Research and Experiment of Robot Obstacle Avoidance Navigation Function Based on VFH Algorithm and Simplified BUG Algorithm in Webots Simulation Environment

Lijing Zhou//2168704 2182343

Abstract—**In complex obstacle scenarios, how to solve the path planning and real-time obstacle avoidance of mobile robots has always been a difficult point in robot navigation [1]. This paper focuses on the research of path planning technology and obstacle avoidance methods for mobile robots in dense obstacle scenarios. This project uses Webots software, based on the research of the obstacle avoidance method of the Vector Field Histogram (VFH) algorithm, realizes the autonomous navigation and obstacle avoidance function, and designs different obstacle situations, and finally verifies and discusses the effectiveness of the algorithm. In addition, we do a brief comparison with the simplified BUG algorithm to show the advantages.**

Keywords: VFH algorithm, BUG algorithm, obstacle avoidance, autonomous navigation, Webots

I. INTRODUCTION

Autonomous positioning and obstacle avoidance are important prerequisites for service robots to provide services. In an unknown environment, autonomous mobile robots need to rely on their own sensors to continuously obtain information about the surrounding environment, identify the location of obstacles, and perform calculations and autonomous decision-making [2]. All mobile robots have certain characteristics of collision avoidance, from the original algorithm to the use of complex algorithms to prevent the robot from avoiding obstacles. Real-time obstacle avoidance algorithms are much more complicated, because they not only involve the detection of obstacles, but also a certain quantitative measurement of obstacle size. Once these are determined, the obstacle avoidance algorithm needs to guide the robot to bypass the obstacle and move towards the original goal. Usually, the program requires the robot to stop from the obstacle, take a measurement, and then resume movement. Obstacle avoidance can lead to non-optimal paths because no existing knowledge of the environment is used [3].

Based on the study of Webots software this semester, our group continue to explore this practical problem as the theme of this project. According to the investigation, it is found that there are many kinds of robots positioning methods, and the obstacle avoidance algorithms are endless. We choose the Vector Field Histogram (VFH) algorithm as our method to realize the robot's real-time obstacle avoidance navigation, and its function is realized by code in Webots simulation environment, and it is verified by experiment. Then we tested the classic simplified BUG algorithm to compare the length of the path to show the performance.

In Section II, this paper will sort out and explain the VFH algorithm and simplified BUG algorithm. Then, in the Section III how to simulate the realization of the robot in the Webots software to follow the VFH algorithm for real-time obstacle avoidance navigation is introduced. The section IV will conduct experimental tests on the VFH algorithm developed in this project, construct obstacles in different positions, test the results of the algorithm to achieve obstacle avoidance navigation, and discuss the comparison with simplified BUG algorithm. The section V puts forward some hypotheses and conclusions based on experimental results.

A. Hypothesis Statement

By applying 8 distance sensors, try to detect the 360-degree total range of the robot. Use the thinking of VFH algorithm and simplified BUG algorithm to complete the code design, so that the robot can achieve the purpose of path planning and obstacle avoidance in a simple obstacle environment.

We investigate this hypothesis through a structured experiment on the Webots simulation, testing the performance of the different environment and different algorithm.

- **Aims**: To learn the VFH algorithm, and use the code in Webots environment to realize the autonomous navigation and obstacle avoidance between point to point, and test and discuss the experimental results.
  1) "Understand how Vector Field Histogram(VFH) algorithm works."
  2) "Achieve it by code."
  3) "Record the length of path as a factor to test the efficiency."
  4) "Design different obstacles and test the feasibility of the code."
- **Objectives:**: The core of this project is the theoretical realization of the VFH algorithm. The simplified BUG algorithm counts for less part of the report. On this basis, some physical data of the movement of the robot between two points are recorded, and multiple obstacle scenarios are designed to verify the feasibility of the code.

1) "To make the robot to run from the start point to the end position with path planning and avoiding obstacles."
2) "To construct a robotic system including and length of path to collect meaningful data."
3) "A complete analysis of the conclusions must be carried out before the whole system assessment, to carry out clear experimental verification."

- **Challenges**:The biggest challenge of this project is how to use the code to implement the VFH obstacle avoidance algorithm in the Webots simulation environment.In addition, due to the limitation of sensors and the mismatch between the size of the robot and the obstacle in the simulated environment, some problems have been caused.
  1) "The robot's own sensors cannot detect the distance between the robot and the obstacle in 360 degrees."
  2) "The limited detection range of the sensor and the size of the robot itself cause obstacle avoidance problems."
  3) "In reality, an overly complex obstacle environment may cause the algorithm to fail to run."

II. Implementation

When we determine the starting position and destination of the robot, there is no doubt that the optimal solution for navigation is a straight line from point to point. To understand from another angle, we need to make the linear velocity direction of the robot equal to the vector from the position of the robot to the destination. However, if obstacles are encountered in the point-to-point straight line, how the robot judges the avoidance direction, and how to set the path planning to efficiently reach the end point with a smooth route is the focus of this article. Therefore, we chose the VFH algorithm as the theoretical background.

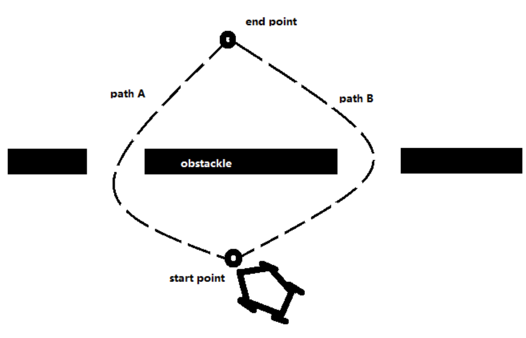

Fig. 1. heading figure.

In this section, we will introduce the VFH algorithm and make a brief description of the robots and sensors used in the Webots environment

A. Vector Field Histogram (VFH) algorithm

The VFH(Vector Field Histogram) algorithm was proposed by Borenstein. This method is a real-time obstacle avoidance strategy based on artificial potential field method and certainty grid method. Through the improvement of the algorithm, some scholars put forward VFH+ algorithm and VFH* algorithm. VFH* algorithm has the characteristics of strong operability and powerful functions, and has gradually become the mainstream local obstacle avoidance algorithm [4].

VFH algorithm introduces the concept of certainty grid, and regards the working environment of the robot as composed of two-dimensional histogram grids, each of which corresponds to an obstacle confidence level. The sensor is used to detect obstacles, and when obstacles are found, the confidence of the grid of obstacle positions is increased by 1. Through the rapid sampling of sensors, the probability histogram of obstacles can be obtained [5].

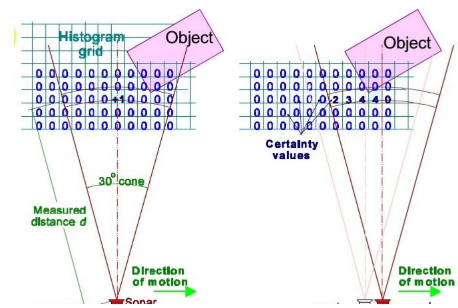

Fig. 2. Only add one cell per range read.

After obtaining the probability histogram of obstacles, the concept of artificial potential field force is introduced into it (that is, the target attracts the robot, and the obstacles repel the robot), and the resultant force of the two forces makes the robot move quickly to the obstacles. The robot moves and avoids obstacles under the action of resultant force.

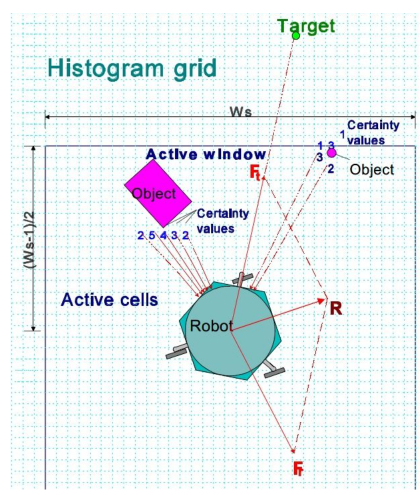

Fig. 3. Occupied cells exert a repulsive force on the robot.

The central idea of the VFH is to represent the occupation of all grids by obstacles and the influence on the direction of motion by statistical histograms [6]. Such a representation is very suitable for inaccurate sensor data and adapts to the fusion of multi-sensor readings. VFH also relies on ranging sensors to update the map. Compared with other algorithms, the advantage is that it can smoothly drive in environment with dense obstacles, respond quickly, and do not need to stop waiting for planning. The speed and direction can be output, and the movement speed is limited by the frequency of sensor observation and update.

VFH has three layers of data. The top layer is a two-dimensional grid map updated based on sensor observation data; the middle layer is a one-dimensional polar histogram, which divides the active window ROI into multiple sectors by angle, each of the sector uses polar obstacle density to represent the occupancy of obstacles (density, quantity, confidence, etc.) in the corresponding angle range; the bottom layer is the output of VFH. The main content of the algorithm is to do the conversion of these three layers of data, the steps are as follows:

Each grid is represented by an obstacle vector, with the direction from the cell to the centre of the robot. The size is proportional to the degree of certainty and inversely proportional to the distance to the centre of the robot. The polar obstacle density is calculated as a superposition of the magnitude of the obstacle vector falling in the same sector.

According to the peak value of the polar obstacles density and density threshold, the choice and the most consistent valley value target direction. Take the middle Angle of the valley as the direction of motion. The adaptive threshold can be set by global planning. Adjust the movement speed according to the density of the current movement direction. Finally, complete the navigation task.

B. BUG algorithm

The BUG algorithm is one of the simple obstacle avoidance algorithm. This algorithm is based on the changing environment. When no obstacle is encountered, move in a straight line toward the target. When the distance sensor detects an obstacle, the robot will go around the boundary of the obstacle, and when the robot reaches the point that has the minimal distance, the robot will leave the obstacle to rotate the direction and continue to go straight [7]. The robot will start from point "qstart" to point "qgoal" and the picture show the basic path.

C. Webots

Webots is a development environment for robot modelling, programming, and simulation of movement. It can describe the characteristics of the robot and the environment 3 d world. It can provide a variety of special-shaped robot models, but also allows users to build their own simulation environment and robot model, to achieve user requirements under specific circumstances of simulation. As a simulation software, Webots support a variety of programming languages (C language/Python and so on), convenient for users to program and control the robot [8].

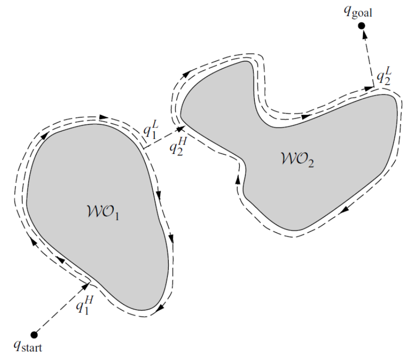

Fig. 4. BUG algorithm.

Webots provides a variety of nodes that simulate the real world. During the simulation, the experimenter can select the corresponding node and change the parameters to achieve the model construction. The simulation model in this article includes two parts: the scene model and the robot model. The scene model is to set several obstacles between the starting point and the end point of the robot and create the world separately. The robot model includes the robot body, a set of differential wheels, eight infrared distance sensors. Among them, the differential wheel is used to realize a two-wheeled robot and can provide a differential steering function. Eight infrared distance sensors are used to measure the distance between it and obstacles.

D. Robot and sensors

The simulated e-puck robot has 8 infrared sensors located around the body of the robot. These sensors are enumerated according to the scheme in the figure below.

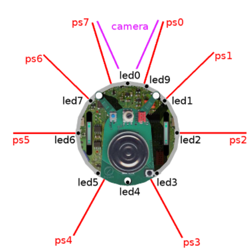

Fig. 5. e-puck robot.

III. Experiment Methodology

In this section, several key parts of the code will be explained, that is, how to realize the robot navigation and obstacle avoidance based on the VFH algorithm.

A. *Overview of Method*

First, the autonomous navigation and obstacle avoid of the robot can be divided into two parts, namely n gation and obstacle avoidance. From another perspec the latter is an additional task based on the former. W there is no obstacle detected on the motion trajector the robot between the starting point and the end p the robot only needs to complete the navigation. I obstacle is detected during movement, the vector anal of the VFH algorithm is used to avoid the obstacle, then drive to the end.

B. *Discussion of Variables*

In the process of our code implementation, we ma introduce angle variables. First, the initial angles of eight sensors can be converted (marked as shown in the figure below), and the angle of the connection between the start point and the end point of the robot can also be passed through Mathematical calculations, we call it theta. angle formed by the robot's real-time position and the end position is called e-theta, and the angle formed by the robot's real-time position and the end position is called pose.th.

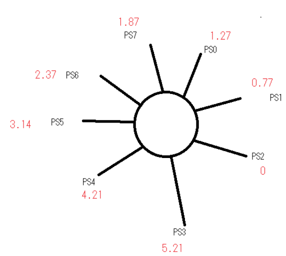

Fig. 6. ps1-7.

When the robot starts to move, it is judged which of the eight distance sensor angles is closest to the connecting angle between the starting point and the end point, and the robot moves in that direction.

During the movement, any distance sensor detects the shortest safe distance of an obstacle, which means a collision may occur, and then a judgment is made: the real-time angle of the other sensors that have not detected the obstacle (not triggered) are compared to the theta point(the angle of the connection between the start point and the end point), the robot will find the closest one and turn to move in that direction. The above detections are all carried out in real time.

The end point coordinate set in this experiment is (400,400), and the error between the stop position of the robot and the end point must not be greater than 2. Based on the size of the robot itself, the size of obstacles, and

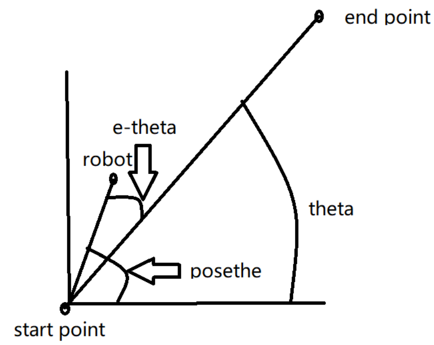

Fig. 7. Angles.

the operating speed setting that cannot be too high, the safety distance is artificially set also.

C. *Discussion of Metric(s)*

The output value of the VFH obstacle avoidance navigation system proposed in this paper includes the path length of the robot, the time to reach the destination and records the path points the robot travels in the form of an array. However, due to the different robot speed settings and the selectivity of safety thresholds, these data cannot form a quantifiable rating system. The data collected above can only help us to provide limited assistance in verifying the feasibility of the code. The main evidence to judge whether the algorithm is implemented is the video of the observation robot autonomously avoiding obstacles and navigating under different obstacle environments in the Webots simulation environment.

D. *Simple Introduction of simplified BUG algorithm*

We simplify the traditional BUG1 algorithm and divide the whole process into two parts. In the first part, the robot will do rotation until the direction of the E-puck is the same as the direction from the current position to the destination. Then, if the robot meets an obstacle, it will do a simple obstacle avoidance until there are no obstacles in the safe distance range.

IV. Results

A. *Results of the VFH algorithm*

In order to test the developed navigation and obstacle avoidance code based on the VFH algorithm, we constructed three Webots worlds. There are three different numbers of obstacles with different distributions between the start point and the end point of the robot. As shown in the following pictures:

In the above three worlds with different obstacle environments, the robot can start from the starting point,

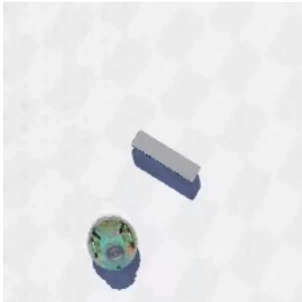

Fig. 8. WORLD 1.

Fig. 9. WORLD 2.

complete navigation and obstacle avoidance, and move to the end point.

It is true that the most intuitive detection method is to watch the video of the actual movement of the robot in the Webots simulation environment, and the actual running path of the robot is recorded through the pictures. As shown in the figure, in three different obstacle environments, each time before the robot encounters the first obstacle, it chooses to follow the best direction from the starting point to the end point to drive out; when the sensor is in the range of receiving signals, the obstacle information is reached, then the algorithm makes a judgment to move around the obstacle (such as bypassing the first obstacle, then the next obstacle is detected), until the obstacle is completely bypassed, and then according to the starting point and the vector direction of the terminal line is used for the best route navigation, and finally moves to the vicinity of the key

Fig. 10. WORLD 3.

point to complete the overall point-to-point navigation and obstacle avoidance.

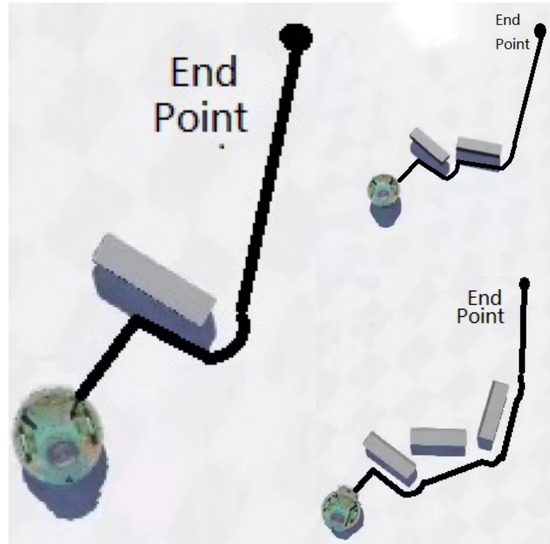

Fig. 11. Path collection.

In order to detect and record data more intuitively, to facilitate the discussion and analysis of the degree of realization of the algorithm to achieve obstacle avoidance navigation. The system records the coordinates of the robot's stopping point (distance from the end point), the total length of the running path, the angle between the start point and the end point, the angle between the robot's real-time position and the end point, the real-time angle of the eight sensors etc on the output terminal.

```
x is 401.64, y is  398.88, theta is 1.27  Total Distance is 681.55
```

Fig. 12. Output of VFH algorithm No1.

In World 1, the end point coordinates are set to (400, 400). The total distance was 681.55. The angles of the sensors also conforms to the setting of the algorithm.

```
x is 401.79, y is  399.15, theta is 1.34  Total Distance is 688.44
```

Fig. 13. Output of VFH algorithm No2.

In World 2, it can be seen that the end point coordinates are also set to (400, 400). The robot ran 688.44 in total. The real-time angles measured by the sensors are also correct.

```
x is 401.62, y is  398.85, theta is 1.29  Total Distance is 681.87
```

Fig. 14. Output of VFH algorithm No3.

In the same way, it can be seen that in World 3. The robot totally run 678.87 respectively. The real-time angles measured by the sensors are also correct. It is worth mentioning that the code running tests in the above three worlds have been repeated many times, and the total length of the robot running path finally recorded is very reliable. In multiple tests, under the same test environment, the route choices given by the algorithm navigation are all fixed routes.

B. Results of the Simplified BUG algorithm

We run the code in the three same world and get the result. The data of the length of the path is collected below.

```
x is 400.81, y is  398.29, theta is 1.78  Total Distance is 715.09
x is 400.81, y is  398.44, theta is 1.81  Total Distance is 727.71
x is 390.79, y is  355.96, theta is 1.16  Total Distance is 668.03
```

Fig. 15. Output of the Simplified BUG algorithm from World1 to World3

V. Discussion and Conclusion

A. Comparison of VFH algorithm and Simplified BUG algorithm

In first place, during the living videos of the attachment, the robot with VFH algorithm run smoothly than the robot with Simplified BUG algorithm from the three worlds. The VFH algorithm allows the robot can choose a smooth path according to the changing environment. In addition, the choice of the Histogram will provide the robot with suitable velocity in order to improve the adaptability to various differential constraints of the robot itself. The robot with VFH algorithm has a shorter path in the world 1 and world 2, however, it has a longer path. To analyse the world, during the world 1 and world 2, the obstacles are closed and can cause a huge influence to VFH algorithm to make the robot move smoothly and shorter. In the world 3 with 3 obstacles, the later two obstacles cause less influence to Simplified BUG algorithm to make the robot go on straight course.

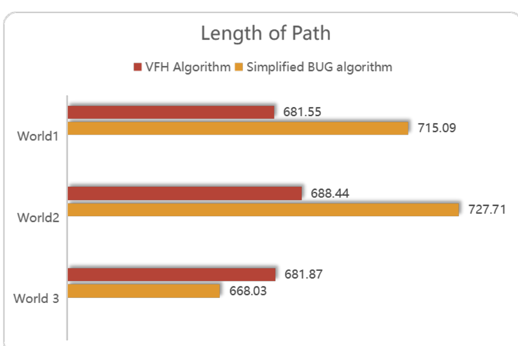

Fig. 16. Output of the Simplified BUG algorithm from World1 to World3

B. Conclusion

Autonomous navigation and obstacle avoidance of robots are of great significance in real life, but due to the simulation test in the Webots environment, many hardware and environmental issues have been ignored. There is an inevitable gap between the real environment and the simulated environment. To achieve reliable and accurate positioning of robots and rapid obstacle avoidance in the actual environment, there are still many problems that need to be solved, such as the limitations of environmental characteristics, the accumulation of errors in related algorithms, the passage of obstacles based on complex terrain, and so on. The indoor environment has the characteristics of space limitation and large contrast of indoor light illuminance, which all affect the choice of positioning method. The starting point of this article is based on point-to-point navigation in a simple ideal environment, with the VFH algorithm as the entry point, aiming to study the practical value of accurate positioning and fast obstacle avoidance methods, complete code development and pass Webots simulation experiments. In the simulation test under the simple obstacles' environment, the robot can complete effective "point-to-point" autonomous navigation and obstacle avoidance.

In the code implementation process of the VFH algorithm, the size and motion characteristics of the robot were not considered, and the robot was only regarded as a mass point, which is also a potential defect of this experiment. In addition, as the minimum safety distance between the robot and the obstacle, this value needs to be manually set. This experiment uses a single threshold. Although the obstacle avoidance navigation algorithm performs well in the three sets of environments tested, it is conceivable that when the robot passes through a narrow channel environment, the sensor continuously samples, and the channel openings may be open and closed. The second transformation causes the robot to appear indecisive, which in turn affects navigation efficiency and obstacle avoidance decisions.

In addition, limited to the characteristics of the analogue components themselves, due to the limited detection distance of the eight distance sensors in the simulated environment, the robot can only detect obstacles through the sensors when it is very close to the obstacles, and the eight distance sensors cannot achieve 360 degrees real-time detection. This can also be a place for subsequent improvement of the simulation method.

It is undeniable that the basic idea of VFH algorithm is still local path planning, that is, it does not try to find the optimal path (the optimal path can only be found after the complete environment information is given). In addition, robots controlled by VFH algorithms may "fall into" blind spots (When get trapped, mobile robots often exhibit what's called "circular behaviour," which involves going in circles between multiple traps. These two points can be regarded as questions to be explored in the next step.

REFERENCES

[1] J. J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, "Footstep planning among obstacles for biped robots," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, vol. 1, pp. 500–505, IEEE, 2001.

[2] T. Lu, K. Yuan, W. Zou, and H. Hu, "Study on navigation strategy of intelligent wheelchair in narrow spaces," in *2006 6th World Congress on Intelligent Control and Automation*, vol. 2, pp. 9252–9256, IEEE, 2006.

[3] Y. Kong, C. Chen, Y. Yang, and Z. Cao, "Path following and obstacle avoidance control based on different fuzzy grains," in *Proceedings of the 2015 Chinese Intelligent Automation Conference*, pp. 55–64, Springer, 2015.

[4] I. Ulrich and J. Borenstein, "Vfh/sup*: Local obstacle avoidance with look-ahead verification," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 3, pp. 2505–2511, IEEE, 2000.

[5] I. Ulrich and J. Borenstein, "Vfh+: Reliable obstacle avoidance for fast mobile robots," in *Proceedings. 1998 IEEE international conference on robotics and automation (Cat. No. 98CH36146)*, vol. 2, pp. 1572–1577, IEEE, 1998.

[6] J. Borenstein, Y. Koren, *et al.*, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE transactions on robotics and automation*, vol. 7, no. 3, pp. 278–288, 1991.

[7] Q.-L. Xu, T. Yu, and J. Bai, "The mobile robot path planning with motion constraints based on bug algorithm," in *2017 Chinese Automation Congress (CAC)*, pp. 2348–2352, IEEE, 2017.

[8] O. Michel, "Cyberbotics ltd. webots™: professional mobile robot simulation," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, p. 5, 2004.