which is in between the midpoint of the largest gap and the robot orientation, is calculated. After that, the goal angle, which is between the goal point and the robot orientation, is determined. In the last step, the final heading angle is calculated using (1). The final heading angle, the gap center angle, the goal point angle, the minimum distance to the obstacles and the safety coefficient are represented as ϕ_{final} , ϕ_{gap_c}, ϕ_{goal}, d_{min} and α, respectively. The effect of the safety coefficient can be found in [16].

$$\phi_{final} = \frac{\frac{\alpha}{d_{min}}\phi_{gap\_c} + \phi_{goal}}{\frac{\alpha}{d_{min}} + 1} \tag{1}$$

The gaps, midpoint of the largest gap, the goal point and the final heading angle are shown in Fig. 1. After the calculation of the final heading angle, the robot moves towards the target without a collision. A simulation result that shows the robot reaching the target using the FGM is given in Fig. 2. As can be seen in Fig. 2, this method generates feasible trajectories which is followed by the robot. In simulations, the safety coefficient is selected as 5. Simulation details are given in Section IV.

Although the selection of the largest gap yields better results in terms of safety, sometimes the trajectory could be longer than necessary. In addition, when the differences between gap size values are too small, the robot switch from one gap to another and this sometimes causes traveling too close or even hit the obstacles during the movement due to change in gap selection caused by the small differences between the gap sizes. These drawbacks are explained in Section II-A and Section II-B.

A. Trajectory Length Problem of FGM

In some obstacle and goal point configurations, the selection of the largest gap used by the FGM causes the path to be elongated, even the robot is able to reach the target from a shorter and safe path. This kind of scenario is shown in Fig. 3. In Fig. 3a the robot could choose going between line and rectangle obstacles while in Fig. 3b the trajectory could be through the rectangle and bottom line obstacles. These would result shorter and safe trajectories. Selecting the largest gap causes the trajectory longer than necessary as it is seen.

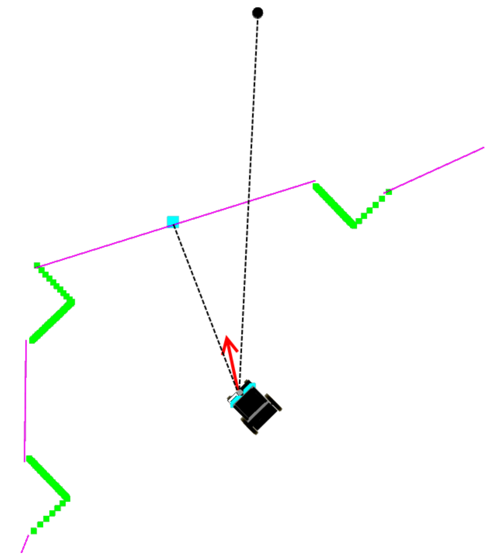


Fig. 1: Visualization of critical results in FGM . (The gaps are illustrated as purple lines. Light blue square indicates the midpoint of the largest gap. The goal point and the final heading angle are shown as a black point and a red arrow respectively.)

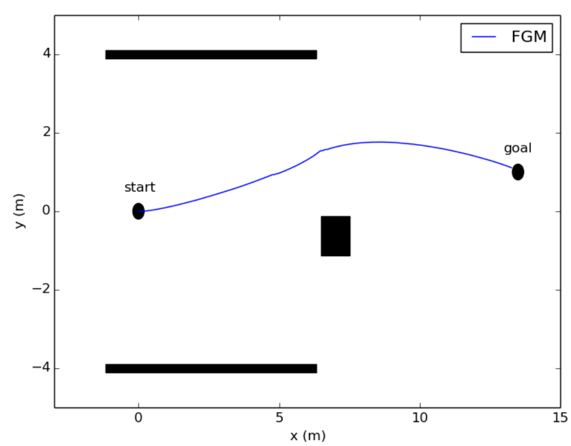
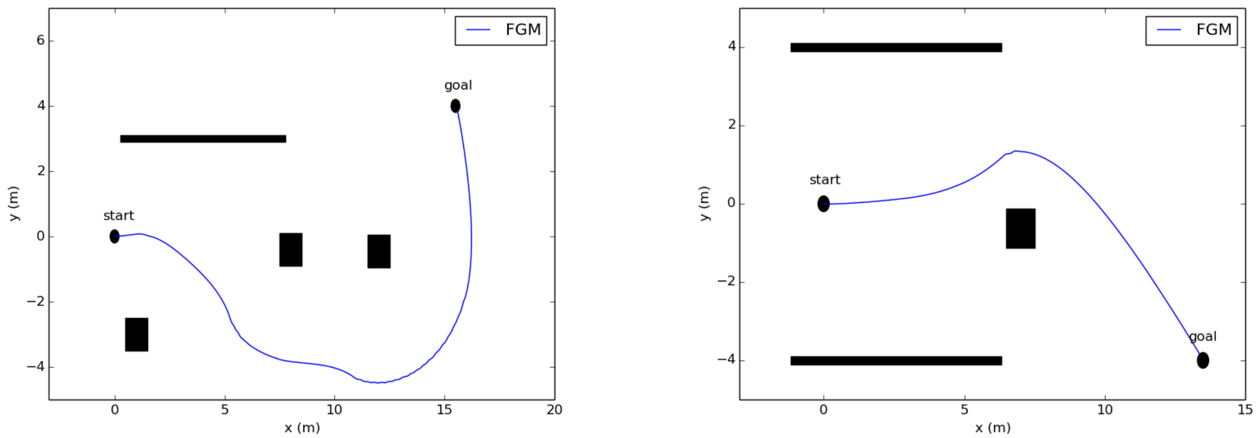


Fig. 2: FGM - Simulated trajectory sample

B. Zigzag Problem of FGM

In FGM, small differences between the gap sizes cause the selected gap to change instantly. Therefore, the chosen gap changes continually. Due to these changes, the robot may move towards the obstacle. The small differences between



(a) Configuration I

(b) Configuration II

Fig. 3: Simulation results of the FGM for trajectory length condition

the gap sizes cause an unstable trajectory and the robot may travel too close or even hit the obstacles during the movement. The zigzag problem of the FGM is shown in Fig. 4 where the gap size values in left and right hand side of the robot are very close to each other.

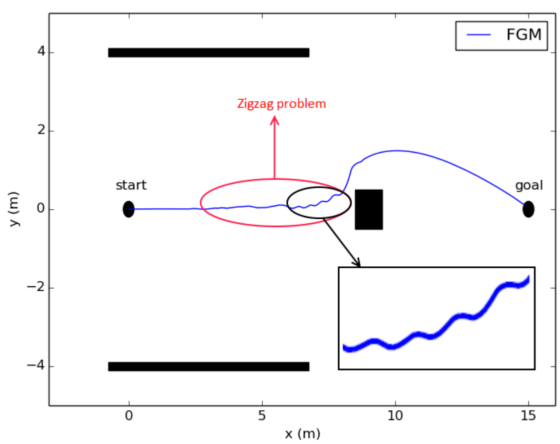


Fig. 4: Simulation scenario for the zigzag problem of FGM

III. IMPROVED FOLLOW THE GAP METHOD (FGM-I)

The FGM is constructed on the gap arrays to calculate heading angle for the robot. During the movement, choosing the largest gap reveals some problems as illustrated in Section II. In our new approach, the gaps are selected according to a new utility function and unnecessary gap changes are penalized for zigzag problem. These two modifications of FGM-I are explained in Section III-A and Section III-B in order to eliminate the FGM drawbacks.

A. Selection of the Gap Using a Utility Function

The utility function for each gap is illustrated in (2) which determines the new gap selection. It consists of two variables; the first one is the size of the gaps d_{gap_n} while the second one is the angle between the gap center and the goal point $\phi_{gap\_n\_to\_goal}$ as shown in Fig. 5. By this way, the gap selection is not only done by its size as in FGM but also its heading to the goal coordinate.

$$U_n = k_1 d_{gap\_n} + k_2(\pi - \phi_{gap\_n\_to\_goal}) \tag{2}$$

(d_{gap_n}: Size of the nth gap. k_1: Weight coefficient for the gap size. $\phi_{gap\_n\_to\_goal}$: Angle between nth gap center and goal point. k_2: Weight coefficient for angle.)

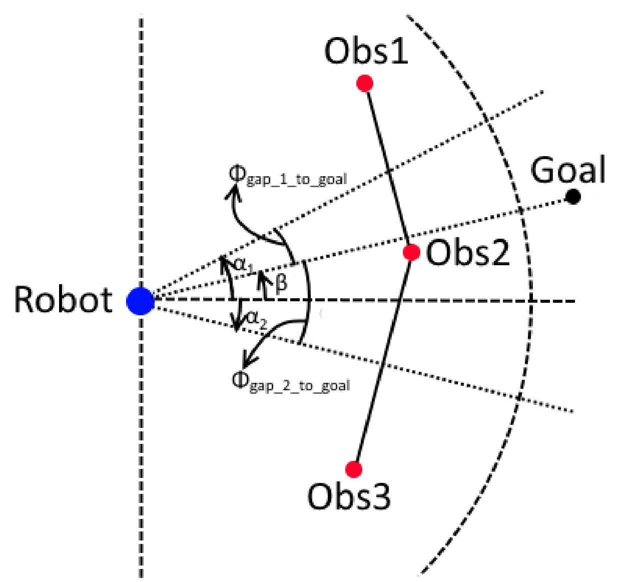


Fig. 5: Representation of the angles which are included within the second parameter of the utility function

The calculation of the $\phi_{gap\_n\_to\_goal}$ values for the scenario shown in Fig. 5 are given in (3).

$$\phi_{gap\_n\_to\_goal} = |\alpha_n - \beta| \tag{3}$$

In classical FGM, gap selection depends only on the gap size d_{gap_n}. In FGM-I the gap size is still very important for the decision but now it is not the only factor. Besides the gap size, the angle between gap center and goal point is an additional decision parameter. This means that the gaps in same direction to goal point from robot's perspective are rewarded more than the others. This brings an advantage on the gaps which results with shorter trajectories.

In addition, the coefficients determine how much each parameter affects the utility function. These coefficients can be chosen to give more importance on either the gap size or the angle between the gaps and the goal point. If safety is a priority for the robot movement towards the target, k_1 should be increased and/or k_2 should be decreased. If the trajectory length is a priority, the previous strategy should be reversed. The utility function is calculated for each gap and the desired gap is selected according to the maximum value of the utility function. The effect of the different coefficients on the same obstacle scenario is shown in Fig. 6.

In rest of the paper, k_1 and k_2 values are selected as 0.4 and 0.6 for FGM-I. The comparison of classical FGM and FGM-I is illustrated in Fig. 7. As it is seen from the figure, FGM-I decreases the path length while keeping the safety criterion.

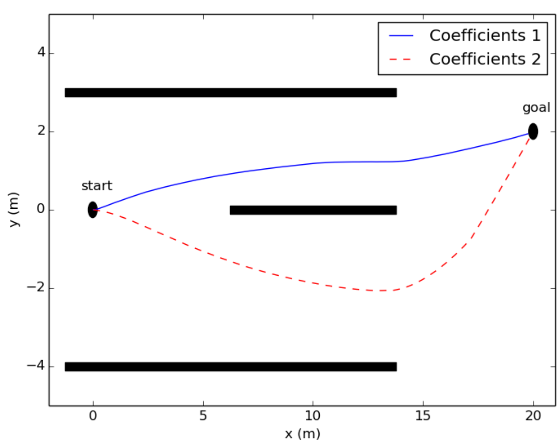


Fig. 6: The effect of the coefficients on the trajectory. Coefficients 1: k_1=0.2, k_2=0.8. Coefficients 2: k_1=0.6, k_2=0.4.

B. Penalizing the Gap Change

In FGM, the robot may change the selected gap instantly due to small differences between the gap size values and this reveals the zigzag problem. The zigzag problem of the orginal FGM is shown in Fig. 4 where the gap size values in left and right hand side of the robot are very close to each other. In FGM-I the selected gap's utility value is the best one which means it has the biggest value, U_{max}. An additional reward constant R is added to the selected gap's utility as shown in (4). For the next gap calculation, the utility value of a gap must be greater than the rewarded maximum R_{max} for a gap change decision. This yields a penalty effect on changing the selected gap.

$$R_{max} = U_{max} + R \tag{4}$$

(R_{max}: Rewarded maximum. U_{max}: Utility value for the best gap. R: Reward constant.)

The effect of adding a reward constant for the selected gap is illustrated in Fig. 8. The reward constant affects intensity of the zigzag movement. If it is selected as big value, the robot can lose its changing the gap ability. If the reward constant is selected as small value, the zigzag problem can continue. Consequently, the reward constant is selected as 0.1 depending on the result of the experiments. It can be observed that after adding the constant value, in FGM-I, the final heading angle and the orientation of the robot are more stable and the trajectory is smoother comparing to original FGM.

IV. SIMULATION

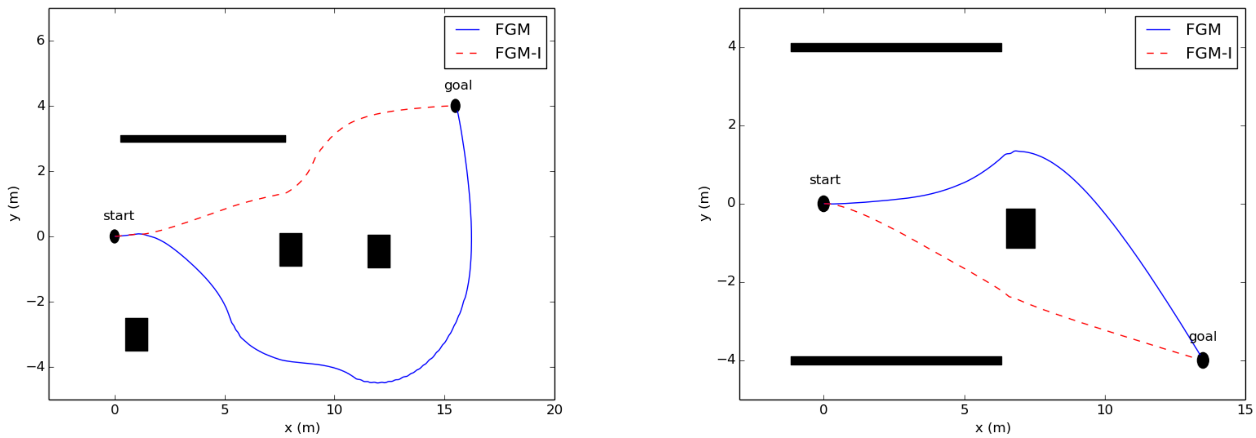
The simulation platform for FGM and FGM-I is a differential drive powerchair. Computer Aided Design (CAD) model of the powerchair is prepared by using SOLIDWORKS® as shown in Fig. 9. In order to use the powerchair in non-deterministic Gazebo environment [24], the CAD model is exported to Universal Robot Description File (URDF) format. The presented algorithms are simulated in Gazebo environment using Robot Operating System (ROS) [25]. The Gazebo simulation environment is shown in Fig. 10 with the powerchair and the obstacles around. Although the angular velocity of the powerchair is calculated by FGM and FGM-I, the linear velocity is defined as a constant 0.4 m/s. To achive desired angular and linear velocities, (5) and (6) [26] are utilized for the calculation of each wheel's angular velocity.

$$v_R = v + \frac{L}{2}w \tag{5}$$

$$v_L = v - \frac{L}{2}w \tag{6}$$

(v: Linear velocity of robot's CoG. w: Angular velocity of robot's CoG v_R: Linear velocity of the right wheel. v_L: Linear velocity of the left wheel. L: Distance between right and left wheel.)

Illustrations of the comparative simulation results are provided in previous sections to give the results of the new approach clearly. As it is seen from the simulations, the FGM-I produces shorter and smoother trajectories than the original FGM. On the other hand, in order to make a fair comparison, 100 Monte Carlo simulations in which the coordinates of the obstacles are specified randomly, are implemented for FGM-I and FGM. A LIDAR (Light Detection and Ranging) sensor is used in simulations to detect the obstacles with a total 180° field of view and 6 m range. In these simulations, 10 rectangular obstacles with



(a) Configuration I

(b) Configuration II

Fig. 7: Comparison of the FGM and FGM-I

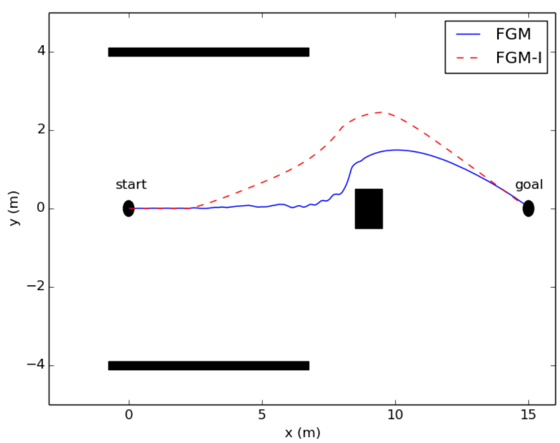


Fig. 8: The effect of additional reward to selected gap

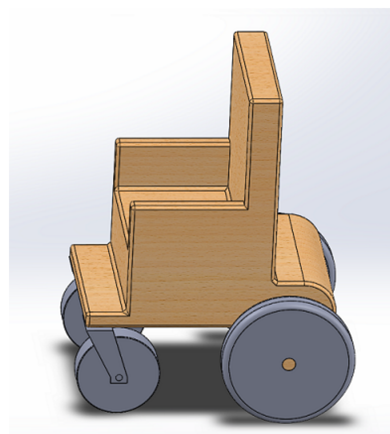


Fig. 9: CAD model of the powerchair

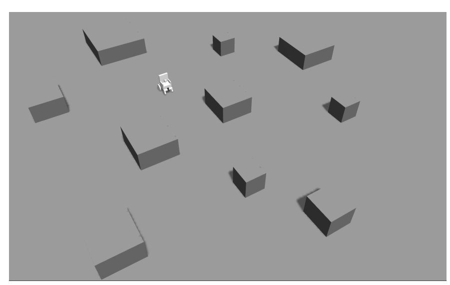


Fig. 10: Gazebo simulation environment with the modeled powerchair

various size and positions are distributed randomly to the 35X25m environment. Initial and goal coordinates are given as [0-0] and [35-25] respectively .

The results are compared in terms of safety, length of path and zigzag density. The safety metric is given in (7) and the calculation of pth norm of any given function is defined in (8) which is the same metric used in [16], [27]. This metric is the measurement of minimum distance to obstacles during the trajectory. d_{min} is the closest distance between the powerchair and obstacles and the given scalar d_0 denotes the distance to an obstacle that poses no danger for collision during execution. More information on this metric can be found in [16], [27].

$$f(t) = \begin{cases} \frac{1}{d_{min}} - \frac{1}{d_0}, & \text{for } d_{min} < d_0 \\ 0, & \text{for } d_{min} \geq d_0 \end{cases} \tag{7}$$

$$\|f\|_p = \left(\int |f(t)|^p \ dt \right)^{1/p} \tag{8}$$

The first norm (p = 1) of the collision avoidance metric is calculated to compare the safety of the trajectories.

In order to calculate the zigzag density, yaw-rate values of the powerchair is integrated during each simulation. First norm of the yaw-rate is used as it is shown in (8) and used as smoothness metric. Finally, the path length is directly calculated from each simulation.

In each Monte Carlo simulation, a random map is generated and classical FGM and FGM-I are tested in same environment. After 100 random simulations, average values of safety metric, yaw-rate metric and traveled distance values are calculated and shown in Table I. (α=5, k_1=0.4, k_2=0.6, R=0.1, d_0=10 m).

TABLE I: Average of 100 Monte Carlo Simulations Results

	Average traveled distance (m)	Average yaw-rate metric	Average safety metric
FGM-I	45.17	0.0365	0.211
FGM	52.55	0.0436	0.202

As can be seen from Table I, the FGM-I generates 14% shorter trajectories than the FGM, while the average safety metric is only 4% worse. In addition, the average yaw-rate metric of FGM-I is 16% lower than the FGM. This means that the resulted trajectories are smoother and zigzag problem of FGM is decreased in a significant rate.

V. CONCLUSION

In this study, two critical problems of original FGM approach are repaired by an improved version, named as FGM-I. In FGM-I, the gaps are selected according to a new utility function and unnecessary gap changes are penalized for zigzag problem. The proposed idea is proven to improve the original FGM by Monte Carlo simulations. According to the simulation results, path length and zigzag problem is repaired in a significant rate while the safety of the path is almost same. The FGM-I is going to be implemented and tested on a real autonomous powerchair platform as a future study.

ACKNOWLEDGMENT

This work was supported by Turkish Scientific and Technological Research Council (TUBITAK) under project no 215E140.

REFERENCES

[1] S. M. LaValle, “Rapidly-exploring random trees: A new tool for path planning,” 1998.

[2] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[3] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*. MIT press, 2011.

[4] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[5] 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, 1968.

[6] H. M. Choset, *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.

[7] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” in *Autonomous robot vehicles*. Springer, 1986, pp. 396–404.

[8] M. G. Park, J. H. Jeon, and M. C. Lee, “Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing,” in *Industrial Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on*, vol. 3. IEEE, 2001, pp. 1530–1535.

[9] M. C. Lee and M. G. Park, “Artificial potential field based path planning for mobile robots using a virtual obstacle concept,” in *Advanced Intelligent Mechatronics, 2003. AIM 2003. Proceedings. 2003 IEEE/ASME International Conference on*, vol. 2. IEEE, 2003, pp. 735–740.

[10] J. Borenstein and Y. Koren, “The vector field histogram-fast obstacle avoidance for mobile robots,” *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 278–288, 1991.

[11] I. Ulrich and J. Borenstein, “Vfh+: Reliable obstacle avoidance for fast mobile robots,” in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 2. IEEE, 1998, pp. 1572–1577.

[12] ——, “Vfh/sup*: local obstacle avoidance with look-ahead verification,” in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 3. IEEE, 2000, pp. 2505–2511.

[13] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[14] P. Ogren and N. E. Leonard, “A convergent dynamic window approach to obstacle avoidance,” *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 188–195, 2005.

[15] P. Saranrittichai, N. Niparnan, and A. Sudsang, “Robust local obstacle avoidance for mobile robot based on dynamic window approach,” in *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2013 10th International Conference on*. IEEE, 2013, pp. 1–4.

[16] V. Sezer and M. Gokasan, “A novel obstacle avoidance algorithm:follow the gap method,” *Robotics and Autonomous Systems*, vol. 60, no. 9, pp. 1123–1134, 2012.

[17] M. Zohaib, S. M. Pasha, N. Javaid, A. Salaam, and J. Iqbal, “An improved algorithm for collision avoidance in environments having u and h shaped obstacles,” *Studies in Informatics and Control*, vol. 23, no. 1, pp. 97–106, 2014.

[18] M. Zohaib, S. M. Pasha, N. Javaid, and J. Iqbal, “Iba: Intelligent bug algorithm–a novel strategy to navigate mobile robots autonomously,” in *International Multi Topic Conference*. Springer, 2013, pp. 291–299.

[19] J. Minguez and L. Montano, “Nearness diagram navigation (nd): A new real time collision avoidance approach,” in *Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, vol. 3. IEEE, 2000, pp. 2094–2100.

[20] J. Minguez, “The obstacle-restriction method for robot obstacle avoidance in difficult environments,” in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, 2005, pp. 2284–2290.

[21] A. Stentz, “Optimal and efficient path planning for partially-known environments,” in *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*. IEEE, 1994, pp. 3310–3317.

[22] A. Stentz *et al.*, “The focussed d^* algorithm for real-time replanning,” in *IJCAI*, vol. 95, 1995, pp. 1652–1659.

[23] S. Koenig and M. Likhachev, “Fast replanning for navigation in unknown terrain,” *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 354–363, 2005.

[24] N. Koenig and A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,” in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 3. IEEE, 2004, pp. 2149–2154.

[25] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “Ros: an open-source robot operating system,” in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, 2009, p. 5.

[26] J. L. Martínez, A. Mandow, J. Morales, S. Pedraza, and A. García-Cerezo, “Approximating kinematics for tracked mobile robots,” *The International Journal of Robotics Research*, vol. 24, no. 10, pp. 867–878, 2005.

[27] S. Nair and M. Kobilarov, “Collision avoidance norms in trajectory planning,” in *American Control Conference (ACC), 2011*. IEEE, 2011, pp. 4667–4672.