Extending Collision Avoidance Methods to Consider the Vehicle Shape and Kinematics of a Mobile Robot Using ARM Processor

G.Arunprasad [1], M.Balakarthikeyan [2], Dr.B.Chidhambararajan[3]

[1,2]Asst.Prof, Department of Electrical and Electronics Engineering,
VelTech MultiTech Dr.R.R & Dr.S.R.Engineering College, Avadi, Chennai.
[3]Principal
Madurai Institute of Engineering & Technology, Madurai, Tamil Nadu, India.

ABSTRACT

Majority of collision avoidance methods does not consider the vehicle constraints mentioned. They assume a point-like and omni directional vehicle with no acceleration constraints. The main contribution of this work is a scheme to consider the exact shape and kinematics, as well as the effects of dynamics in the collision avoidance layer. The idea is to abstract these constraints from the usage of avoidance methods. This technique can be applied to many vehicles with arbitrary shapes.

***Index Terms*—Mobile robots, motion constraints, reactive collision avoidance.**

I.INTRODUCTION

One of the main objectives of indoor mobile robotics research is to build robots that can safely carry out missions in hazardous and populated environments. For example, a service-robot that assists humans in indoor office environments should be able to react rapidly to unforeseen changes, and perform its tasks under a wide variety of external circumstances.

Most of today's commercial mobile devices scale poorly along this dimension. Their motion planning relies on accurate, static models of the environments, and therefore they often cease to function if humans or other unpredictable obstacles block their path. To build autonomous mobile robots one has to build systems that can perceive their environments react to unforeseen circumstances, and (re)plan dynamically in order to achieve their missions.

This paper focuses on one particular aspect of the design of such a robot: the reactive avoidance of collisions with obstacles with both dynamic and kinematics constraints. In order to endow vehicles with true versatility, they must execute tasks autonomously in unknown, unstructured, dynamic and unpredictable environments. Under these circumstances, motion must be generated by an obstacle avoidance method driven by sensory information. An obstacle avoidance method is a procedure that, given a sensorial measurement (obstacle description) and a final position, calculates a collision free motion towards a target. It works within a perception - action cycle where the motion is executed by the vehicle and the process restarts .The result is an on-line motion sequence that drives the vehicle to the target while avoiding collisions. The avoidance task is further complicated since many robots have shape and kinematics constraints that limit motion.

The study described here centres on the consideration of the vehicle shape, as well as kinematic and dynamic constraints, during the application of a collision avoidance method. The idea is to project distance measurements into a space in which the robot can be regarded as a holonomic point. The projection accounts for collision constraints as well as for kinematics and dynamic motion constraints (the trajectories are restricted to a family of circular arcs). In this space, many reactive collision avoidance methods can be applied to the holonomic point, as all constraints are encoded in the obstacles and space itself. The computed motion command is projected back and applied to the robot. Therefore, the proposed method encompasses a complete set of well-known obstacle avoidance approaches to consider the vehicle shape, as well as the kinematics and dynamic constraints. This method has been demonstrated in real-world experiments by wrapping a potential field method to perform obstacle avoidance on a differentially-driven wheelchair.

II.OBSTACLE AVOIDANCE METHOD

Majority of collision avoidance methods does not consider the vehicle constraints mentioned. They assume a point-like and omni directional vehicle with no acceleration constraints. The main contribution of this work is a scheme to consider the exact shape and kinematics, as well as the effects of dynamics in the collision avoidance layer. The idea is to abstract these constraints from the usage of avoidance methods. This technique can be applied to many vehicles with arbitrary shapes. The construction of this abstraction layer comprises three parts that correspond to the three contributions of this study.

First, the 2-D manifold of the 3-D configuration space defined by elemental circular paths is constructed, centred on the robot. This manifold contains all the configurations that can be reached at each step of the obstacle avoidance. The contribution is the exact calculation of the obstacle representation in this manifold for any vehicle shape. In this manifold, a point represents the vehicle.

Second, the exact calculation of the admissible configurations is described, which result from the obstacle regions computed previously (with the assumption that the braking path is a circular elemental path, typical in obstacle avoidance). Furthermore, the reachable configurations obtained by reachable commands in the manifold are represented. The effect of dynamics is represented in the manifold.

Third, a change of coordinates in the manifold is proposed so that the circular paths become straight segments. With the manifold represented in such coordinates, the motion is free of kinematics constraints. As a result, the 3-D collision avoidance problem with shape, kinematics, and dynamics is transformed into a simple problem of moving a point in a 2-D space with no constraints. Thus, methods that ignore these constraints become applicable.

We characterize next a feasible circular motion. In the robot system of reference, an admissible circular path contains the origin, and the instantaneous turning centre is on the *Y*-axis. Then, if (*x*, *y*) is a point in the workspace, there is only one circle going through {(*x*, *y*), (0, 0)} and having its centre in the *Y*-axis. The radius of that circle is:

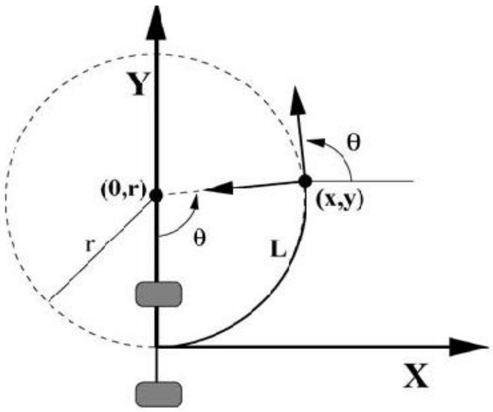

Fig.1 Plane showing the obstacle(x,y)

$$R = \frac{X^2 + Y^2}{2Y}$$

Such that by sensory information we can identify (x, y) and by means of (x, y) we can identify R, θ

We have discussed how the admissible elementary paths of the vehicles considered are circles. We identified the manifold of the configuration space, as a function, which represents all the configurations reachable under circular motions. We also provided a calculus to compute the exact bounds of the collision region. In this section we propose a change of coordinates so that elementary paths become straight segments with the new coordinates. The Ego-Kinematics change of coordinates transforms the domain of the manifold R2 into R × *S*1,

$$(x, y) \rightarrow (L, \alpha)$$

Where the distance to a point is the arc length L measured over the circle that reaches that point, and the angle univocally represents this circle. Next, we discuss the computation of both coordinates. In the robot system of reference, the radius r of the circle that goes through point (x, y) and the vehicle orientation θ .The distance to the point measured along the circle is the arc length:

$$L = \begin{cases} |x|, & y = 0 \\ |r \cdot \theta|, & y \text{ not equal to '0'} \end{cases}$$

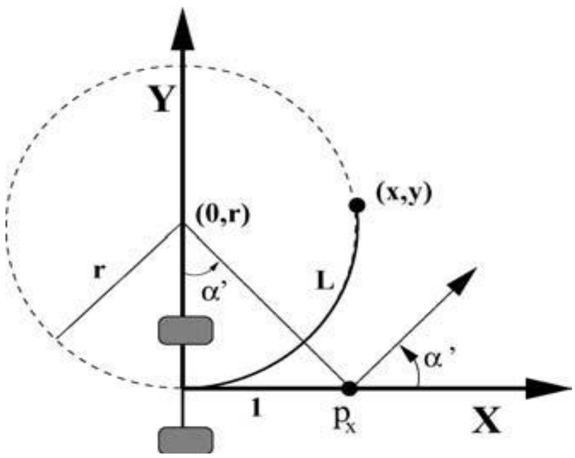

Fig2: This figure shows how a differential-drive vehicle reaches a point of the space (*x*, *y*) by a circular path (of radius *r*).On a point of the *X*-axis, the angle *α* is tangent to the circle.

This distance is the first coordinate. The second coordinate has to identify the circle univocally and give the sense of travel. The turning radius r is a unique descriptor of the circle going through a point and that complies with the motion constraints. However, this descriptor is unbounded while we search a bounded representation. This is achieved through an angular variable, constructed as follows. Let px be a point in the X-axis (for example the (0, 1)). Let T be the line joining (0, r) and px.Then, α is the angle comprised between the perpendicular line to T and the X-axis:

$$\alpha' = \arctan(1/R)$$

This definition implies that when $x \geq 0$ the direction of travel is "forward" (the angle α is equal to α), and when $x < 0$ the motion is "backward" (α is the result of the same calculus assuming the symmetrical problem with respect to the Y-axis). Notice that each value of αs univocally determines a turning radius, Rs

Rs = 1/tan (αs) For αs=+90 to -90

= 1/ tan (sign (αs).π – αs) otherwise

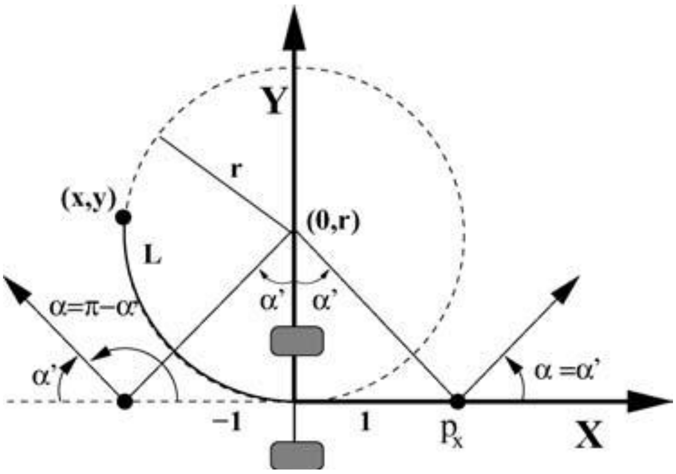

Fig3: This figure shows how, where the point is located in the positive *X*-axis, the value of *α* is *α*. If the point is in the negative axis, the problem is symmetrical with respect to the *Y*-axis.

Thus the coordinate α distinguishes the direction of travel cos $\alpha s \geq 0$ is "forward" motion while the opposite is "backward" (although r and α do not differentiate the direction).

In summary, we represent ARM in a new coordinate system where the motion is omni directional (without kinematics constraints) whereas it represents a motion over an admissible path for the robot

I. ABSTRACTION OF THE SHAPE AND KINEMATICS CONSTRAINTS FROM THE OBSTACLE AVOIDANCE METHOD

In this section we use the previous results to abstract the shape and the kinematics of the vehicle from obstacle avoidance methods. These techniques work within a cycle, computing on-line collision free motion given a description of the obstacles and a destination. The motion is executed by the vehicle and the process restarts. The idea is to build an abstraction layer so that the solutions computed consider the shape and the motion constraints of the vehicle without redesigning the method. This is achieved by including two stages:

(1) Incorporate the shape and the kinematics before the method application and

(2) Motion computation.

At each iteration the procedure is:
Shape: Construction of the region in collision with the obstacles.

Kinematics: Change of coordinates of the collision region.

Obstacle avoidance*:* Application of the obstacle avoidance method, to compute the most promising Motion direction αs

Motion: The direction solution αsol is transformed into a motion command (v, w) as follows. First, we compute the radius solution rsol. Then, we compute a command (v, w) that preserves the turning radius v =w.rsol.

Any command on the line with slope rsol is valid. One strategy to select one command is to reduce the module of the speed vector mv as a function of the distance to the closest obstacle:

mv = mv max , for dobs ≥ dmin
= mv max. dobs/ dmin , otherwise

Where mv max is the distance from the velocity origin to the bound of the rectangle of maximum velocities, dobs is the distance to the closest obstacle in, and dmin is a distance threshold to check whether the velocity is maximum.

(V, w) = (mv · cos γsol, mv · sin γsol), for αs=+90 to -90
= (−mv · cos γsol, mv · sin γsol) , otherwise

Where γsol = arc tan (1/*r*sol).

III.EXPERIMENTAL RESULTS

The objective of this section is to validate our methodology with two obstacle avoidance methods working on a real vehicle with shape (square) and kinematics constraints (differential-drive).

The robot is square (0.8 × 0.8m) differential-drive. In order to collect information about the obstacles, the vehicle was equipped with 3 IR sensors for the view of 180 degrees.
All calculations were carried out on ARM LPC 2148.

In the experiments, two aspects were tested:

1) The collision avoidance task was carried out with the method using the abstraction layer. The vehicle was driven to the target while collisions with obstacles were avoided;

2) The computed motion considered the shape, kinematics, and dynamics of the vehicle.

General Obstacle Avoidance Task With Abstraction:

Two experiments carried out in scenarios in which obstacles were randomly placed in order to hinder the wheelchair motion (unknown, dynamic, unpredictable, and unstructured scenarios). The difference between the experiments was the settings: Experiment 1 had higher obstacle density (more difficulty to maneuver), while Experiment 2 was more dynamic (unpredictable). In both cases, the vehicle reached the target location without collisions. The introduction of the abstraction layer did not penalize the work of the method in avoiding obstacles. Shape, kinematics, and dynamics of the vehicle were taken into account at all times during the experiment. As a result, the vehicle successfully achieved the avoidance task. Notice that while ignoring such constraints, the obstacle avoidance with this vehicle could have been heavily penalized and it is doubtful that it could reach the target otherwise.

2) Shape, Kinematics, and Dynamics in Obstacle Avoidance:

Next is the description of how the vehicle restrictions were taken into account during the experiments. The commands computed by the method were always kinematically admissible, as they resulted from admissible circular paths. This occurred because the avoidance method was applied to all directions, where directions corresponded to a turning radius. The motion command solution is the command that performs this turn.

In order to address the vehicle dynamics, the method computes commands that are reachable in a short period of time, also taking into account the braking distance. The computed commands are reachable because the avoidance method computes a direction solution βsol, which is then used to select a location in RCP

As a consequence, the vehicle executed the planned motion strictly. The motion commands assure that the vehicle can be stopped without collision by applying maximum deceleration

IV. CONCLUSIONS

This study presented a general scheme to extend collision avoidance methods for addressing shape, kinematics, and dynamics of the vehicle. The most important aspect of this study is its generality. With this framework, existing methods can be reutilized on a wide variety of any-shape nonholonomic vehicles, without any extra design or implementation effort.

REFERENCES

[1] J. Alvarez, A. Shkel, and V. Lumelsky, "Building topological models for navigation in large scale environments," presented at the IEEE *Int. Conf. Robot. Autom, Leuven, Belgium, 1998.*

[2] K. Arras, J. Persson, N. Tomatis, and R. Siegwart, "Real-time obstacle avoidance for polygonal robotswith a reduced dynamic window," in *Proc. IEEE Int. Conf. Robot. Autom., Washington, DC, 2002, pp. 3050–3055.*

[3] J. Asensio and L. Montano, "A kinematic and dynamic model-based motion controller for mobile robots," presented at the *15th IFAC World Congr, Barcelona, Spain, 2002.*

[4] A. Bemporad, A. D. Luca, and G. Oriolo, "Local incremental planning for car-like robot navigating among obstacles," in *Proc. IEEE Int. Conf. Robot. Autom, Minneapolis, MN, 1996, pp. 1205–1211.*

[5] J. A. Beyanas, J. Fernandez, R. Sanz, and A. Dieguez, "The beamcurvature method: A new approach for improving local real-time obstacle avoidance," presented at the 15th *IFAC World Congr., Barcelona, Spain, 2002.*

[6] J. Blanco, J. Gonzalez, and J. Fernandez-Madrigal, "Extending obstacle avoidance methods through multiple parameter-space transformations," *Auton. Robots*, vol. 24, no. *1, pp. 29–48, Jan. 2008.*

[7] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Trans. Syst., Man, Cybern., vol. 19, no. 5, pp. 1179–1187, Sep. /Oct. 1989.*

[8] J. Borenstein and Y. Koren, "The vector field histogram—Fast obstacle avoidance for mobile robots," *IEEE Trans. Robot. Autom, vol. 7, no. 3, pp. 278–288, Jun. 1991.*

[9] O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," in *Proc. IEEE Int. Conf. Robot. Autom. Detroit, MI, 1999, pp. 341–346.*

[10] O. Brock and O. Khatib, "Real-time replanning in high-dimensional configuration spaces using sets of homotopic paths," in *Proc. IEEE Int. Conf. Robot. Autom, San Francisco, CA, 2000, pp. 550–555.*

[11] D. Conner, H. Choset, and A. Rizzi, "Integrated planning and control for convex-bodied nonholonomic systems using local feedback control policies," presented at the *Robot.: Sci. Syst., Philadelphia, PA, and Aug. 2006.*

[12] W. Feiten, R. Bauer, and G. Lawitzky, "Robust obstacle avoidance in unknown and cramped environments," in *Proc. IEEE Int. Conf. Robot. Autom, San Diego, CA, 1994, pp. 2412–2417.*

[13] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Autom. Mag., vol. 4, no. 1, pp. 23–33, Mar. 1997*