Point-to-Point Paths Generation for Wheeled Mobile Robots

Diogo P. F. Pedrosa Adelardo A. D. Medeiros Pablo J. Alsina

UFRN – CT – DCA
59072-970, Natal, RN, Brazil
[diogo, adelardo, pablo]@dca.ufrn.br

Abstract

This paper proposes a point-to-point path generation method consistent with the non-holonomic constraints of a two-wheeled mobile robot. The generated path is described by continuous curves, instead of the traditional chaining of different curves. Parametric polynomials of third degree are used to calculate the robot configuration variables, $x(\lambda)$ and $y(\lambda)$. The orientation angle, $\theta(\lambda)$, is imposed to respect the non-holonomic constraint. The free polynomial coefficients are used to refine the path, avoiding maximal or minimal values of $x(\lambda)$ and $y(\lambda)$ which tends to generate shorter paths, avoiding unnecessary motions.

1 Introduction

Path generation is one of the main problems in mobile robot navigation. Frequently, this generation starts by planning what we call a *geometric path* (a path avoiding collisions with obstacles). The main techniques to find these geometric paths were compiled by Latombe [10]. They are based on well-founded algorithms and widely used in robotics systems [2, 3, 8, 12].

When a robot has kinematics constraints, the paths computed by these classic geometric planners are not directly executable by the robot. To solve this problem, it is necessary to adapt the geometric path, finding an *admissible path*. After that, the desired velocities are incorporated into the admissible path to generate the robot trajectory. Finally, the trajectory can be executed by the control level. An overview of a complete navigation robot system based on this approach is shown in fig. 2. This work proposes a technique to be used in the path adaptation module of such a system (level 2 in fig. 2): we do not deal neither with the obstacle avoidance problem (level 1) nor with the execution control of the robot trajectory (level 4).

One well-known method to compute admissible paths was proposed by Laumond *et al* [11]:

- we divide the geometric path in n segments, creating $n+1$ vertices;
- we compute *point-to-point paths* between all pairs of adjacent vertices, without considering collisions; the point-to-point paths must be directly executable by the robot;
- we check if these point-to-point paths introduce collisions with obstacles; in affirmative case, we increase the number n of segments and repeat the process; if not, this set of chained point-to-point paths is the definitive admissible path to the robot.

This approach requires calculating point-to-point paths between two distinct configurations of the robot. Many solutions were proposed [4, 5, 6, 7, 15, 17, 18]: most of them are direct or indirectly influenced by the seminal technique proposed by Reeds and Shepp [16]. They proved that the shortest path between two distinct configurations for a car which can move forward and backward is composed by arcs of circle of minimum radius and line segments. For a two wheeled mobile robot as in fig. 1, however, the minimum turning radius is zero, degenerating the arc of circle motion into a rotation about the robot's geometric center. In this case, the point-to-point paths calculated using the Reeds and Shepp's technique are composed of:

- a rotation about its geometric center to point to the next desired position;
- a straight line motion to this position;
- a rotation to reach the final orientation;

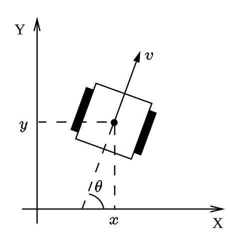

Figure 1: Wheeled mobile robot with null turning radius and configuration variables x, y and θ

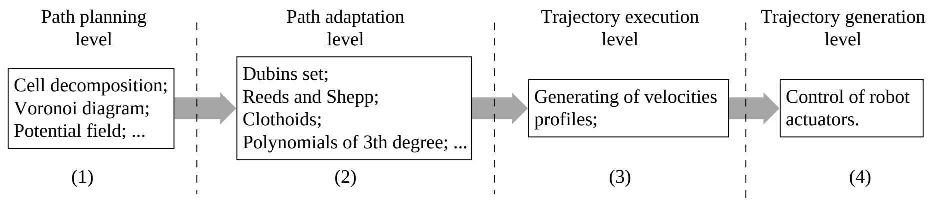

Figure 2: A robot navigation system based on adaptation of geometric paths

In some dynamics situations where the goal can quickly change its position, like soccer robots environments, the Reeds and Shepp's paths are not appropriate because the robot should stop its current line trajectory to adjust itself to a new orientation or can stand rotating for a long time trying to point to the moving goal. To avoid these inconveniences, this work proposes a alternative method to link two distinct configurations in the robot workspace.

The proposed technique is based on using parametric polynomials of third degree to calculate the configuration variables $x(\lambda)$ and $y(\lambda)$. The orientation angle $\theta(\lambda)$ is imposed to coincide with the path tangent because of the non-holonomic constraints. The λ parameter has values between $(0, 1)$, where these extremal values indicate the initial and final configurations, respectively. We choose polynomials of third degree because they have the minimum degree where there are free parameters to refine the paths, as shown in the section 5. An interesting characteristic of the proposed method in real time application is the path determination by a closed formula of fast calculus, permiting the path determination at each sampling step, while Reeds and Shepp's method, for example, has 48 possible paths which must be analyzed in order to find the shortest path.

In section 2, the proposed point-to-point path generator is presented; in section 3, the mathematical singularities are analyzed; in section 4, it is shown how to refine the path in accord to the criterion presented in the section 5; some examples and conclusions are shown in sections 6 and 7, respectively.

2 Paths Generation

Point-to-point paths generation has to consider the non-holonomic constraints. In the case of robots with differential drive[1], the orientation angle must point to its linear velocity vector. We propose generating the configuration variables x and y by parametric polynomials of third degree with unknown coefficients (eqs. 1). Many others works has also used polynomials to generate feasible paths [1, 9, 13].

$$\begin{aligned} x(\lambda) &= a_0 + a_1\lambda + a_2\lambda^2 + a_3\lambda^3 \\ y(\lambda) &= b_0 + b_1\lambda + b_2\lambda^2 + b_3\lambda^3 \end{aligned} \tag{1}$$

The orientation angle θ is imposed to satisfy the non-holonomic constraints (eq. 2):

$$\theta(\lambda) = \tan^{-1}\left(\frac{dy/d\lambda}{dx/d\lambda}\right) = \tan^{-1}[d(\lambda)] \tag{2}$$

where $d(\lambda)$ is introduced to simplify the notation.

$$d(\lambda) = \tan[\theta(\lambda)] = \frac{b_1 + 2b_2\lambda + 3b_3\lambda^2}{a_1 + 2a_2\lambda + 3a_3\lambda^2} \tag{3}$$

In eqs. 1 and 3, λ varies in the interval $(0, 1)$. When $\lambda = 0$, the robot is on its initial configuration $(x_i, y_i, d_i = \tan(\theta_i))$ and similarly when $\lambda = 1$, the robot is on its final configuration $(x_f, y_f, d_f = \tan(\theta_f))$. Applying these contour conditions to eqs. 1 and 3, a linear system with 6 equations and 8 variables is obtained. This system is shown in eq. 4.

$$\begin{cases} a_0 = x_i \\ b_0 = y_i \\ b_1 = d_i a_1 \\ a_0 + a_1 + a_2 + a_3 = x_f \\ b_0 + b_1 + b_2 + b_3 = y_f \\ b_1 + 2b_2 + 3b_3 = d_f(a_1 + 2a_2 + 3a_3) \end{cases} \tag{4}$$

We can arbitrate two of the eight coefficients to solve this linear system. Choosing a_1 and a_2 as the free variables, we can deduce the others coefficients without division operations, avoiding division-by-zero singularities. The solution is given by eq. 5, where $\Delta x = x_f - x_i$ and $\Delta y = y_f - y_i$.

$$\begin{cases} a_0 = x_i \\ a_3 = \Delta x - a_1 - a_2 \\ b_0 = y_i \\ b_1 = d_i a_1 \\ b_2 = 3(\Delta y - d_f\Delta x) + 2(d_f - d_i)a_1 + d_f a_2 \\ b_3 = 3d_f\Delta x - 2\Delta y - (2d_f - d_i)a_1 - d_f a_2 \end{cases} \tag{5}$$

[1]Two parallel and independently driven wheels.

Any values can be attributed to the coefficients a_1 and a_2, resulting in a path that respects the non-holonomic constraints and the contour conditions. However, the generated paths sometimes will not be adequate, as shown in section 4. This leads us to use some criterion to find good values to the free coeficients.

3 Mathematical Singularities

The result in eq. 5 is applicable to most situations, excepting when θ_i and/or θ_f are equals to $\pm\pi/2$, because d_i and d_f tend to $\pm\infty$. Thus, there are three situations where the eq. 3 has to be analyzed in order to redefine the system 4.

The first case occurs when both θ_i and θ_f are equals to $\pm\pi/2$. The new resulting system is given by eqs. 6, with b_1 and b_2 as free variables.

$$\begin{cases} a_0 = x_i \\ a_1 = 0 \\ a_2 = 3\Delta x \\ a_3 = -2\Delta x \\ b_0 = y_i \\ b_3 = \Delta y - b_1 - b_2 \end{cases} \quad (6)$$

The second case occurs if only $\theta_i = \pm\pi/2$. The free coefficients a_3 and b_3 are chosen, resulting in eqs. 7.

$$\begin{cases} a_0 = x_i \\ a_1 = 0 \\ a_2 = \Delta x - a_3 \\ b_0 = y_i \\ b_1 = 2(\Delta y - d_f \Delta x) - d_f a_3 + b_3 \\ b_2 = (2d_f \Delta x - \Delta y) + d_f a_3 - 2b_3 \end{cases} \quad (7)$$

Finally, the third case occurs when only θ_f is equal to $\pm\pi/2$. We can choose a_1 and b_2 as free variables and the result is shown in eqs. 8.

$$\begin{cases} a_0 = x_i \\ a_2 = 3\Delta x - 2a_1 \\ a_3 = a_1 - 2\Delta x \\ b_0 = y_i \\ b_1 = d_1 a_1 \\ b_3 = \Delta y - d_i a_1 - b_2 \end{cases} \quad (8)$$

Similarly to the general case, any values to the free coefficients will result in paths respecting the non-holonomic constraints. However, these paths are not necessarily appropriate to the mobile robot.

4 Non-Adequate Paths

Although the paths generated by eqs. 5, 6, 7 and 8 satisfy the non-holonomic constraints and the contour conditions, sometimes these paths are not appropriate. An example is shown in the fig. 3, where the a_1 and a_2 values ($a_1 = -1.6863$ and $a_2 = 2.4863$) were specially chosen to generate a non-adequate path.

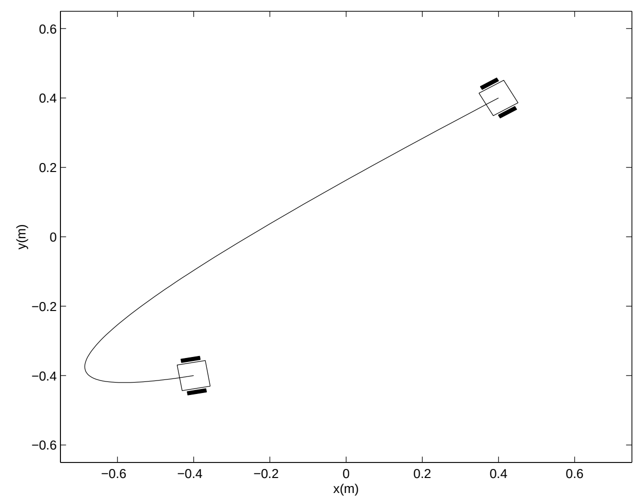

Figure 3: Example of non adequate path

This path is not "intelligent" because it contains a unnecessary backward movement with relation to the final configuration, that could be attained by a completely forward movement. Thus, it is important to use a criterion to calculate the free coefficients in order to improve the path generation.

5 Improvement Criterion

Several improvement criteria can be used to calculate the free coefficients, like minimal length, minimal or maximal rotation radius, etc. The criterion we decide to adopt in this article is to force the $x(\lambda)$ and $y(\lambda)$ polynomials (eqs. 1) to be monotonically increasing or decreasing between $0 < \lambda < 1$. As we do not allow the polynomials having maximum or minimum points in the interval, we eliminate unnecessary backward movements. Another interesting benefit is that the paths are bounded by a rectangle defined by the initial and final positions, if this criterion could be satisfied in both x and y directions. Thus, considering the following derivatives with relation to λ (eqs. 9):

$$\begin{aligned} dx/d\lambda &= a_1 + 2a_2\lambda + 3a_3\lambda^2 \\ dy/d\lambda &= b_1 + 2b_2\lambda + 3b_3\lambda^2 \end{aligned} \quad (9)$$

their roots must satisfy at least one of the following conditions:

- both roots are complex; or
- both roots are smaller or equals to 0; or
- both roots are greater or equals to 1; or
- one root is smaller or equal to 0 and the other is greater or equal to 1.

The first condition implies the discriminant (Δ) must be negative; the other three conditions can be mathematically calculated using the Routh-Hurwitz algorithm [14]. Applying them separately to eqs. 9 we can find two distinct systems of inequations which bound regions in the free coefficients' space which improve the path in the x and y directions. These free coefficients are different for the general and singular cases and a separated analyze will be realized.

5.1 Regions for the General Case

The region $a_1 \times a_2$ satisfying the improvement criterion for $x(\lambda)$ is bounded by the system of ineqs. 10. For $\Delta x > 0$, this region can be visualized in fig. 4.

$$\begin{cases} a_1 \geq 0 \\ a_2 \geq -a_1 \\ a_2 \leq 3\Delta x - 2a_1 \end{cases} \qquad (10)$$

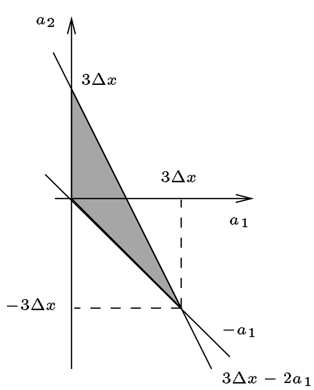

Figure 4: Region satisfying the x constraints

A similar calculus can be realized for the y direction, obtaining the region bounded by the system of ineqs. 11. Considering Δx, Δy, d_i and d_f positives this region can be shown in fig. 5.

$$\begin{cases} d_i a_1 \geq 0 \\ d_f a_2 \geq (d_i - 2d_f)a_1 + 3(d_f\Delta x - \Delta y) \\ d_f a_2 \leq 3d_f\Delta x - 2d_f a_1 \end{cases} \qquad (11)$$

To improve the path in both directions, it is necessary to find a region derived from the intersection

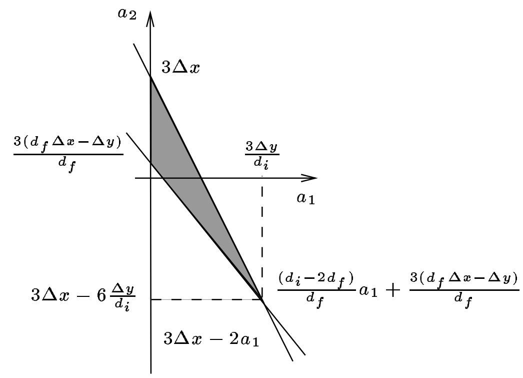

Figure 5: Region satisfying the y-constraints

between the regions for the x and y directions. In the examples shown in figs. 4 and 5 this intersection exists, but this is not always the case. Analyzing the systems 10 and 11 we can conclude that an intersection only exists when the initial and final orientations are pointing to inside of a rectangle defined by the initial and final positions. If this does not occur, the path can only be improved in one of the two directions. Figure 6 illustrates these two situations.

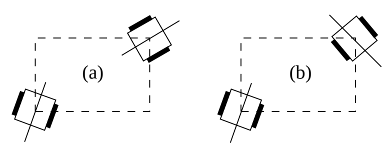

Figure 6: Situations where (a) it is and (b) it is not possible improving the path in both directions

5.2 Regions for the Singular Cases

The same improvement criterion imposed to the general case is applied to the cases where there are singularities, shown in section 3.

Considering the case where θ_i and θ_f are equals to $\pm\pi/2$ the region that improves the path in the y direction is defined by ineqs. 12. The criterion is always satisfied in the x direction.

$$\begin{cases} b_1 \geq 0 \\ b_2 \geq -b_1 \\ b_2 \leq -2b_1 + 3\Delta y \end{cases} \qquad (12)$$

When only $\theta_i = \pm\pi/2$ the region that satisfies the criterion in the x direction is:

$$-2\Delta x \leq a_3 \leq \Delta x$$

while for the direction y is:

$$\begin{cases} a_3 \geq -2\Delta x \\ b_3 \leq \Delta y \\ b_3 \geq d_f a_3 - 2(\Delta y - d_f \Delta x) \end{cases}$$

Finally, when only $\theta_f = \pm\pi/2$ the result in the x direction is:

$$0 \leq a_1 \leq 3\Delta x$$

and to the direction y the region is:

$$\begin{cases} a_1 \geq 0 \\ b_2 \geq -d_i a_1 \\ b_2 \leq -2d_i a_1 + 3\Delta y \end{cases}$$

6 Examples

Figure 7 shows a path where the improvement criterion could be applied in both directions. The initial and final configurations are the same used in fig. 3. As we can see, this result exhibits a better performance.

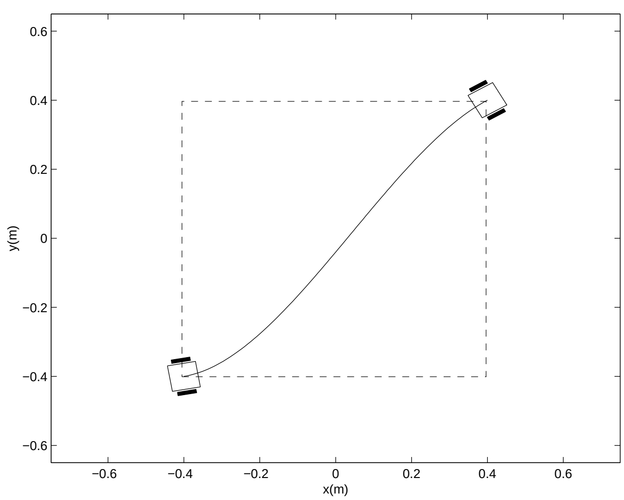

Figure 7: Improved path satisfying the criterion in both x and y directions

Figure 8 exemplifies a situation where it is not possible attending the improvement criterion in both x and y directions simultaneously. Thus, we show two paths where the first one is improved in the x direction and the second one, in the y direction.

7 Conclusions

The main contribution of this work is a point-to-point path generator which obeys the non-holonomic constraints for a wheeled mobile robot with differential

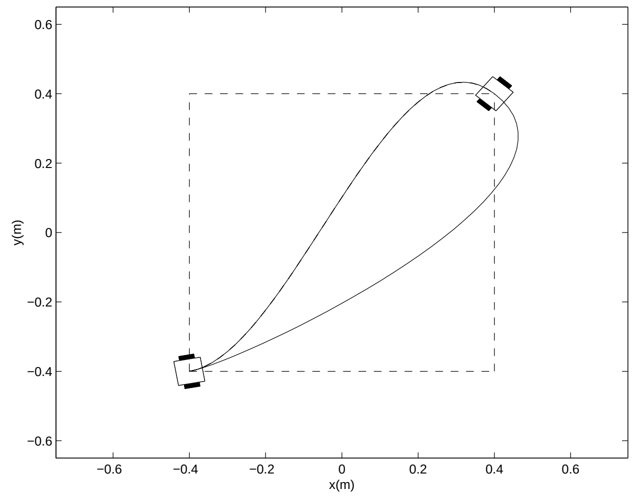

Figure 8: Improved paths satisfying the criterion in the x and y directions, separately

drive. The paths are described by continuous polynomials functions and not by a chaining of different segments. They are easily calculated, allowing a new path to be found at each sampling step. As there are no phases in the path execution, a new path can be easily adopted during the execution of a previous one.

The proposed method can be used to convert geometric paths in feasible paths in such a way they can be executed by non-holonomic robots. In environments without or with sparse obstacles (as a robot soccer environment), the geometric planner can be suppressed and the proposed method can be used as the unique path planner, combined with local avoiding collision techniques.

The adopted improvement criterion guarantees a monotonic reduction of the distance from the current robot position to the goal position, in the better situation. In the worst case the criterion guarantees the improvement in one of the x or y directions.

Possible extensions of this work include velocity profile generators for the point-to-point paths, comparative tests with others adapters and defining some other improvement criteria. Higher order polynomials can also be used to guarantee multi-criteria improvement.

8 Acknowledgments

This work was partially supported by the Brazilian agencies FINEP and CNPq.

References

[1] Akira Arakawa, Masayuki Hiyama, Takashi Emura, and Yoshiyuki Kagami. Trajectory generation for wheeled mobile robot based on landmarks. In *Proceedings of the 1995 IEEE International Conference on Systems, Man and Cybernetics*, Vancouver, Canada, October 1995.

[2] Howie Choset and Joel Burdick. Sensor based planning, part 01: The generalized voronoi graph. In *Proceedings of the IEEE International Conference of Robotics and Automation*, Nagoya, Japan, 1995.

[3] Howie Choset and Joel Burdick. Sensor based planning, part 02: Incremental construction of the generalized voronoi graph. In *Proceedings of the IEEE International Conference of Robotics and Automation*, Nagoya, Japan, 1995.

[4] Guy Desaulniers and François Soumis. An efficient algorithm to find a shortest path for a car-like robot. *IEEE Transactions on Robotics and Automation*, 11(6), December 1995.

[5] Adam W. Divelbiss and John T. Wen. A path space approach to nonholonomic motion planning in the presence of obstacles. *IEEE Transactions on Robotics and Automation*, 13(3), June 1997.

[6] Sara Fleury, Philippe Souères, Jean-Paul Laumond, and Raja Chatila. Primitives for smoothing mobile robot trajectories. *IEEE Transactions on Robotics and Automation*, 11(3), June 1995.

[7] Atsushi Fujimori, Peter N. Nikiforuk, and Madan M. Gupta. Adaptive navigation of mobile robots with obstacle avoidance. *IEEE Transactions on Robotics and Automation*, 13(4), August 1997.

[8] Yong K. Hwang and Narendra Ahuja. A potential field approach to path planning. *IEEE Transactions on Robotics and Automation*, 8(1), February 1992.

[9] Maher Khatib, Hazem Jaouni, Raja Chatila, and Jean-Paul Laumond. Dynamic path modification for car-like nonholonomic mobile robots. In *Proceedings of 1997 IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico, USA, April 1997.

[10] Jean-Claude Latombe. *Robot Motion Planning.* Kluwer Academic Press, 1991.

[11] Jean-Paul Laumond, Paul E. Jacobs, Michel Taïx, and Richard M. Murray. A motion planner for nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 10(5), October 1994.

[12] J. L. Díaz de Léon and J. H. Sossa. Automatic path planning for a mobile robot among obstacles of arbitrary shape. *IEEE Transactions on Systems, Man, and Cybernetics*, 28(3), June 1998. Part B: Cybernetics.

[13] Mark B. Milam, Kudah Mushambi, and Richard M. Murray. A new computational approach to real-time trajectory generation for constrained mechanical systems. In *Proceedings of the 39th IEEE Conference on Decision and Control*, Sydney, Australia, December 2000.

[14] Katsuhiko Ogata. *Modern Control Engeneering.* Pretence Hall, 1997. Third Edition.

[15] Leszek Podsędkowski, Jacek Nowakowski, Marec Idzikowski, and Istvan Vizvary. A new solution for path planning in partially known or unknown enviroment for nonholonomic mobile robots. *Robotics and Autonomous Systems*, 34, 2001.

[16] J. A. Reeds and L. A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, 145(2), 1990.

[17] A. Scheuer and Th. Fraichard. Continuous-curvature path planning for car-like vehicles. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Grenoble, France, September 1997.

[18] Andrei M. Shkel and Vladimir Lumelsky. Classification of the dubins set. *Robotics and Autonomous Systems*, 34, 2001.