A Path Tracking Method For Autonomous Mobile Robots Based On Grid Decomposition

A. Pozo-Ruz*, C. Urdiales, A. Bandera, E. J. Pérez and F. Sandoval

Dpto. Tecnología Electrónica.
E.T.S.I. Telecomunicación, University of Málaga.
Campus de Teatinos, 29071-Málaga
Spain

Abstract. This paper presents an algorithm for path following applications represented by grid decomposition in order to provide smooth tracking. In the first stage, the method calculates the corners that define the path based on a vision algorithm proposed by [2], so a new reference trajectory is composed of a sequence of straight lines. Then, the reference path is divided into a set of equispaced points. Using this new path, the path tracking algorithm computes the steering commands for the robot employing the pure pursuit algorithm [1]. This technique is simple and easy to implement. The full method of path planning and path tracking has been successfully implemented in the Nomad 200 mobile robot.

1 Introduction

The control system of an autonomous robot involves several functions such as perception, path planning and path tracking, that are strongly interconnected.

The ability of a real vehicle to track a path depends on the characteristics of the path, so continuity in position, orientation and curvature are basic requirements to be satisfied by the path generator [8]. However, smooth variation in curvature is still needed because of the relation with the characteristics of the required steering motion.

In [11] the authors presented a complete hierarchical path planning method for mobile robots based on pyramidal structures, traditionally employed in image processing [3]. Experimental results showed that the system improved the global performance of most path planning algorithms and allowed the calculation of optimal paths in huge environments in real time applications. However, the path provided by this method consists of a list of cells where the robot might pass, that is, they represent an area instead of a position. Thus, an algorithm for efficient path tracking is needed.

In order to provide a list of desired points that form the path, the centre of every cell could be computed. In this case, the trajectory obtained is very shaky and noisy, and even impossible to follow a determinated velocity [10]. Another possibility which could be considered is the calculation of a simplified path, by obtaining the main corners from the reference path (the list of cells) and connecting two consecutive vertexes by straight lines [4]. But this causes the robot to go straight on, stop and change its orientation each time it runs a segment, in other words, this soluction does not provide the desired continuous tracking.

In order to solve this problem, a low computational method for path following applications represented by grid decomposition that provides smooth tracking is presented in this paper. As a first step, the method proposes the calculation of the corners that define the path based on a vision algorithm presented by [2], so a new reference trajectory is composed of a sequence of straight lines. Then, the reference path is discretized into a set of equispaced points. Using this new path, the path tracking algorithm computes the steering commands for the robot employing the pure pursuit algorithm [1]. This technique is simple and easy to implement. The full method of path planning and path tracking has been successfully implemented in the Nomad 200 mobile robot.

This paper is organized as follows. The next section, presents the path tracking algorithm. Section 3 shows its implementation in the Nomad 200 mobile robot and the experimental results obtained. Finally, the last sections are dedicated to conclusions and future work.

* Email: anapozo@dte.uma.es

2 Tracking Strategy

Although the hierarchical path planning obtains an optimal feasible path, the result of following is very high. For this reason, a simpler trajectory than the original is needed in order to achieve a smooth tracking. Thus, the method described in [2] is used to detect the corners of the path calculated by the hierarchical path planning. Using these corners, the algorithm divides the straight line that joins two consecutive corners, obtaining a new reference trajectory as a list of new equispaced computated points (x_i, y_i).

Taking into account this new path, the navigation strategy is based on determining a goal point at a certain distance ahead from the vehicle. While the vehicle moves along trying to reach this, the goal point is always maintained in front of the robot.

The mobile robot changes its curvature by repeatedly fitting circular arcs to the goal point in a pure pursuit strategy [1]. This method is based on geometric considerations and it is very easy to computate. The required curvature of the vehicle (γ_r) is computed by:

$$\gamma_r = -\frac{2\ \Delta x}{D^2} \tag{1}$$

where Δx is the x displacement of the centre of the goal point in vehicle coordinates and D is the distance between the current position of the centre of the robot and the goal point (Fig. 1).

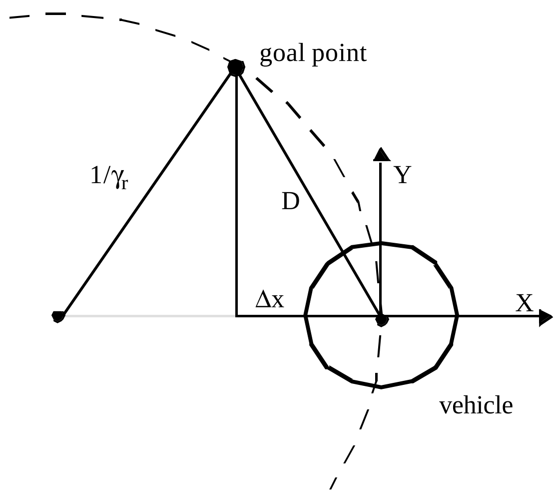


Fig. 1.: Pure pursuit strategy.

When the robot is tracking a previously defined path, the objective is to generate the control commands for the vehicle in order to follow a path in spite of external perturbations. So, it must take into account the present position and the constraints imposed by the vehicle and its lower motion controllers.

The controller approaches the knowledge of the points that form the trayectory, so it is able to make a better tracking of the desired path [9].

Since the vehicle cannot correct the errors with respect to the nearest point in the desired path, the goal point is chosen some distance ahead from the nearest point. The lookahead distance is measured over the desired path in order to make the search easier (Fig. 2).

A bigger lookahead implies smoother control but it also means worse tracking. A smaller lookahead can reduce the tracking errors, however, command control increases and can even become unstable [7]. Explicit path tracking is based on position estimation produced by the odometric system of the mobile robot.

Considering that the points that define the path and the succesive positions of the robot *(x_0, y_0)* are refered with respect to the same coordinates system, the recursive algorithm for path tracking in each period of control consists of the following steps:

1) Obtain the initial position of the robot. This position is provided by the odometric system of the vehicle.

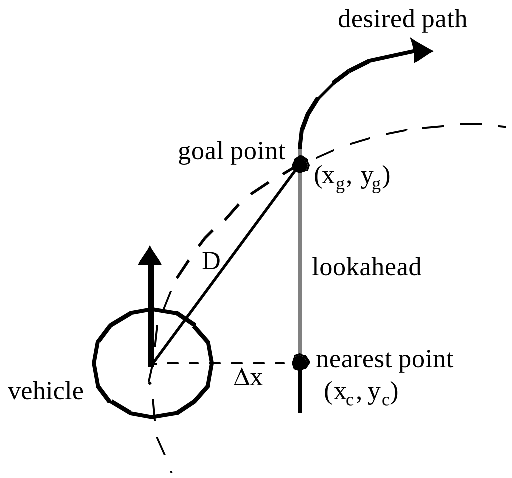


Fig. 2. Lookahead distance.

2) Generate the commands that obtain the tracking of the desired path in a smooth manner:

2.1) Choose a goal point *(x_g, y_g)* at a certain distance ahead from the vehicle:

2.1.1) Calculate the point in the path closer to the vehicle. This nearest point is called *(x_c, y_c)*. The algorithm searches for the nearest point in the proximity of the last nearest point. In this way, the robot is able to follow paths with intersection points.

2.1.2) Compute a certain lookahead distance.

2.1.3) Obtain the goal point *(x_g, y_g)* as the point over the path to follow that is separated from the nearest point *(x_c, y_c)* the lookahead distance.

2.2) Compute the desired curvature of the vehicle γ_r by using the pure pursuit control law defined in Eq. (1).

In this equation, D and Δx are calculated as:

$$D = \sqrt{(x_g - x_0)^2 + (y_g - y_0)^2} \tag{2}$$

$$\Delta x = (x_g - x_0) \ \cos(\phi_0) + (y_g - y_0) \ \sin(\phi_0) \tag{3}$$

where ϕ_0 is the current orientation of the robot, defined as the angle between D and the Y axis in the vehicle coordinates.

3) Move the robot towards the goal point with the desired curvature.

4) Wait a period of control, obtaining the new position of the robot provided by the odometric system and go to point 2 of the algorithm.

The path tracking is aborted when the goal point is placed out of the desired path.

3 Experimental results

The path planning and path tracking algorithms have been tested on the Nomad 200 mobile robot from Nomadic (Fig. 3) [5]. This robot is specially designed for research applications in indoor environments.

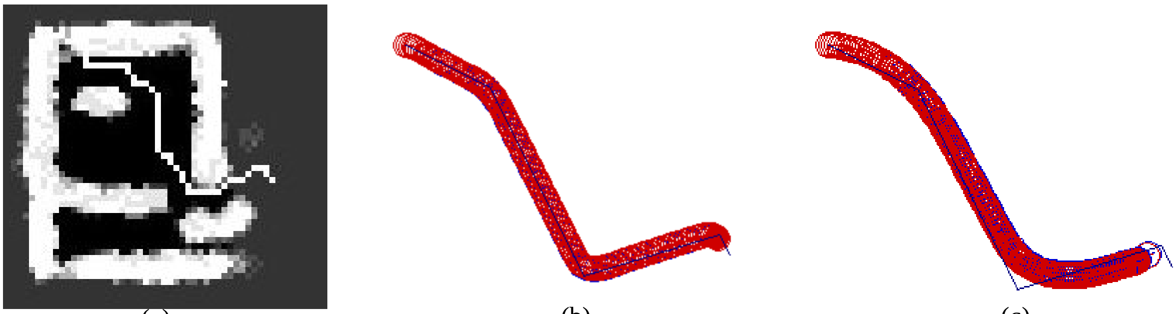
The sensor system of the robot includes tactile (bumper) sensors, infrared sensors and ultrasonic sensors. The Nomad 200 mobile base is a three servo, three-wheeled synchronous drive non-holonomic system with zero gyro-radius. The three wheels translate and rotate together using independent motors for each task. A third motor controls the angular position of the turret.

Fig. 3. The Nomad 200 mobile robot

In order to test the proposed path tracking, a series of experiments has been succesfully performed with the Nomad 200 mobile robot [6].

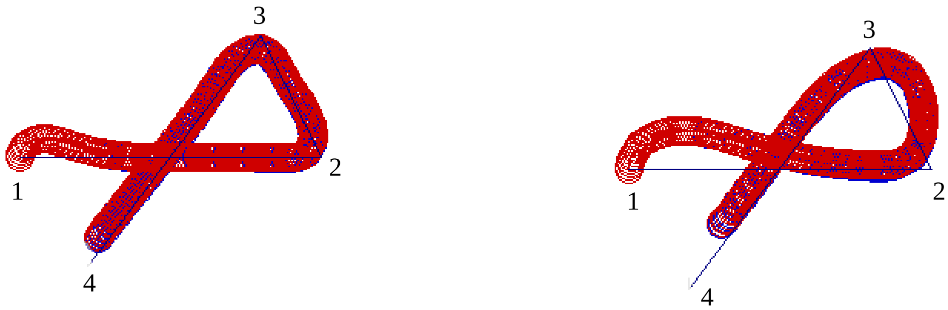
The path obtained by the pyramidal path planning, that connects the departure and arrival points as a list of cells, is shown is Fig. 3a. The main corners of this path have been extracted, and a new reference path composed of straights lines is obtained. Figs. 3b and 3c show in continuous line the new reference trajectory and in outline circles the succesive movements of the robot when tracking the path, using the pure pursuit algorithm, at a constant speed of 0.6 m/s. The value of the lookahead employed in Figs. 3b and 3c is 0.4 metres and 1.2 metres respectively. As can be observed, for a given velocity, if the lookahead is too long, the vehicle cuts corners but small oscillations may result.

Other experiments have been achieved in order to provide the efficiency of the path tracking algorithm proposed. Thus, Fig. 4 shows a path with overlapped points. The point have been numbered sequentially. However, since the algorithm calculates the objetive point in the vecinity of the last objetive point, the robot is able to follow the path without ambiguities. Thus, Fig. 4a presents the tracking with a lookahead value of 0.6 metres, and in Fig. 4b the lookahead is 1.2 metres.



(a) (b) (c)

Fig. 3. a) Path generated by the pyramidal path planning; b) Path tracking with a lookahead of 0.4 metres; c) Path tracking with a lookahead of 1.2 metres.



(a) (b)

Fig. 4. Path with overlapped points. a) Path tracking with a lookahead of 0.6 metres; c) Path tracking with a lookahead of 1.2 metres.

4 Conclusions

This paper presents an algorithm for path following applications represented by grid decomposition in order to provide smooth tracking. The method simplifies the path to follow, calculating its main corners and defines a new trajectory as a list of equispaced points. The path tracking algorithm computes the steering commands in order to follow the path employing the pure pursuit algorithm, where the objetive point is chosen a determinated distance ahead from the robot in the desired path.

This technique is simple and easy to implement. The full method of path planning and path tracking has been successfully implemented in the Nomad 200 mobile robot, presenting excellent results in real time.

Future work includes the development of a navigation system by using this method can avoid any obstacle in its path and continue the tracking once the obstacle has been avoided.

Acknowledgements

This work has been partially supported by the Spanish Comisión Interministerial de Ciencia y Tecnología (CICYT), Project No. TIC098-0562.

References

[1] Amidi, O. (1990), *“Integrated Mobile Robot Control”*, Carnegie Mellon Univ. Robotics Institute, Technical Report CMU-RI-TR-90-17, Pittsburgh, Pennsylvania, (USA).

[2] Arrebola, F., Bandera, A., Camacho, P. and Sandoval, F. (1997), *"Corner detection by local histograms of contour chain code"*, Electronics Letters, Vol. 33, No 21, pp. 1769-1771.

[3] Hong, T., Narayanan, K. A., Peleg, A., Rosenfeld, A. and Silberberg, T. (1982), *“Image Smoothing and Segmentation by Multirresolution Pixel Linking: Further Experiments and Extensions”,* IEEE Trans. on Systems, Man and Cybernetic, Vol. 12, No 5, pp. 611-622.

[4] Moravec, H. P. (1983),*"The Stanford Cart and the CMU Rover"*, Autonomous Robot Vehicles, I. J. Cox and G. T. Wilfong, Ed. Springer-Verlag.

[5] *Nomad 200 Mobile Robot Simulator Language Manual*, Nomad Host Software Development Environment, Release 2.1, Nomadic Technologies, Inc., Mountain View, CA.

[6] *Nomad 200 Mobile Robot User’s Guide*, Nomadic Technologies, Inc., Mountain View, CA.

[7] Ollero, A., García-Cerezo, A. and Martínez, J. L. (1994), *“Fuzzy Supervisory Path Tracking of Mobile Robots”*. Control Engineering Practice, Vol. 2, No. 2, pp. 313-319.

[8] Rombaut, M., Segovia, A., Meizel, D. and Preciado, A. (1991), *“Displacement of a Mobile Robot in a Known Environment”*, IMACS-MTCS Symposium, Villeneuve d’Ascq (France).

[9] Shin, D. H., Singh, S. and Whittaker, W. (1993), *“Path Generation for a Robot Vehicle Using Composite Clothoid Segments”*, SICICA’92 IFAC Symposium, Málaga (Spain), pp. 533-538.

[10] Sordalen, O. J. and Canudas de Wit, C. (1992), *“Exponential Control for Mobile Robot: Extension to Path Following”*, IEEE Int. Conf. on Robotics and Automation, Nice (France), pp. 2158-2163.

[11] Urdiales, C., Bandera, A., Arrebola, F. and Sandoval, F. (1998), *“Multi-Level Path Planning Algorithm for Autonomous Robots”*, Electronics Letters, Vol. 34, No. 2, pp. 223-224.