Model-based Dynamic Obstacle Avoidance on Inclined Surface

Xing Yan*
yan.xing@u.nus.edu
Mechanical Engineering
National University of Singapore

Guillaume Sartoretti
guillaume.sartoretti@nus.edu.sg
Mechanical Engineering
National University of Singapore

ABSTRACT

In this paper, we present a Gaussian process (GP) potential field method for dynamic obstacle avoidance on an inclined surface. We implement this method on a wheeled robot in simulation. We propose this new switching artificial potential field control method according to the characteristic of a specific situation, reducing issues in traditional artificial potential field methods. The main parts of the paper focus on the establishment of GP potential field and the overall idea of the control loop. Experiments in three different cases are done to demonstrate the effectiveness of the method.

KEYWORDS

gaussian process, dynamic obstacle avoidance, simulation, artificial potential field control

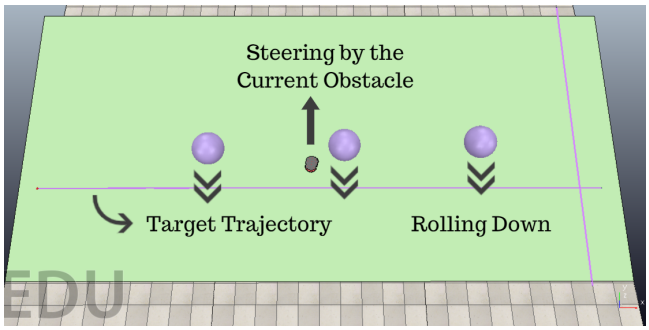

Figure 1: The general view of the simulation

1 INTRODUCTION

With the extensive applications of mobile robots in the industrial field and people's daily lives, robotic research is getting increasing attention. Obstacle avoidance is the basic function in the robot area, and avoidance in real-time is a rapidly growing field. Real-time obstacle avoidance robots are at the heart of search and rescue, logistics, unmanned workshops, plumbing and so on [1]. In this article, we propose a GP potential obstacle avoidance method for a specific robotic locomotion problem. In our problem, a robot is given a path to follow on an inclined surface, while having to dodge obstacles that roll down the surface usually on his flank (perpendicular to his path). To simplify the condition, we assume that all the obstacles are distant from each other, so that the robot can pass through the gaps to avoid the obstacles when necessary.

To date, there are many dynamic obstacle avoidance methods, such as rolling path planning [2], genetic algorithm [3], etc. Artificial potential field methods are widely used because of its simpler computing and superior real-time performance [4]. These methods rely on building a potential field around the robot. The control is defined by attractive and repulsive potential parts. The repulsive parts and attractive parts are usually built around obstacles and the goal of the robot respectively. The resulting control drives the robot to avoid obstacles based on repulsive energy while going towards the target based on the attractive force.

The artificial potential field we use in this article is different from other traditional artificial potential methods, in that it only has a repulsive part. Moreover, our control resembles more that of a switching potential method, as if and only if there is no repulsive force from the obstacles, a PD control, working as the attractive potential field, will work to lead the robot back to the original desired trajectory.

The structure of this article is as follows: Section 2 presents some previous work about artificial potential field methods, demonstrates and details the GP potential field and the simulator we used. Section 3 introduces details about the GP potential field and the overall establishment of our control method. In section 4, we conduct experiments in three different cases and analyze our experimental results. Section 5 discusses the reasons for failures. Finally, in section 6, we put forward some directions for further researches.

2 BACKGROUND

2.1 Artificial Potential Field Method

The first artificial potential field method was proposed by Khatib [5] in 1985. It is a useful way to obstacle avoidance in the time-varying system, and computationally light real-time control method. In our work, as the robot's goal is to follow a straight path in the world is simple and clear, we just use the repulsive concept to avoid the obstacles and a simple PD controller to realize the goal of the robot as the attractive force.

However, artificial potential field approaches have two major limitations [6].

- Goals non-reachable with obstacle nearby (GNRON).
  The repulsive potential force of obstacles and the attractive potential force of targets can sometimes end up balanced, thus blocking the robot in a local extremum of the potential.
- Local minima between obstacles.
  Get into a local minimum of the potential field formed between obstacles' repulsion.

Lots of researchers have been trying to solve these two key problems. Recently, Li proposed an effective improved artificial potential field (IAPF) -based regression search method to overcome the local minima with doing small oscillations [7]. Fedele presented an avoidance method based on switching potential function [8], which is similar to the idea in this article.

Fortunately, for the simple problem we consider, no such limitations will happen. We use a switching potential method, but not

traditional potential field method. In this case, GNRON problems cannot happen. Additionally, by proposing a clever way to leverage properties of the problem (explained in subsequent sections), we are able to propose a potential field that does not exhibit local minima.

2.2 Gaussian Process (GP)

GP is an interpolation method. Each point in this aggregation is seen as a point in Gaussian distribution. And each random set in this interpolation is seen as points in a multivariate normal distribution [9]. The kernel function of the GP is used (and user-defined) to measure the similarity between points, predict more points and to smooth the overall curve/surface [10]. In this article, we build a GP potential model that can meet our obstacle avoidance requirements.

2.3 CoppeliaSim

CoppeliaSim also called V-rep, is a robot simulator gathering modeling, programming and physical engine [11]. In this paper our experiments are done based on CoppeliaSim, using python as the remote API to set the control commands. The wheeled robot we used is a mobile robot model 'Pioneer P3DX'. To make the control more precise we use the synchronous mode in CoppeliaSim. This mode allows us to send orders step by step.

3 BUILD UP OF THE OVERALL CONTROL

In this section, we briefly illustrate the setup procedures of the GP model and the control loop. There are three different parts during the whole course. The differential velocity principle is the basic theory used to give commands to the robot. We then describe how the potential field is built as a GP, and then how to implement closed-loop obstacle avoidance using this GP.

3.1 Differential Velocity Principle

In this paper, we consider a differential drive robot, for which we use standard differential velocity principles [12].

- When the velocities of the left and right motor are different, the wheeled robot moves in an arc.
- When the velocities are set the same, the robot moves in a straight line.
- (A special case, unused in this work) When the velocity of the left motor is negative of the right one, the robot goes circle in place.

These principled are summarized in Eqs.(1) and (2). v_c is the instantaneous linear velocity of the robot. v_r and v_l are the speed of the right and left motors respectively. R is the radius of rotation. l is the length of the interval between two motors. These two equations determine the behavior of the robot together.

$$v_c = \frac{v_r + v_l}{2} \tag{1}$$

$$R = \frac{l}{2}\frac{v_r + v_l}{v_r - v_l} \tag{2}$$

To simplify the control process, we fix the robot's forward speed, i.e., fix v_c in Eq.(1). Therefore, the increment or decrement we send to the left and right motors should be of the same magnitude but opposite signs. We choose to develop a simple PD controller to calculate the increment or decrement sent and track a straight trajectory. The principle of our PD controller is shown below. K_p and K_d are proportionality and differential coefficients respectively. U is the increment/decrement in each simulation step. e is the distance between the robot and the target line/trajectory. $\frac{de}{dt}$ is the (discrete) derivative of the error at each step.

$$U = K_\mathrm{p} e + K_\mathrm{d}\frac{de}{dt} \tag{3}$$

In particular, we set the target velocity of the left motor greater when the robot is above the target line on the incline. Meanwhile, when the robot is below the target line, the target velocity of the right motor is required to be larger.

3.2 GP for Artificial Potential Fields

As the speeds of the obstacles falling are quite greater than the speed of the robot, it is dangerous for the robot to go across them below. The premise of our idea about the GP potential model is to only send upwards commands to the robot, which means during the obstacle avoidance period we only let the robot to go towards the top of the slope, as the steering process shown in Fig. 1.

According to our specific problem, each of our GP potential models is built by a column of Gaussian distributions very close to each other in the direction in which obstacles fall. As shown in Fig. 2. The peaks of these Gaussian distributions are seen as the projection of the possible future locations of the current obstacle on the surface (based on a simple physical model). It means that each possible future location we choose is surrounded by a potential energy field. We build these Gaussian distributions close enough, overlap each other and then do the kernel function to build the GP potential field. We want every possible area of the robot, which may have collision without GP potential method control, is covered by the GP potential model. Therefore, the further away from the current obstacle in its descent direction, the wider the Gaussian distribution should be. That is, the further the distance, the sooner the preparation.

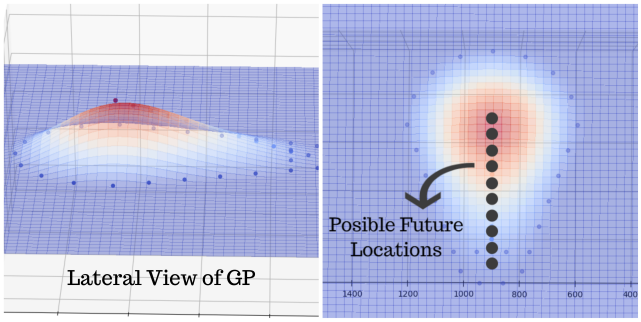

Figure 2: Different perspectives of GP potential field

The robot always drives from left to right in our specific problem and after going across the obstacles, no more control needed in the right parts of the obstacles. The robot can just turn to go towards its target. In this condition, we propose to simply remove the right half of every GP model to decrease the computation levels. After we do the gradient operation on the GP potential field, we can get

the repulsive force provided by the obstacles. The gradient of the GP model is a mesh grid with 2D vectors spread out. We use its x-direction gradient as our repulsive energy. As we said above, we want only upwards commands, the negative parts of its gradient can be deleted. Then the sketch map is shown in Fig. 3.

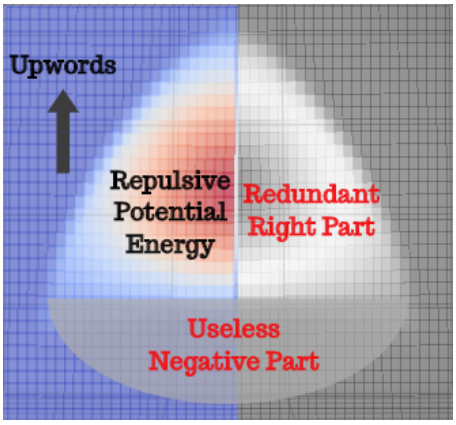

Figure 3: Gradient of the GP

3.3 The Establishment of GP Potential Model and Present of Potential Energy

A GP is built by multiple Gaussian distribution, based on the kernel function. However, to have the potential mode all over the inclined surface we need to have an overall position of the obstacles and to superpose these different modes onto one GP. The flow path is shown below.

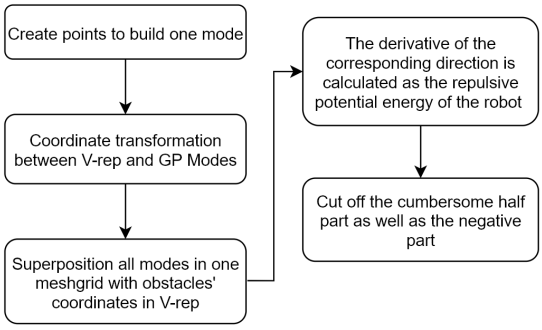

Figure 4: Establishment of GP potential model and acquire of potential energy

For choosing sampling points, we use the standard normal table as a reference and choose no more than ten points for each fine Gaussian distribution representation, considering the computational limitation. Then one mode can be achieved by about 5 or 6 sets of fine Gaussian distribution points and several essential modes to guarantee the integrity of the GP. Fig. 5 shows the sampling points of one mode.

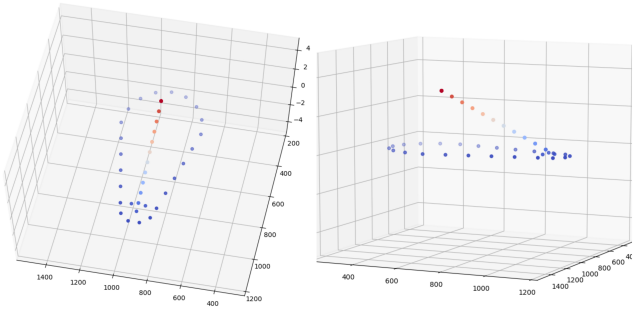

Figure 5: Sample points of one mode viewing from different directions

The coordinates transformation relation between V-rep and GP should be solved before superposing all the modes to finish the elementary potential model according to real-time varying positions of obstacles.

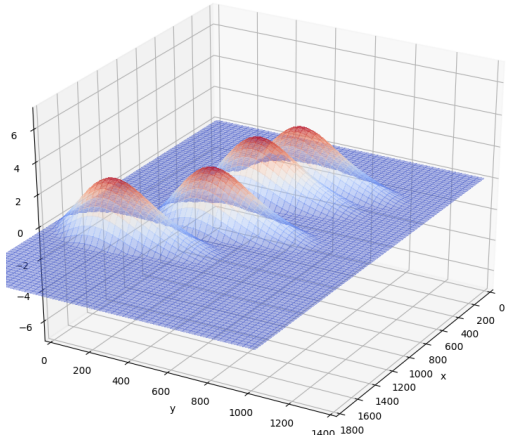

Figure 6: Final GP showing a superposition of different modes (located at the current position of each target)

3.4 Closed-loop Control

For the overall control method, two different conditions arise. If the wheeled robot is below the target trajectory, it is a simpler way, as the regression PD control and the potential field (if any) have commands that align. They both drive the robot upwards on the slope. In the other case, two possible conditions should be divided depending on whether these two forces align. In particular, the main issue arises when the PD controller drives the robot downward on the slope, while the GP-based obstacle avoidance drives the robot upwards. In this case, we remember that our main purpose is to avoid obstacles, so that we should judge whether there is repulsive energy given first. If the repulsive energy given is not zero, we follow the command given by the potential mode only. Otherwise, we will follow the command give out by the regression PD control. The resulting behavior of our controller resembles that of a switching controller, like what Fedele presented [8].

The flow diagram is Fig. 7. The calculation of the commands, in other words the target velocity set to the motors, is divided into two parts as shown in the flow diagram. To avoid erratic controls, we cap the maximum wheel speed of the robot experimentally.

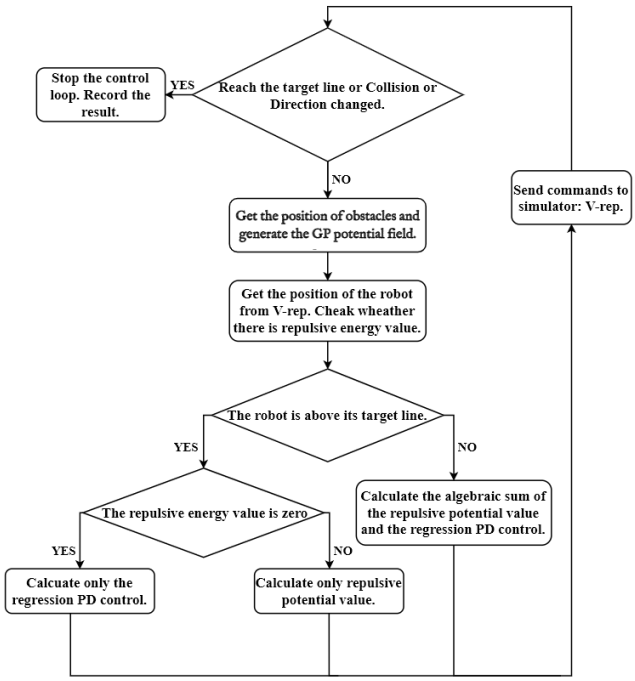

Figure 7: Reference diagram of control loop

- The trajectory tracking PD control.
- The repulsive potential control.

At the beginning, the calculation of the repulsive potential control is just proportional control. However, in the progress of the program experiment, we figured out that a differential part was needed. We let the angle changes of the robot as the multiplier of the differential part to decrease the possibility of over-rotate.

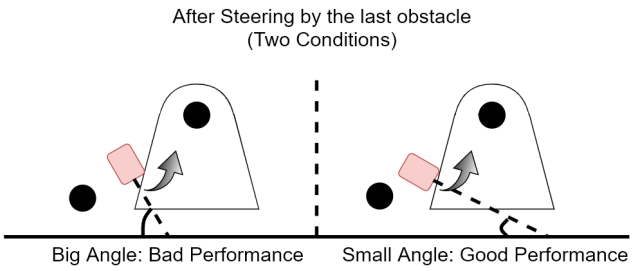

Figure 8: Schematic Diagram for two different conditions

Another detail is worth mentioning. When the robot is going down to the target line with a large angle just under the control of attractive PD command and soon afterward meets the next obstacle in the environment, we expect the robot to perform badly (sometimes resulting in collisions). That is, such case is hard for the robot to do great turning from big angle downwards to upwards, as shown in Fig. 8. To solve this problem, an item proportional to the angle (the angle between the heading of the robot and the target line) is introduced to the downwards PD control commands. When the robot heading upwards, the bigger the angle the more intense the command. Otherwise, the bigger the angle the weaker the command.

Table 1: Settings in Each Experiment

Environment Conditions	Number & Diameter of Spheres
Simple (Constant Environment)	3 Spheres (0.8m)
Medium (Random recall position)	4 Spheres (0.8m)
Hard (Random recall position and time)	4 Spheres (1.6m)

4 SIMULATION EXPERIMENTS

In this section, we describe three sets of simulation experiments. To show the effects of the GP potential method, three different environmental conditions are tested, a simple, a medium and a hard one.

4.1 Simulation Environment Settings

An inclined slope is set in the simulator first. we set it to be 30m*26m, with eight degrees inclined angle. The robot is the 'Pioneer P3DX' model in V-rep. The length and width are 0.8m and 0.64m respectively. It starts its traveling from 10.5m distance from the bottom of the slope in each experiment. And the initial target speed of the robot is set to be 12rad/s on both motors.

Obstacles are set as spheres in V-rep. However, they can be irregular shapes in the real world. They are released from the top of the slope. We want to build the environment with a continuous flow of obstacles appearing and falling. Therefore, we propose to reset these balls back to their original place or around their original place after exceeding the boundary of the slope, as there is no remote API for python to create new obstacles in the simulator. The resulting continuous recalling of spheres in the simulator can be considered as ceaseless obstacles falling from the slope. For each of the three considered cases, we set different sizes of spheres, which are shown in detail in table 1. In the same experiment, spheres are the same size as each other.

For the simple experiment, we set three recurrent spheres. These obstacles are set to release at the same decided time in each simulation. To some extent, ignoring the time error in each simulation, the environment is near-deterministic.

Four spheres are added in the medium complexity of the environment with a sort of random in the position they recalled and released. The random positions where they start from and return are mutually independent square areas. The left and right order of obstacles will not be changed. And enough gaps should be left between the square areas for the robot to pass. This is depicted in Fig. 9 below.

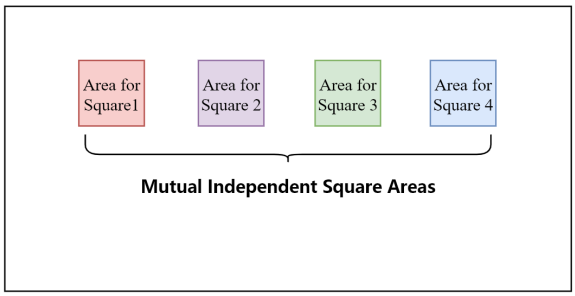

Figure 9: Randomness of obstacles released positions

The hard condition for the control is to set not only the randomness of the areas the obstacles recalled but also the starting time and the recall time of each sphere.

To demonstrate the superiority of our control, for each of the three conditions, we made a comparison with the experiment with only PD control (without GP potential field) in the same condition. We observe the results from the probabilities of collision and the time needed to reach the goal position (end of the trajectory). Before simulating, we need to define and open the collision detection mechanism between obstacles and the robot in the simulator, so that we can read the collision conditions in every simulation step. The robot's linear damping and angular damping sometimes need to be turn up to enhance the stability. Or it may go side slip in some conditions. The environment in V-rep is shown in Fig. 10.

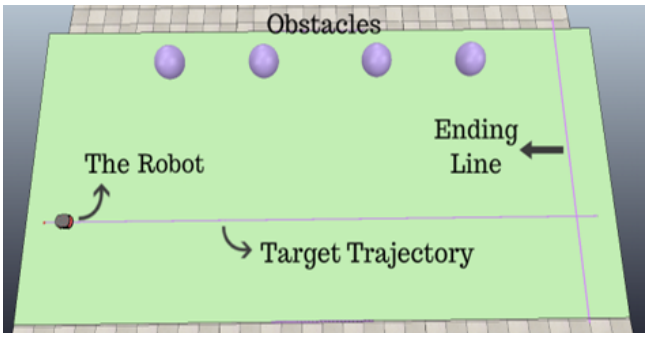

Figure 10: Environment in V-rep

4.2 Main Control Loop Details

For the overriding control loop, we set it as a while loop. Sending commands to the simulator in each step unless the robot goes unexpectedly or reaches the finishing line. First of all, the parameters' transfer should be done. With the position of obstacles getting from the simulator, we can establish the GP model mentioned in the previous subsection. The built potential model gives the repulsive energy value at the coordinate position of the robot. This parameter transfer is done in dual threaded mode, as the calculation of the GP potential field spends three times time compared to the main control loop. As finer the mesh or bigger the environment, the time spent will increase. After calculating commands according to the control method mentioned above, the robot receives the commands, does simulation and then back to the first step, judging the condition of the simulation. A video of the experiments can be found at https://youtu.be/1R-jX3F9-Nc.

4.3 Simulation Results

The success rate of the three experiments in each test case and their contrast tests without control are shown as Fig. 11. It is the result of 1000 experiments for each condition and each controller. We can see that no matter in which conditions, our potential method can effectively improve the success rate of collision avoidance.

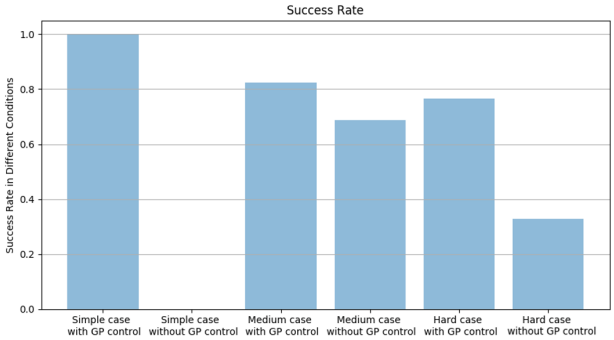

Figure 11: Success rate of different conditions

In the simplest condition, our GP potential method together with PD control can achieve 100% obstacle avoidance. It validates the feasibility of our method. Besides, note that we created the simplest way with a great possibility for the robot to collide with the obstacles on purpose. Therefore, the contrast tests without control is shown to be no success.

In the medium complex condition, the success rate is increased by 13.7% after using our potential method, 20% of the original rate. And we can see the success rate without GP is 68.7%. It is high. Upon closer observations of the results, we believe this is due to the small size of the spheres. That is, the bigger obstacles the bigger possibility of collision. So, we changed the size of the spheres in the following experiment.

For the most sophisticated condition with bigger obstacles, the performance improvement is above 200%. The successful avoidance rate is improved by 43.9%. However, the success rate for the one without potential control is as low as 32.8%. On the one hand the decreased success rate compared to 68.7% in the second test is because of the bigger size of spheres. On the other hand, the randomness of the release time improved the randomness of the environment, which cause it more likely to collide.

The average travel time and its error bar is shown as Fig. 12. We recorded the arriving time of the finishing line for each successful obstacle avoidance and counted. Note that the comparison test for three obstacles is inconclusive because each time in this relatively deterministic environment there must be a collision.

The interval sizes of time changing in the tests with the GP potential method become bigger when the simulation environment goes more random. Also, in this aspect, the travel time results of the simple and medium conditions are not so representative. According to our result of three obstacles with a certain environment we

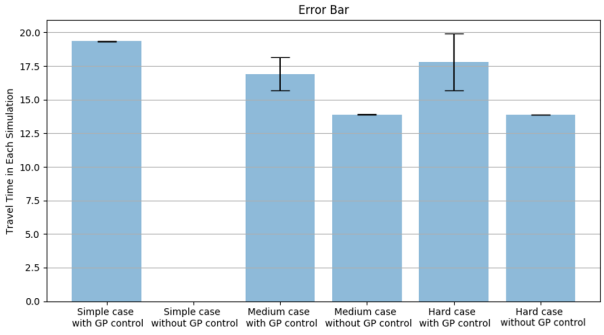

Figure 12: Error bar of time travel in each simulation of different conditions

can see as the robot has to avoid all three obstacles in the defined environment every time, the average travel time is relatively high compared to other conditions. This is to be expected, as actively avoiding collisions will increase the path length of the robot to finally reach its goal.

Looking at the most significant increase in time in the third comparative group, we find that the increase in travel time was only about 28% (17.8s and 13.9s for the two tests respectively) contrast to the one without potential control. We believe that this small timing increase is acceptable, especially for the hardest conditions where it is linked to a more-than-double performance improvement in collision avoidance.

The possibility distribution of the routes traveled by the robot in the three different conditions are demonstrated below in Fig. 13 and Fig. 14. It is clear that the more random the environment, the more random the distribution. We can explicitly pick out the magnitude of the likelihood of avoiding each sphere.

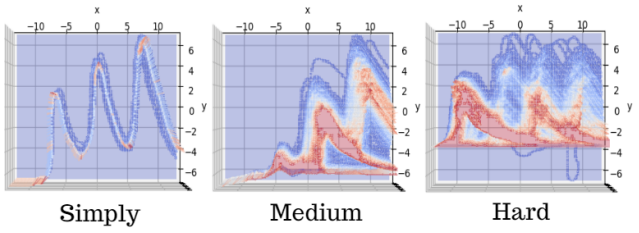

Figure 13: The travel routes of 1000 results of different conditions

For the medium complexity test, the robot nearly never passes above the first sphere. We believe that might be because the initial set of the first sphere is not near the starting point of the robot enough. And as the robot advances, it gets further and further away from the target, that is because the spaces between the spheres are so small, and the robot does not have the time to return to the target line. However, we can see in the last section of the route, the robot tends to return to the target line in the end.

For the hardest condition, we can distinctly see four hump there, representing the avoidance route of the four spheres. The robot is

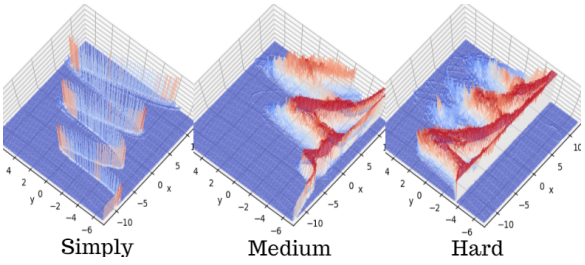

Figure 14: The side view of the 1000 travel routes of different conditions

more able to avoid the first and the third spheres in this test. And we can point out that the robot sometimes goes downwards the target line and can turn back immediately.

5 DISCUSSION

In this section, we mainly analyze the success rates of our approach. Some reasons for failure will be mentioned. Actually, there are three main reasons why there are several collisions when the environment is random.

First, the defect of the GP potential model may cause collisions. When the robot goes very close to an obstacle in the horizontal direction, and then just starts feeling the repel force from the potential field, it will go upwards immediately. However, there might not always be enough space for the robot to rotate enough and avoid the obstacle, while the obstacle is quickly falling to its vicinity. Then the robot and the obstacle will collide. Similarly, we observed cases where there is just enough space for the robot to change its direction. Since the robot is an object with a width, but not a point, sometimes its edges would scrape obstacles and cause a collision.

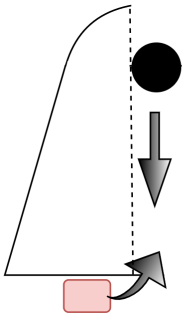

Figure 15: The sketch map of the defect of GP potential model

To solve this problem, we could delete several parts of the GP potential model after calculation to avoid such a problem. For example, just like Fig. 16 below. And for the over width of the robot, we could add the width of the robot to the obstacles and shrink the robot to zero width [13].

Second, when there is no always enough space between neighboring obstacles, collisions may happen. When two obstacles are too close to each other, the robot might not have enough space to

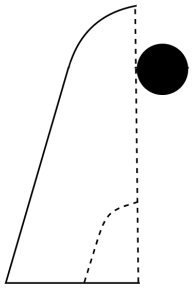

Figure 16: The sketch map of the improvement idea of GP potential model

turn up if it just avoided the obstacle ahead and start to turn down to the target line.

Third, the sudden appearance of the obstacles can cause collisions. There are also some limitations in the simulator that caused several failures. As the inclined surface we build is finite in size, sometimes the obstacles have no extra length to be released. when the robot goes towards the top of the slope and close to the place where we release the obstacles, the suddenly appeared (recalled) obstacles might not allow the robot time to react.

6 CONCLUSION

In this paper, we propose a well-worked GP potential field method for obstacle avoidance in a specific problem. We can find out that for the sufficiently complex conditions, the GP potential field method can improve the success rate effectively with relatively little increment in the travel time. In addition, for the conditions which are more likely to have collisions, the more remarkable the improvement in the success rate can be achieved. However, more improvement works deserve to explore.

Two main points about perfecting the GP potential field should be mentioned. The design of its width and length and its shape optimization. The width and length of the GP determined when to send the robot dodge commands. If the width and length are too big, not only a waste of time but also may lead to excessive turn. About the shape of the GP potential field, just as mentioned in the above subsection, there are some defects. The best way to solve it is to optimize its shape like Fig. 16. It's worth mentioning that a GP comes with its associated uncertainty distribution, that could be used in cases where the position of the obstacles cannot be 100% perfectly measured, and that this could be an interesting extension that should be investigated.

In practical application, this obstacle avoidance method also has some practical factors to be considered. For example, we need an expensive radar or depth camera to detect the whole circumstances of the environment.

ACKNOWLEDGMENTS

First of all, I would like to thank my tutor for his patient guidance and encouragement, which enabled me to learn a lot in this one-year project. Besides, I would like to thank my lab partners for their help and companionship in programming and daily life. I also want to thank YiXuan for comforting and accompanying me when I was frustrated, so that I could successfully complete this project.

REFERENCES

[1] Enxiu Shi, Tao Cai, Changlin He, and Junjie Guo. Study of the new method for improving artifical potential field in mobile robot obstacle avoidance. In *2007 IEEE International Conference on Automation and Logistics*, pages 282–286, 2007.

[2] ZHANG Chun2Gang XI Yu2Geng. Robot rolling path planning based on locally detected information. , 29(1), 2003.

[3] Lin Lei, Houjun Wang, and Qinsong Wu. Improved genetic algorithms based path planning of mobile robot under dynamic unknown environment. In *2006 International Conference on Mechatronics and Automation*, pages 1728–1732. IEEE, 2006.

[4] Zhenhua Pan, Dongfang Li, Kun Yang, and Hongbin Deng. Multi-robot obstacle avoidance based on the improved artificial potential field and pid adaptive tracking control algorithm. *Robotica*, 37(11):1883–1903, 2019.

[5] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 500–505. IEEE, 1985.

[6] Min Gyu Park, Jae Hyun Jeon, and Min Cheol Lee. Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing. In *ISIE 2001. 2001 IEEE International Symposium on Industrial Electronics Proceedings (Cat. No. 01TH8570)*, volume 3, pages 1530–1535. IEEE, 2001.

[7] Guanghui Li, Yusuke Tamura, Atsushi Yamashita, and Hajime Asama. Effective improved artificial potential field-based regression search method for autonomous mobile robot path planning. *International Journal of Mechatronics and Automation*, 3(3):141–170, 2013.

[8] Giuseppe Fedele, Luigi D'Alfonso, Francesco Chiaravalloti, and Gaetano D'Aquila. Obstacles avoidance based on switching potential functions. *Journal of Intelligent & Robotic Systems*, 90(3-4):387–405, 2018.

[9] Zexun Chen, Bo Wang, and Alexander N Gorban. Multivariate gaussian and student-t process regression for multi-output prediction. *Neural Computing and Applications*, pages 1–24, 2019.

[10] J Bernardo, J Berger, A Dawid, A Smith, et al. Regression and classification using gaussian process priors. *Bayesian statistics*, 6:475, 1998.

[11] Eric Rohmer, Surya PN Singh, and Marc Freese. Coppeliasim (formerly v-rep): a versatile and scalable robot simulation framework. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[12] D. J. Balkcom and M. T. Mason. Extremal trajectories for bounded velocity differential drive robots. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 3, pages 2479–2484 vol.3, 2000.

[13] Jang-Ho Cho, Dong-Sung Pae, Myo-Taeg Lim, and Tae-Koo Kang. A real-time obstacle avoidance method for autonomous vehicles using an obstacle-dependent gaussian potential field. *Journal of Advanced Transportation*, 2018, 2018.