A Real-Time Streamline-based Obstacle Avoidance System for Curvature-Constrained Nonholonomic Mobile Robots

Pei-Li Kuo, Chung-Hsun Wang, Han-Jung Chou and Jing-Sin Liu

Abstract—This work presents a streamline-based strategy for curvature-constrained nonholonomic robots to safely navigate in real-world partially unknown environments with static cylinder-shaped obstacles whose locations are detected on-line. We propose the use of three primitive curvature-constrained collision-free paths derived from the manipulation and search of streamlines generated from harmonic potential function of a stationary circle. The primitive paths allow emergent turning and reduce the clearance with an obstacle. Assuming enough clearances between adjacent obstacles, the algorithm is extended to avoid multiple cylinder obstacles by pursuing a primitive path based on lateral distance of the robot and the closest obstacle, which can be calculated and updated in real time using pure pursuit tracker. Experiments validate that the obstacle avoidance system allows a two-wheel driving mobile robot to on-line navigate safely along a path without violating curvature constraint in partially unknown cluttered environments.

1. INTRODUCTION

Due to advances in sensing and computing technology, development and application of autonomous mobile robots are prevalent in academia and industry [1], [2]. Along with increasingly heavy interaction between human and robots and more autonomy, there have been a lot of real-time obstacle avoidance algorithms [2], [3], [4] to avoid static as well as dynamic obstacles in open space or in narrow passages that cause difficulty of local obstacle avoidance, such as human, objects, and other robots, in cluttered outdoor or domestic environments.

One of the most well-known obstacle avoidance method is the artificial potential field (APF) approach, first introduced by Khatib, to create a real-time collision avoidance path for manipulators and mobile robots [5] The goal position of the robot is assigned as an artificial attractive potential and obstacles are applied as artificial repulsive forces. Then, the collision avoidance path is derived by using the gradient of the total APF composed of linear superposition of each potential. The characteristic of APF-based approaches is robust and efficient while reacting with surrounding obstacles, however APF suffers from the local minima, i.e. positions where the robot gets stuck. In general, it is desired that no other local minima except the goal exists in APF so that the navigation could reach the goal. Harmonic potential functions (HPFs) [7]-[9], [11], [17] are a special type of APFs derived from the velocity potential of Laplace equation, which has no local minima in the interior of cluttered or bounded environments with Dirichlet boundary conditions (i.e. the motion on the obstacle boundary is along the normal direction of the obstacle boundary/wall) or Neumann boundary conditions (i.e. the motion on the obstacle boundary/wall is parallel to the tangential direction of the obstacle boundary). Laplace equation could be solved numerically using finite difference methods such as the Jacobi iteration, Gauss-Seidel iteration and SOR iteration methods in grid environment, and the gradient of the obtained potential values gives the streamline, or the velocity at each grid [7]. A log-space algorithm with GPU acceleration was proposed to fix the numerical precision problem of numerical solution of Laplace equation (a linear system of linear equations) via the finite difference methods [10].

Notably, HPF-based path planner is a complete [7] and anytime algorithm [10], in which the streamlines cover the free regions of the workspace so that a robot modeled as a point particle could smoothly reach the target without collision with the (circular) obstacles [6]-[9], [11] by following a streamline. Motion planning based on fluid dynamics applies different fundamental elements such as a point sink (representing the goal), a point source (representing the robot location), or a uniform flow (defined as a flow with constant speed) plus a doublet (representing the obstacle), and their superposition, to create a new HPF [11]-[15] in multiple obstacle cases. The path corresponds to the streamline of the flow with velocity defined by weighted superposition of velocity which is induced by a single obstacle. Kim and Khosla [15] developed a motion planning and control strategy based on solving harmonic functions with panel method for the fluid flow around an arbitrarily shaped obstacle. Wang et al. [16] introduced a reactivity parameter to adjust the amplitude of the path’s deflection around an obstacle and an optimal 3D path is obtained by genetic algorithm.

Kinodynamic motion planning [22], in particular curvature-constrained motion planning, takes into account the robot’s dynamic capabilities and allows mobile robots to navigate safely and comfortably even in high speed along a trajectory compatible with the kinodynamic constraints of the motion. Nonholonomic robots motion planning in multiple obstacle environments is challenging since motion planners have to simultaneously control the heading and the speed and deal with collision avoidance and nonholonomic constraint [17]. With the advantage of smooth trajectories generated by APF approaches, Lau et al. [18] provides a streamline-based kinodynamic motion planning approach to avoid elliptical obstacles, guaranteeing both velocity and curvature are within limits by adjusting the strength of a source and a sink if a portion of trajectory violates kinematic constraints.

Pei-Li Kuo, Han-Jung Chou and Jing-Sin Liu are with Institute of Information Science, Academia Sinica, Taipei, Taiwan, R.O.C. email: r01525004@ntu.edu.tw, r0221012@gmail.com, liu@iis.sinica.edu.tw.
Chung-Hsun Wang is with Institute of Information Science, Academia Sinica, Taipei, Taiwan, R.O.C., and Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C. email: b02901014@ntu.edu.tw.
This paper presents a real-time planning and control strategy for circular nonholonomic robots to navigate in real-world partially unknown environments cluttered with cylinder-shaped obstacles. An obstacle avoidance system using primitive streamline-based paths is proposed and developed further by implementing on Dr. Robot X80 robot, which detect obstacles by sonar and infrared sensors attached in front of the robot in unknown areas. The motion planner generates an initial collision-free streamline path and determines on-line whether to update primitive path according to the sensors data. The new primitive path is updated with new sensor information and the change to a new streamline is enabled by exploiting the richness of streamlines, which could be computed off-line based on a prior obstacle information (distribution, i.e. shape, size, location and number). Pure pursuit algorithm is implemented to enable a smooth transition without violating the curvature constraint from the current robot position in one streamline to a selected goal in a new streamline in real time.

The paper is organized as follows. Section II gives a brief introduction of a mobile robot with two independently driven wheels for the experiment. Section III mentions the harmonic potential field approach for avoidance of a cylinder obstacle. Then, we present three primitive paths based on streamlines and a distance-based path selection strategy of a primitive path for nonholonomic mobile robots. Section IV proposes a new real-time obstacle avoidance framework using primitive paths and pure pursuit algorithm. Experimental results in a simple environment are presented in Sect. V. Sect. VI ends with conclusion of the paper.

II. THE MOBILE ROBOT

A. Wheeled mobile robot system

A wireless mobile robot platform Dr. robot X80 is developed for our real-time planning and control algorithm. Fig. 1 shows the physical and sensor configurations of the mobile robot and the front view of the mobile robot. The X80 mobile robot is an integrated electronic and software robotic system, and it is available for designers through a set of ActiveX control components (SDK) developed for C/C++. The navigation algorithm runs directly on the remote PC through wireless communication.

B. Kinematic model

The mobile robot is controlled by two wheels driven by DC motors independently, where the differential velocity is used to drive moving direction. Fig. 2 illustrates the kinematic model, where \((x, y), \theta\) denote the coordinates and heading of the center point of the differential-drive circular mobile robot of radius \(r_{\text{Robot}}\). The kinematics describing the rolling without slipping of wheels can be expressed as the nonholonomic unicycle (1).

\[
\begin{bmatrix}
  \dot{x} \\
  \dot{y} \\
  \dot{\theta}
\end{bmatrix} =
\begin{bmatrix}
  \cos \theta & 0 & v \\
  \sin \theta & 0 & 0 \\
  0 & 1 & \omega
\end{bmatrix}
\begin{bmatrix}
  r/2 & r/2 & w_L \\
  -r/d & r/d & w_R
\end{bmatrix}
\begin{bmatrix}
  v \\
  \omega
\end{bmatrix}
\]  

(1)

where \(v\) and \(\omega\) denote velocity and angular velocity, respectively; \(w_L\) and \(w_R\) denote left and right wheel velocity, respectively; \(r\) is wheel radius; \(d\) is distance between two wheels. The detailed parameters are listed in Table 1.

C. Obstacle detector

Fig. 3 shows the sensor configuration. DUR5200 Ultrasonic Sensor and GP2Y0A21YK Sharp Infrared Sensor are equipped in the mobile robot. There are three sonars (Sonar 1, Sonar 2, and Sonar 3) and four infrared sensors (IR 1, IR 2, IR 3, and IR 4) in the front part of the mobile robot, where \(\theta_1\) is 12°, \(\theta_2\) is 18° and \(\theta_3\) is 15°. The ultrasonic sensor can measure a distance range from 4 to 255cm, while the detecting distance range of IR sensor is between 10 and 80 cm. Both sensors’ update rate is 10 Hz. If only a single sensor detects an obstacle and receives value, then we assume the obstacle is located in the direction of the sensor. Otherwise, when two sensors detect an obstacle at the same time, we assume that the obstacle lies on the angle bisector of these two sensors’ directions. Fig. 4 shows the detection of obstacle. For instance, if Sonar 2 detects an obstacle, the robot knows the obstacle is located in front of the robot with azimuth angle 0°. Likewise, Sonar 1, Sonar 3, IR 1, IR 2, IR 3 and IR 4 detect the obstacle with azimuth of 45°, −45°, −30°, −12°, 12°, and 30° respectively. Otherwise, if both Sonar 2 and IR 3 detect an obstacle, then the obstacle’s direction will be 2°, which is between Sonar 2 and IR 3. The estimated obstacle location is
transferred into the global frame for the motion planner. The estimation error is accommodated by a safety distance $r_{\text{Safe}}$ in practical implementation of navigation system.

III. OBSTACLE AVOIDANCE MODEL BY HARMONIC POTENTIAL FIELD AND PURE PURSUIT CONTROLLER

We briefly summarize the $C^2$ smooth path produced by streamlines of harmonic potential field and then introduce three primitive paths and pure pursuit algorithm for real-time obstacle avoidance. We assume the obstacles are far apart.

A. Mathematical model of the artificial potential field

Harmonic potential functions are solutions to Laplace’s equation, so functions generated by Laplace’s equation do not exhibit local minima [7]. In two-dimensional space, the velocity potential $\phi$ is a solution of Laplace’s equation $\nabla^2 \phi = 0$, governing the flow of the nonviscous, incompressible fluids. A streamline indicates local flow direction: its tangent at every point is in the direction of local fluid velocity (vector field) associated with the flow defined as

$$u = \nabla \phi$$

We assume that the robot and the obstacle are modeled as a circle defined by a radius $r_{\text{Robot}}$, $r_{\text{Obstacle}}$, respectively. The distance between the robot’s center and the obstacle’s center is $D = D_{\text{Sensor}} + r_{\text{Obstacle}} + r_{\text{Robot}}$, where $D_{\text{Sensor}}$ is the range value measured by the sensor, $r_{\text{Obstacle}}$ is the radius of the obstacle, and $r_{\text{Robot}}$ is the radius of the robot. Given a safety distance $r_{\text{Safe}}$ between a robot and an obstacle, the radius of obstacle is enlarged by taking account of the robot radius as $r_{\text{Obstacle}} = r_{\text{Obstacle}} + r_{\text{Robot}} + r_{\text{Safe}}$. For a circular robot and a circular obstacle, the collision free criterion for safe navigation is that the distance between the point robot (robot center) and the obstacle center is larger than $r_{\text{Obstacle}}$.

Consider a mobile robot at $x = [x, y]^T$ moves in the +x-axis direction with speed $U$ to avoid a circular obstacle of radius $r$ located at origin, the velocity potential field $\phi(x, y)$ can be represented as the superposition of an uniform rectilinear flow and a doublet [9], [14], [19] as

$$\phi(x, y) = Ux + \frac{A}{x^2 + y^2}$$

where constant $A = Ur^2$. According to (2), the robot’s velocity in Cartesian coordinates is

$$u = \frac{\delta \phi}{\delta x} = u_x = \frac{2y^2A}{(x^2 + y^2)^2} - \frac{A}{x^2 + y^2}, \quad v = \frac{\delta \phi}{\delta y} = \frac{2Ax y}{(x^2 + y^2)^2}$$

In practice (and in our experiments in Sec. V), we assume constant robot speed $U$, so the velocity in (4) is normalized to unity while its direction is preserved. Then, normalized velocity, acceleration, curvature of each point on the streamline in the uniform flow are given by (5), (6) and (7).

$$u_N = \frac{u}{\sqrt{u_x^2 + v_y^2}}, \quad v_N = \frac{v}{\sqrt{u_x^2 + v_y^2}}$$

$$a_x = \frac{\partial u_N}{\partial x}u_N + \frac{\partial u_N}{\partial y}v_N, \quad a_y = \frac{\partial v_N}{\partial x}u_N + \frac{\partial v_N}{\partial y}v_N$$

$$\kappa = \frac{u_x a_y - v_y a_x}{(u_N^2 + v_N^2)^{3/2}}$$

From (5), given a start position, a streamline can be derived by numerical integration of the velocity vector that specifies the tangent of the path or the robot heading.

B. Three primitive paths with curvature constraint

For local obstacle avoidance, three collision-free curvature-constrained primitive paths that most aligned to the current robot’s heading are proposed based on the richness of streamlines. First, a point robot could circumvent an obstacle with maximum curvature from its left or its right side. Second, it could also pass an obstacle with maximum curvature via a sharp turn. An initial streamline is chosen based on initial robot configuration and a priori known obstacle distribution, taking into consideration of curvature constraint. This initial path may collide with obstacles. To ensure safe navigation, one of our obstacle avoidance avoidance strategy is to make the robot change from one streamline to another streamline at a lookahead distance via a local, on-line pure pursuit algorithm without violating curvature constraint. Details are as follows. For simplicity, we assume that the robot is moving in the +x-direction and a circular obstacle of radius $r$ is located at origin, so that the maximum curvature of streamline occurs at the $y$-axis.

1) Sharp left or right turns

Consider the avoidance of the closest obstacle in front of the robot. Two curvature-constrained streamline-based turns
paths symmetrical with respect to the line connecting robot center and obstacle center could be used as two primitive paths on left or right side of the obstacle based on the obstacle’s radius. We search the two streamlines corresponding to left turn and right turn with curvature maximum equal to maximum curvature $\kappa_{\text{max}}$. The desired streamline is obtained by shifting the selected streamline parallelly until its curvature maximum point grazes the obstacle boundary.

2) **Minimum curvature turn**

We exploit the property that the deflection and curvature of a streamline become smaller as it is farther from the obstacle. First, search the streamlines whose maximum curvature is not larger than $\kappa_{\text{max}}$ farther from the obstacle than the current streamline the robot stays to find the first streamline when shifted to the robot current location is collision-free. This streamline is used as the minimum curvature turn path.

C. Distance-based path selecting strategy

The path selecting strategy depends on the upcoming obstacle position or lateral displacement relative to the obstacle. It is illustrated in Fig. 5. The robot is initially located at $x = -2$ with different lateral distance $y$ related to the a cylinder obstacle at the origin. Let points $b^+$ and $b^-$ be the points which two sharp turn paths intersect with the line $x = -2$. The interval $[r, r]$, denoted by $[d_l, d_u]$ in Fig. 6 at the vertical line $x = -2$ is partitioned into intervals $B^+ \sim B^-$ by the labeled points $d_l \sim d_u$ according to the start points of the primitive paths, symmetrically with respect to the robot current position. We define $L_{\text{sharp}}$ as the distance between points $c^+$ (the start point with right sharp turn path) and $c^-$ (the start point with minimum curvature path).

![Fig. 5](image_url)

**Fig. 5** Concept of primitive path selecting strategy. Robot with different lateral displacements related to an obstacle will pursue different primitive paths symmetrical with respect to the line connecting robot center and obstacle center. Path 1 and Path 2 are tangent to current robot heading. Path 1 and Path 2 are tangentially traverse the enlarged circular obstacle boundary.

Specifically, the path selecting strategy is decided according to the relative lateral distance of robot with the obstacle as

$$
\begin{align*}
|L_{\text{obs}}| > L_{\text{sharp}} & \rightarrow \text{Path 1} \\
L_{\text{r}} \geq L_{\text{obs}} > 0 & \rightarrow \text{Path 2} \\
0 \leq L_{\text{obs}} - L_{\text{sharp}} & \rightarrow \text{Path 3}
\end{align*}
$$

(8)

This strategy is designed to use a minimum curvature turn in Intervals $A^+ \sim A^-$, while it pursues a sharp left turn in Interval $B^+$ and a sharp right turn in Interval $B^-$. Once an obstacle is sensed by the obstacle detector, the motion planner determines the proper primitive path for the mobile robot to follow. Note that for obstacles with the same radius, the turn paths from a given position generated by streamlines of Laplace equation are identical. Hence, we can store the streamline-based turn paths for obstacles with different radii in the dataset. The path obtained by transforming a sample path with obstacle located at origin to the estimated obstacle position could then be reused to on-line plan the collision-free movement with reduced time-complexity.

IV. REAL-TIME OBSTACLE AVOIDANCE STRATEGY

A. Overview of the obstacle avoidance system

Fig. 6 depicts the building blocks of the obstacle avoidance system. In robot hardware part, we used robot’s kinematic model to estimate robot’s own kinematics and sensors to detect surrounding environments. The real-time obstacle avoidance system is built by three subsystems, which are the obstacle detector, the motion planner that incorporates curvature constraint, and the pure pursuit controller used to control the robot to follow the specific path primitive with allowable angular velocity satisfying curvature constraint. The obstacle detector receives range data from sonar and infrared sensors, and then estimates obstacle global location. Our motion planner initially selects a streamline starting from the robot start position, which is generated based on a priori-known obstacle distribution. When the obstacle detector updates obstacle’s location based on new sensor readings during robot forward motion, the motion planner will decide whether to enable local re-planning or to retain original path for the robot to follow. Local re-planning is performed by generating and updating a local subgoal that is reached by a primitive collision-free path via pure pursuit.

B. Pure pursuit controller for mobile robots

Pure pursuit is a path tracking method enabling a nonholonomic mobile robot to simultaneously plan a path and follow the planned path. Due to the fact that nonholonomic robots cannot directly move in the lateral direction, a robot pursues a goal position ahead of the robot from its current position along an arc of curvature $\kappa$ via pure pursuit [20], [21]. Fig. 7 illustrates the concept of pure pursuit controller. An algorithm is shown in Algorithm 1. Once a primitive path is generated, the next step is to find a goal point $x_g$ which is located ahead the closest point. Let a local coordinate system be attached to the robot with its origin set as rotation center of the robot and +x-axis of the local frame aligned with the forward motion direction. Then transform a goal point $X_g$ in the global coordinate to $X_g_{\text{Robot}} = [X_g_{\text{Robot}} X_y_{\text{Robot}}]^{T}$ in local frame with $X_g_{\text{Robot}}$ and $X_y_{\text{Robot}}$ denoting the longitudinal and the lateral displacements, respectively.
\( x_{g,\text{Robot}} = R(\theta)(X_{g} - X_{\text{Robot}}) \)  

where \( X_{\text{Robot}} \) is current robot position in global frame, and \( \theta \) is the heading angle of robot in global frame. The goal point to pursue keeps a specific lookahead distance \( L = \sqrt{X_{g,\text{Robot}}^2 + X_{g,\text{Robot}}^2} \), since nonholonomic robots cannot correct errors directly with respect to the nearest point on the path. The curvature \( \kappa \) of the robot is defined as the inverse of radius of curvature \( r \), i.e. the distance between the origin of the robot frame and its instantaneous center of rotation. In addition, curvature can also be defined as the instantaneous change of the heading angle \( \Delta \theta \) with respect to the travel distance \( U \Delta t \) in sampling time \( \Delta t \). Curvature then could be further related with robot’s velocity and angular velocity. Therefore, the angular velocity of a robot moves along a path at a constant speed \( U \) could be computed from the path curvature via the relation

\[
\kappa = \frac{1}{r} = \frac{\omega}{U} = \frac{\Delta \theta}{U \Delta t},
\]

where \( \kappa = \frac{\omega}{U} \) [20]. To satisfy the maximum allowable curvature, we regularize the curvature as

\[
\kappa_{\text{constraints}} = \text{Sign}(\kappa) \cdot \kappa_{\text{max}} \quad \text{if} \quad \kappa > \kappa_{\text{max}}
\]

Thus, the motion in the local frame of the robot can be applied to command motion controller via the inverse kinematics of (1). Then, the displacement \( \Delta x_{g} \) in the global frame can be derived from the displacement \( \Delta x_{\text{Robot}} \) in the local frame

\[
\Delta x_{g} = R(\theta + \Delta \theta) \Delta x_{\text{Robot}}
\]

V. EXPERIMENT RESULTS AND DISCUSSION

In experiments, three aspects related to the feasibility of trajectory are presented, which are (1) curvature constraints, (2) arrangement of obstacles, and (3) sensor detection error. First, while maximum curvature constraint decreases, it’s more difficult for robots to achieve primitive paths. Second, for all three primitive paths, a robot needs to keep enough distance from the obstacle to enable pursuit primitive paths. Third, we rely on sonar and infrared sensors to estimate obstacle location. Furthermore, for guaranteed obstacle avoidance by the streamline-based approach, the obstacles are arranged so that only one obstacle is detected within a prespecified lookahead distance of the mobile robot current location at a time.

A. Experimental setting

The mobile robot is initially located at origin of the global coordinate system and its forward moving direction is + x axis with constant linear velocity \( U = 0.5 \text{ m/s} \). The environment is partially known in that the shape of obstacle is assumed known as cylinder, and several location-unknown but identical cylindrical obstacles with radius \( r_{\text{Obs}} = 0.1 \text{ m} \) are located around the forward motion route. We assume that the clearance between any two adjacent obstacles is smaller than the sensing range of the sensors but is large enough to allow the application of pure pursuit algorithm to generate a local collision-free path. Hence, the enlarged radius of an obstacle \( r_{\text{Obs}} = r_{\text{Robot}} + r_{\text{Obs}} + r_{\text{Safe}} = 0.4 \text{ m} \). In addition, the minimum distance between two obstacle’s center is \( 2 r_{\text{Obs}} \) which is wide enough for the robot to pass between two obstacles. The maximum allowable curvature for the mobile robot is set as \( 1.5 (1/\text{m}) \), as shown in Table 1. The pure pursuit command rate is 10 Hz, and the safety distance is 0.1 m, which is the robot displacement in a period. Lookahead distance is the only parameter important for a tradeoff of stability and tracking performance in pure pursuit algorithm. A longer lookahead distance results in a smoother but a late response, and a shorter lookahead distance reduces tracking error more quickly but could be unstable due to curvature constraint. In our experiment, the lookahead distance is the robot radius 0.2 m.

B. On-line static cylinder obstacles avoidance

The experimental setup and the trajectory are depicted in Fig. 9 with the velocity field of the Laplace equation shown in Fig. 8. The map of the environment is created by Hector SLAM, an open source SLAM algorithms available in ROS [23]. Similar to [11], there are four cylinder obstacles placed at \((1, 0), (1.8, -0.6), (2.6, 0)\) and \((2.6, -1.2)\) in meter. We assume the obstacle shape projected onto the ground is known to be circles of the same radius, but the number of static obstacles and their locations are unknown. The primitive paths can be computed within 0.2 milliseconds in our implementation. The reaction of the robot that is capable of localizing itself autonomously is immediate once new sensor readings are available (within 1 millisecond). In the experiments, two feasible smooth paths shown in Fig.9 are found for on-line safe navigation. Different paths are obtained due to the slight variation in initial position and heading of the mobile robot.
This paper presents a real-time streamline-based obstacle avoidance system for curvature-constrained nonholonomic robots in partially unknown environments. Primitive paths are searched from the modified streamlines of harmonic potential functions of single circular obstacle so as to satisfy curvature constraint, to allow emergent turning and to reduce the clearance with an obstacle. Different primitive paths are selected online based on the lateral displacement of current robot position with the closest obstacle. Pure pursuit controller is modified to enable path tracking within curvature constraint. Experiments were conducted in a simple partially unknown environment to validate the safe navigation. Future work is toward the guaranteed safe navigation of nonholonomic mobile robots in more complex environments with the anytime, complete real-time HPF-based path planner.

REFERENCES


