International Conference on Control, Automation and Systems 2008
Oct. 14-17, 2008 in COEX, Seoul, Korea

Second-order Nonlinear Function Navigation Method for Fast Mobile Robots

Dong-Han Kim[1] and Keun-Ho Rew[2]

[1] Department of Electrical Engineering, Kyung Hee University, Yongin-si, Korea
(Tel : +82-31-201-3831; E-mail: donghani@khu.ac.kr)
[2] Department of Robotic Engineering, Hoseo University, Chungnam, Korea
(Tel : +81-41-540-9571; E-mail: khrew@hoseo.edu)

Abstract: There are two different approaches to the mobile robot navigation problem including that of robot soccer: deliberative and reactive. In the deliberative approach, a detailed geometric model of the world is used to create a plan that accomplishes the task. This approach suffers from high computational requirements and lacks of robustness in the face of the world's uncertainty. The reactive approach overcomes the disadvantages of the deliberative approach, but the design that could obtain a desired behavior is difficult. The work reported in this paper overcomes this difficulty by adapting a new navigation method, the limit-cycle navigation method. The defects of the potential field method are also presented. In the proposed method, with the information on the relative position of obstacles, target and robot, the obstacle that affects the robot's path is identified. Then, direction of turn, counter-clockwise or clockwise is decided using the radius of the virtual obstacle. Real experiments ascertain the merits of the proposed method.

Keywords: Limit-cycle, robot navigation, robot soccer

1. INTRODUCTION

The problems confronting robot soccer can be divided into three parts: exact positioning through vision system, navigation planning using the position information and following the planned path. Among these, navigation plan is the representative of robot soccer team's winning prospects. From the view of navigation, robot soccer faces a dynamic navigation problem with moving obstacles (opponent robots) and a moving target (ball).

Slack [4] defines the navigation problem as, "The general problem for a robot navigator is to move the robot in service of the current navigation task while accounting for both internal and external constraints (e.g., actuator limits and obstacle)." There have been two different approaches to the mobile robot navigation problem including that of robot soccer: deliberative and reactive. In the deliberative approach [2], a detailed geometric model of the world is used to create a plan that accomplishes the task. In theory, three-step navigation process (sensing, planning and acting) should allow a robot to display the desired smooth goal-directed behavior. In practice, however, such approaches lack robustness in the face of the world's uncertainty, causing failure and requiring frequent new planning. Repeated cycling through the sense-plan-act loop to accomplish tasks could potentially be a robust solution to the problem, as it allows the system to handle the world's uncertainties. However, the computational complexity of such systems is generally too great to attain the cycle rates needed for the resulting action to keep pace with the changing environments. So, this approach suffers from the high computational requirements and lack of robustness in the face of the world's uncertainty.

In the reactive approach [3], [7], [9], little or no model of the robot's surroundings is needed and the robot's sensors and actuators are coupled through a transfer function which produces an emergent navigation behavior. This approach generally requires relatively little computation and commits the robot to a particular action for only a short span of time. Thus, a reactive system handles the world's uncertainty by continually updating its action. However, because of limited representational and reasoning ability, purely reactive systems are not goal-directed.

Several researchers have looked at combining reactive and deliberative approaches [8]. This paper proposes a novel navigation method, the limit-cycle navigation method that combines advantages of the above two previous approaches.

Section 2 describes defects of the potential field method and proposes the limit-cycle method. Section 3 presents the local navigation method using the limit-cycle method. The real system, its implementation and the experiments conducted are presented in Section 4. Section 5 presents the experimental results. Concluding remarks and directions for future research follow in Section 6.

2. NEW APPROACH: AN INTRODUCTION TO LIMIT-CYCLE METHOD

A. Problem of potential field method

The potential field method adopts the attractive-repulsive features attributed to many of the sensor-actuator transformations of reactive machines. Guiding a robot's action with such a plan is accomplished by summing the effect of the individual gradients at the robot's location. For example, as shown in Fig. 1, a plan to move the robot across the playground avoiding the obstacle consists of an attractive field towards the left end of playground and a repulsive field for an obstacle and a bottom wall.

978-89-93215-01-4-98560/08/$15 ©ICROS

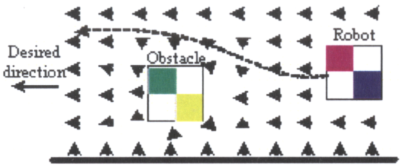

Fig. 1 Potential field method

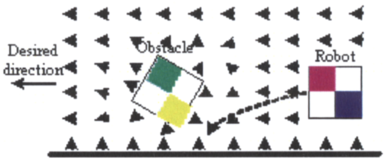

Fig. 2 Model with local minima

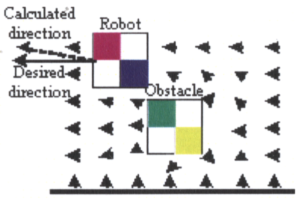

Fig. 3 Undesirable obstacle influence

There are, however, some disadvantages in the potential field method that limit their usefulness. The potential field method provides no information about the way in which the robot should avoid obstacles. This lack of information can lead to task failure as shown in Fig. 2. If an obstacle is near the bottom wall, fields guide the robot to collide the wall. Another disadvantage is that objects that are not in the way of the robot can produce unwanted effects (repulsive fields) as shown in Fig. 3, whereby the robot moves in a different direction different from the desired one.

B. Limit-cycle method

Consider the following 2nd order nonlinear system [10]:

$$\begin{aligned} \dot{x}_1 &= x_2 + x_1(1 - x_1^2 - x_2^2) \\ \dot{x}_2 &= -x_1 + x_2(1 - x_1^2 - x_2^2) \end{aligned} \qquad (1)$$

and the Lyapunov function

$$V(x) = x_1^2 + x_2^2$$

The derivative of $V(x)$ along the trajectories of the system is given by

$$\begin{aligned} \dot{V}(x) &= 2x_1\dot{x}_1 + 2x_2\dot{x}_2 \\ &= 2x_1x_2 + 2x_1^2(1 - x_1^2 - x_2^2) - 2x_1x_2 + 2x_2^2(1 - x_1^2 - x_2^2) \\ &= 2V(x)(1 - V(x)) \end{aligned}$$

The derivative $\dot{V}(x)$ is positive for $V(x) < 1$ and negative for $V(x) > 1$. Hence, on the level surface of $V(x) = c_1$ with $0 < c_1 < 1$ all the trajectories will be moving outward, while on the level surface of $V(x) = c_1$ with $c_1 > 1$ all the trajectories will be moving inward, as shown in Fig. 4. This can replace the potential field method.

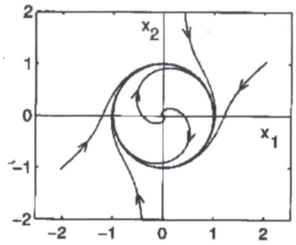

Fig. 4 Phase portrait of limit cycle (CW)

The general form of (1) can be derived by replacing 1 with *r* as follows:

$$\begin{aligned} \dot{x}_1 &= x_2 + x_1(r - x_1^2 - x_2^2) \\ \dot{x}_2 &= -x_1 + x_2(r - x_1^2 - x_2^2) \end{aligned} \qquad (2)$$

and the Lyapunov function is

$$V(x) = x_1^2 + x_2^2 .$$

The derivative of $V(x)$ along the trajectories of the system is given by

$$\begin{aligned} \dot{V}(x) &= 2x_1\dot{x}_1 + 2x_2\dot{x}_2 \\ &= 2x_1x_2 + 2x_1^2(r - x_1^2 - x_2^2) - 2x_1x_2 + 2x_2^2(r - x_1^2 - x_2^2) \\ &= 2V(x)(r - V(x)) \end{aligned}$$

The derivative $\dot{V}(x)$ is positive for $V(x) < r$ and negative for $V(x) > r$. The general form of the limit-cycle method is thus derived. Fields in Fig. 4, however, move inward in a clockwise direction. The robot should avoid the obstacle by moving in a clockwise or counter-clockwise direction, depending on the relative position of the robot, target and obstacle. The counter-clockwise field can be derived in the following. (1) is transformed to:

$$\begin{aligned} \dot{x}_1 &= -x_2 + x_1(r - x_1^2 - x_2^2) \\ \dot{x}_2 &= x_1 + x_2(r - x_1^2 - x_2^2) \end{aligned} \qquad (3)$$

Then, all the trajectories will be moving inward, as shown in Fig. 5.

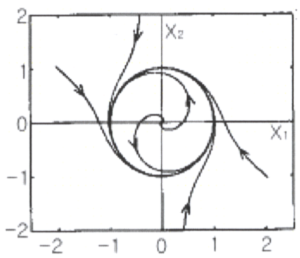

Fig. 5 Phase portrait of limit cycle (CCW)

Limit-cycle method is an element existing in the local navigation plan. This selects the efficient way by which the robot should avoid obstacles rather than moving far away from them.

3. Local Navigation using Limit-cycle Method

A. Local Navigation

Fig. 6 depicts the limit-cycle method which can drive a robot towards the desired direction using without colliding with an obstacle by introducing a virtual obstacle.

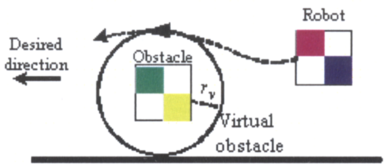

Fig. 6 Navigation using the limit-cycle method

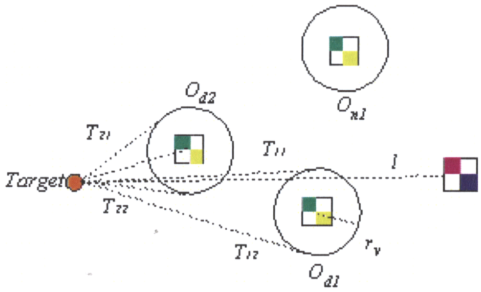

Fig. 7 A situation

Some definitions are now listed in the following.

1. Rotational direction (R_d): It decides the direction in which the robot avoids the obstacles: counter-clockwise (CCW) or clockwise (CW)
2. Virtual obstacle (O_v): In general, the robot is assumed as a point mass. This may lead to collision with actual obstacles. Virtual obstacle is an obstacle of which radius is decided by the relative position to the robot and the size of the obstacle and the robot. In this paper, for simplicity, the virtual obstacle is assumed to be circular.
3. Virtual radius (r_v): The radius of the virtual obstacle. It varies with the size of the robot and the relative position to the robot.
4. Disturbing obstacle (O_d): Virtual obstacles that are between the robot and the target point. These obstacles are assigned numbers in accordance with their distances from the robot, O_{d1} will designate the disturbing obstacle that is nearest to the robot.
5. Non-disturbing obstacle (O_n): Virtual obstacles that are not in between the robot and the target point.
6. Tangent points (T_{n1}, T_{n2}): Points that are tangent to the line of n-th disturbing obstacle to the target point. There are two tangent points.

It should be noticed that r_v is used as r in (2), (3) to apply the limit-cycle method.

Now, the steps to be followed in the process of local navigation by the limit-cycle method are:

1. Make a line l from the robot to the target in a global coordinate Σ_{OXY} as follows:
$$ax + by + c = 0$$
2. Consider virtual obstacles as disturbing obstacles, O_d's if the line l crosses them, else, they are considered as non-disturbing obstacles, O_n's.
3. Move towards the target if there is no O_d.
4. In Fig. 8, we can calculate the distance d from the center of the obstacle O_d to the line l as
$$d = \frac{aQ_x + bQ_y + c}{\sqrt{a^2 + b^2}}$$
where (Q_x, Q_y), (G_x, G_y) and (R_x, R_y) are the xy-value of the obstacle, the target and the robot, respectively. If d is positive, the robot avoids the obstacle, O_d clockwise. If d is negative, the avoidance takes place in a counter-clockwise direction.
5. Adopt the limit-cycle method with the direction decided in step 4.

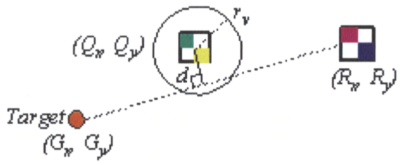

Fig. 8. Decision of rotational direction

6. While the robot moves, the line l varies. So, repeat steps 2 ~ 4 until the task is completed.

For example, suppose, as shown in Fig. 9, there are three obstacles between the robot and the target. The robot should move towards the target avoiding these obstacles which are marked as A, B and C.

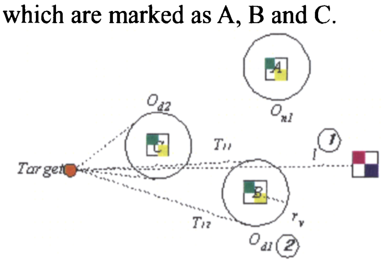

Fig. 9 Navigation example

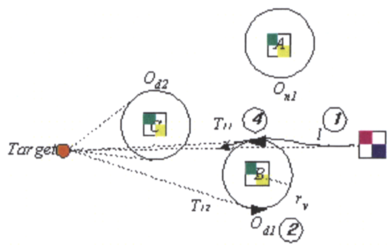

Fig. 10 Navigation example: after step 4

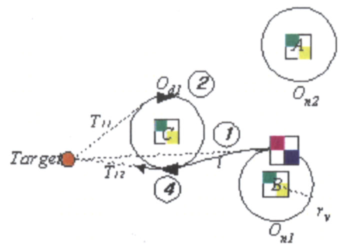

Fig. 11 Navigation example: after avoiding obstacle B

A line L can be marked from the robot to the target (step 1). This line goes through two obstacles B and C, so they are considered as O_{d1}, O_{d2}, respectively and obstacle A as O_{n1} (step 2). Using the direction of line l through O_{d1}, the robot decides the direction in which it should avoid obstacle B. The counter-clockwise direction is chosen (step 4). It follows the chosen fields until it avoids obstacle B. Once the robot passes obstacle B, the line l ceases to go through obstacle B. So, obstacle B becomes O_{n1} and obstacles C and A become O_{d1} and O_{d2}, respectively, (step 1)~(step 2). Then, if the limit-cycle method is adopted again, the navigation path obtained will be as shown by the solid line in Fig. 11.

4. Modeling of a Mobile Robot

Differential-drive mobile robots with non-slipping and pure rolling are considered in this paper. The velocity vector $Q = [v \quad w]^T$ consists of the translational velocity of the center of the robot and the rotational velocity with respect to the center of the robot. The velocity vector Q and a posture vector $P = [x \quad y \quad \theta]^T$ are associated with the robot kinematics, as follows::

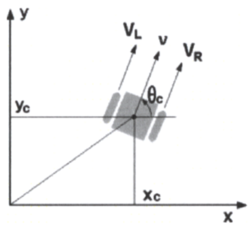

Fig. 12 Kinematics modeling

$$\dot{P} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = J(\theta)Q$$

$$Q = [v \quad w]^T = \begin{bmatrix} \frac{V_r + V_l}{2} & \frac{V_r - V_l}{L} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{L} & \frac{1}{L} \end{bmatrix} \begin{bmatrix} V_l \\ V_r \end{bmatrix}$$

where V_l is the left wheel velocity, and V_r is the right wheel velocity.

(1) represents a 2nd-order nonlinear system. Assume that x_1 and x_2 in (1) are x and y in xy-plane, respectively. Then, using the derivative of x_1 and the derivative of x_2, the desired direction in (x, y) can be obtained as follow:

$$\theta_{desired} = \tan^{-1}\left(\frac{\dot{x}_2}{\dot{x}_1}\right) \quad (4)$$

$$\theta_{error} = \theta_{desired} - \theta_{robot}$$

Using θ_{error}, V_l and V_r can be obtained.

$$V_l = v - K_p \cdot \theta_{error} - K_d \cdot \dot{\theta}_{error} \quad (5)$$

$$V_r = v + K_p \cdot \theta_{error} + K_d \cdot \dot{\theta}_{error}$$

where K_p and K_d are a proportional and a derivative gain, respectively. The robot follows the field of the limit-cycle method with the velocities designated in (5).

5. Experiments

A. System

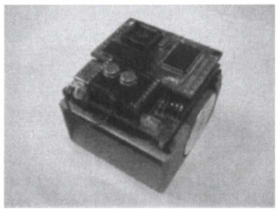

Fig. 13 SOTY IV

SOTY IV, by which the experiments were conducted, consists of three parts.

1. Host Computer: It has all strategies and robot controllers and a vision system. A 450MHz Pentium III was used.
2. Robot: It is 7.5 x 7.5 x 7.5cm^3 in size and uses two DC-motors, INTEL 80296 for CPU. Normally, a robot soccer team consists of 3 robots.
3. Vision System: There is an overhead CCD camera which monitors the robots wearing a color uniform. The location of the overhead camera should be at a height of 2m or higher from the playground (150 x 130 cm). Vision system consists of a CCD camera, Matrox meteor II frame grabber and millennium I VGA card. This system can monitor the positions and angles of three different color robots at 60Hz.

B. Experimental Results

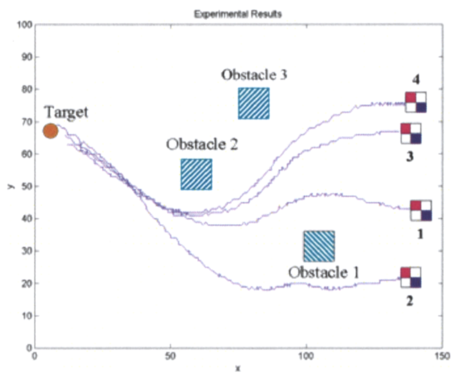

Fig. 14 Experimental results

Fig. 14 shows the real experimental results of the proposed limit-cycle navigation method with SOTY IV system. The robot should go to the goal avoiding three obstacles. Solid lines represent paths of the robot and numbers 1, 2, 3 and 4, initial points of the robot. When the robot starts from the position 1, it avoids the obstacle 1 in a counter-clockwise direction, then the obstacle 2, in a clockwise direction. Finally it goes to the goal. When the robot starts from the position 2, it avoids the obstacle 1 and obstacle 2 in a clockwise direction. As seen in Fig 14, the robot is not affected by any of the passed obstacles and reaches the goal successfully.

6. Conclusion and Further Works

In this paper, the limit-cycle navigation method that is fast and efficient was proposed. The proposed method has two merits over the potential field met hod. First, it is not affected by obstacles that are not in its way. Second, there are not local minima that are major problems in the potential field meth od. These merits are ascertained through real expe riments. In this paper, although moving obstacles was not considered in this paper, when the robot i s to avoid moving obstacles, there are many aspec ts to be considered, e.g., the obstacle's moving dir ection and velocity. The robot will have to predict them. These will be further research

7. Acknowledgement

This work was supported by grant No. RTI04-01-02 from the Regional Technology Innovation Program.

REFERENCES

[1] Ronald C. Arkin, "*Motor Schema-Based Mobile Robot Navigation*," The International Journal of Robotics Research, vol. 8, no. 4, pp. 92-112, 1989.

[2] Ronald C. Arkin, "*Integrating Behavioral, Perceptual and World Knowledge in Reactive Navigation*," Robotics and Autonomous Systems, vol. 6, pp. 105-122, 1990

[3] Rodney A. Brooks, "*A Robust Layered Control System for a Mobile Robot*," IEEE Journal on Robotics and Automation, vol. RA-2, no. 1, March, 1986

[4] Marc G. Slack, "*Navigation Template: Mediating Qualitative Guidance and Quantitative Control in Mobile Robots*," IEEE Trans. on Systems, Man and Cybernetics, vol. 23, no. 2, March/April, 1993.

[5] Y. Koren and J. Borenstein, "*Potential Field Methods and their Inherent Limitations for Mobile Robot Navigation*," Proceedings of the IEEE Conference on Robotics and Automation, 1991.

[6] Erann Gat, "*Navigation Templates: Enhancements, Extensions and Experiments*," IEEE International Conference on Robotics and Automation, pp. 541 –547, vol.1, 1993

[7] D-H Kim, Y-J Kim, K-C Kim, Jong-Hwan Kim and P. Vadakkepat, " *Vector Field Based Path Planning and Petri-net Based Role Selection Mechanism with Q-learning for the Soccer Robot System*", Intelligent Automation and Soft Computing, Vol. 6, No. 1, P.75-88, 2000

[8] H.S Sim, M.J Jung, H.S Kim, Jong-Hwan Kim and P. Vadakkepat, "*A Hybrid Control Structure for Vision Based Soccer Robot System*", Intelligent Automation and Soft Computing, Vol. 6, No. 1, P.89-101, 2000

[9] Yong-Jae Kim, Jong-Hwan Kim and Dong-Soo Kwon, "*Evolutionary Programming-based Uni-vector Field Navigation Method for Fast Mobile Robots*," IEEE Trans. on Systems Man, and Cybernetics. (Accepted for Publication)

[10] Hassan K. Khalil, *Nonlinear Systems*, 2nd ed., Prentice Hall, 1996.