ABCM Symposium Series in Mechatronics - Vol. 3 - pp.300-307

PARAMETRIC TRAJECTORY GENERATION FOR MOBILE ROBOTS

Trajano Alencar de Araujo Costa, trajano@ctex.eb.br
Centro Tecnolgico do Exrcito - CTEx - Av das Amricas, 28705 - Guaratiba - Rio de Janeiro, RJ, Brazil.
Armando Morado Ferreira, armando@ime.eb.br
Instituto Militar de Engenharia - IME - Praa General Tibrcio, 80 - Urca - Rio de Janeiro, RJ, Brazil.
Max Suell Dutra, max@mecanica.coppe.ufrj.br
Robotics and Automation Lab - COPPE/UFRJ, Federal University of Rio de Janeiro - Rio de Janeiro, RJ, Brazil.

Abstract. *The problem associated with trajectory generation for a mobile robot moving from an initial configuration to any final configuration is studied in this paper. The aim to find a trajectory with minimum energy cost of execution, by using primitive five order polynomials to generate both paths and trajectories for coordinates x and y. The method allows for the imposition of initial and final position, orientation and velocity constraints, and the solution complies with kinematic and dynamic constraints, particularly the typical non-holonomic constraint of differential drive robots.*

Keywords: *path planning, trajectory generation, mobile robots.*

1. INTRODUCTION

Automatic trajectory generation is one of the most important functions which an autonomous mobile robot has to perform, since such a robot must be able to move in his workspace whilst avoiding obstacles and reaching specific goal configurations, as well as saving as much energy as possible.

Usually, a trajectory is built from a geometric path that takes the robot from the initial configuration to its goal. However, a typical path does not necessarily comply with the robot's particular kinematics and dynamics, such as the non-holonomic constraint of a wheeled mobile robot. Therefore, if the actual motion is admissible or even possible at all, a greater control effort may be required.

Hence, trajectory generation methods shall integrate geometric path generation, kinematic and dynamic constraints, and control design. However, most of the trajectory generation methods found in the literature, as in Latombe (1991) and Choset (2005), do not properly integrate these factors, choosing instead to assume simpler models such as "free flying robots" or to decouple trajectory and path generation.

Among the exceptions, we find those who do not over-simplify the problem, such as Pedrosa, Medeiros, and Alsina (2002), who propose a path generation method using third degree polynomials to comply with the non-holonomic constraint of a two wheels mobile robot. However, they fail to comply with velocity and dynamic constraints, and do not obtain optimal solutions.

In this work, we extend their method to achieve compliance with velocity and dynamic constraints. In addition, we make it possible to find an optimal trajectory regarding to a minimum control input criteria.

We do such by using fifth degree polynomials for a parameterized path function $X\left(\lambda\right)$, which is constructed to comply with the non-holonomic constraint.

After a proper choice of a velocity profile $\lambda\left(t\right)$ and the imposition all of the initial and final conditions, from such path function, we derive a trajectory function, which still has free coefficients. These are found by minimization of the control input, while the dynamic compliance is guaranteed by computing the corresponding control torques by dynamic inversion.

Such optimization may also take into account a virtual potential field associated with obstacles in the robot's workspace, what further extends the method.

The article is organized as follows: in section 2 we show the parameterized polynomial formulation for the path generation, in section 3 we present the velocity profile which accounts for the dynamics constraints within the path, in section 4 we discuss an application of the proposed method, and in section 5 we show experimental results, followed by the conclusions.

2. PARAMETRIC POLYNOMIALS

This fifth degree polynomial method, as mentioned in the introduction, was inspired by the third degree polynomial method of Pedrosa et al. (2002). When applied to a two wheel, differential drive, mobile robot, such greater polynomial degree allows for attending initial and final velocity constraints, in addition to the non-holonomic constraint. If needed or convenient, such properties also allow for the generation of smooth and fully kinematically compliant path segments, hence composing a continuous path.

In this method, two polynomials are used, one for each of the cartesian planar coordinates, x and y:

$$x(\lambda) = a_0 + a_1\lambda + a_2\lambda^2 + a_3\lambda^3 + a_4\lambda^4 + a_5\lambda^5; \tag{1}$$

$$y(\lambda) = b_0 + b_1\lambda + b_2\lambda^2 + b_3\lambda^3 + b_4\lambda^4 + b_5\lambda^5; \tag{2}$$

where $a_0 \ldots a_5$ are the coefficients of the polynomial which corresponds to the path in x, $b_0 \ldots b_5$ are the coefficients of the polynomial which corresponds o the path in y, and λ is the parameter which defines the trajectory evolution within the path from the initial point to the final point, according to a velocity profile $\lambda(t)$. The parameter λ takes values from zero to one, with $\lambda = 0$ at the initial point and $\lambda = 1$ at the final point.

Considering both x and y polynomials, we have a total of twelve coefficients, of which ten are determined by imposing initial and final conditions for the states:

$$\begin{array}{l} (x_i, y_i, \theta_i) = configuration\ initial \\ (x_f, y_f, \theta_f) = configuration\ final \end{array},$$

and corresponding velocities:

$$\begin{array}{l} (v_i, \omega_i) = velocity\ initial \\ (v_f, \omega_f) = velocity\ final \end{array}.$$

The remaining two coefficients, one for each polynomial, are kept free, in order to be used for dynamic compliance and trajectory optimization, as illustrated in the Section 4.

Differentiating the expressions in Eq. (1) and (2) with respect to λ, it results in

$$\frac{d}{d\lambda}\left[x(\lambda)\right] = v_x(\lambda) = a_1 + 2a_2\lambda + 3a_3\lambda^2 + 4a_4\lambda^3 + 5a_5\lambda^4, \tag{3}$$

and

$$\frac{d}{d\lambda}\left[y(\lambda)\right] = v_y(\lambda) = b_1 + 2b_2\lambda + 3b_3\lambda^2 + 4b_4\lambda^3 + 5b_5\lambda^4, \tag{4}$$

where $v_x(\lambda)$ e $v_y(\lambda)$ are the velocity components with respect to λ.

Now, differentiating the expressions in Eq. (3) and (4) with respect to λ, we have

$$\frac{d}{d\lambda}\left[v_x(\lambda)\right] = a_x(\lambda) = 2a_2 + 6a_3\lambda + 12a_4\lambda^2 + 20a_5\lambda^3, \tag{5}$$

and

$$\frac{d}{d\lambda}\left[v_y(\lambda)\right] = a_y(\lambda) = 2b_2 + 6b_3\lambda + 12b_4\lambda^2 + 20b_5\lambda^3, \tag{6}$$

where $a_x(\lambda)$ e $a_y(\lambda)$ are the acceleration components with respect to λ.

The magnitude of the total velocity with respect to λ is given by

$$v(\lambda) = \sqrt{v_x(\lambda) + v_y(\lambda)}. \tag{7}$$

Notice that even tough the total velocity with respect to λ implies an orientation for the velocity vector, the actual magnitude of the velocity with respect to time might be different, depending on the chosen velocity profile, as shown in section 3.

The robot's orientation angle is given by

$$\theta = \arctan\left(\frac{\frac{d}{d\lambda}\left[y(\lambda)\right]}{\frac{d}{d\lambda}\left[x(\lambda)\right]}\right) = \arctan\left(\frac{v_y(\lambda)}{v_x(\lambda)}\right), \tag{8}$$

and in this way the non-holonomic constraint is satisfied, as the orientation corresponds to the path's tangent angle.

Hence, by straightforward differentiation of Eq. (8), we have the angular velocity with respect to λ

$$\omega(\lambda) = \frac{a_y(\lambda)\, v_x(\lambda) - a_x(\lambda)\, v_y(\lambda)}{v_x^2(\lambda) + v_y^2(\lambda)}. \tag{9}$$

We find the polynomial coefficients by imposing the initial and final conditions:

$$x(\lambda = 0) = a_0 = x_i, \tag{10}$$

$$y\left(\lambda=0\right)=b_0=y_i, \tag{11}$$

$$x\left(\lambda=1\right)=a_0+a_1+a_2+a_3+a_4+a_5=x_f, \tag{12}$$

$$y\left(\lambda=1\right)=b_0+b_1+b_2+b_3+b_4+b_5=y_f, \tag{13}$$

where (x_i, y_i) is the initial position corresponding to $\lambda = 0$, and (x_f, y_f) is the final position corresponding to $\lambda = 1$.

We also apply the initial and final velocity conditions and impose the initial and final orientations:

$$\frac{d}{d\lambda}\left[x\left(\lambda=0\right)\right]=a_1=v_i^p\cos\theta_i, \tag{14}$$

$$\frac{d}{d\lambda}\left[y\left(\lambda=0\right)\right]=b_1=v_i^p\sin\theta_i, \tag{15}$$

$$\frac{d}{d\lambda}\left[x\left(\lambda=1\right)\right]=a_1+2a_2+3a_3+4a_4+5a_5=v_f^p\cos\theta_f, \tag{16}$$

$$\frac{d}{d\lambda}\left[y\left(\lambda=1\right)\right]=b_1+2b_2+3b_3+4b_4+5b_5=v_f^p\sin\theta_f, \tag{17}$$

where v_i^p and v_f^p denote velocities with respect to λ.

Setting a_2, a_3, b_2, and b_3 as free coefficients and solving Eq. (10) to (17) we have:

$$\left[\begin{array}{l} a_0=x_i \\ a_1=v_i^p\cos\theta_i \\ a_2=Free\ coefficient \\ a_3=Free\ coefficient \\ a_4=5\left(x_f-x_i\right)-3a_2-2a_3-4v_i^p\cos\theta_i-v_f^p\cos\theta_f \\ a_5=4\left(x_i-x_f\right)+2a_2+a_3+3v_i^p\cos\theta_i+v_f^p\cos\theta_f \end{array}\right], \tag{18}$$

and

$$\left[\begin{array}{l} b_0=y_i \\ b_1=v_i^p\sin\theta_i \\ b_2=Free\ coefficient \\ b_3=Free\ coefficient \\ b_4=5\left(y_f-y_i\right)-3b_2-2b_3-4v_i^p\sin\theta_i-v_f^p\sin\theta_f \\ b_5=4\left(y_i-y_f\right)+2b_2+b_3+3v_i^p\sin\theta_i+v_f^p\sin\theta_f \end{array}\right]. \tag{19}$$

The next step is to apply the conditions corresponding to initial and final angular velocities with respect to λ:

$$\varpi\left(\lambda=0\right)=\omega_i^p, \tag{20}$$

$$\varpi\left(\lambda=1\right)=\omega_f^p. \tag{21}$$

Therefore, only two coefficients are to remain free. Also, we notice that some choices of this pair of free coefficients may be associated with a division by zero, what we should avoid it implies in great numerical imprecision (Burden and Faires, 2003). Nonetheless, it is always possible to pick a proper pair of free coefficients as shown next.

Table (1) shows four possible pairs of free coefficients.

	Free Coefficients
Case 1	a_2, b_3
Case 2	b_2, b_3
Case 3	a_2, a_3
Case 4	a_3, b_2

Table 1. Possible pairs of the free coefficients.

For each of the four possible pairs, we find the respective expressions for the corresponding dependent coefficients:

CASE 1

$$a_3 = \frac{\left(10\left(y_i - y_f\right) + 3b_2 + b_3 + 6v_i^p \sin\theta_i\right)\cos\theta_f}{\sin\theta_f} - 3a_2 + 10\left(x_f - x_i\right) - 6v_i^p \cos\theta_i - \frac{v_f^p \omega_f^p}{2\sin\theta_f} \tag{22}$$

$$b_2 = \frac{2a_2 \sin\theta_i + v_i^p \omega_i^p}{2\cos\theta_i} \tag{23}$$

CASE 2

$$a_2 = -\frac{2b_2 \cos\theta_i + v_i^p \omega_i^p}{2\sin\theta_i} \tag{24}$$

$$a_3 = \frac{\left(10\left(y_i - y_f\right) + 3b_2 + b_3 + 6v_i^p \sin\theta_i\right)\cos\theta_f}{\sin\theta_f} - 3a_2 + 10\left(x_f - x_i\right) - 6v_i^p \cos\theta_i - \frac{v_f^p \omega_f^p}{2\sin\theta_f} \tag{25}$$

CASE 3

$$b_2 = \frac{2a_2 \sin\theta_i + v_i^p \omega_i^p}{2\cos\theta_i} \tag{26}$$

$$b_3 = \frac{\left(10\left(x_i - x_f\right) + 3a_2 + a_3 + 6v_i^p \sin\theta_i\right)\sin\theta_f}{\cos\theta_f} - 3b_2 + 10\left(y_f - y_i\right) - 6v_i^p \sin\theta_i - \frac{v_f^p \omega_f^p}{2\cos\theta_f} \tag{27}$$

CASE 4

$$a_2 = \frac{2b_2 \cos\theta_i + v_i^p \omega_i^p}{2\sin\theta_i} \tag{28}$$

$$b_3 = \frac{\left(10\left(x_i - x_f\right) + 3a_2 + a_3 + 6v_i^p \sin\theta_i\right)\sin\theta_f}{\cos\theta_f} - 3b_2 + 10\left(y_f - y_i\right) - 6v_i^p \sin\theta_i - \frac{v_f^p \omega_f^p}{2\cos\theta_f} \tag{29}$$

From Eq. (22) through (29), we see that some initial or final orientations could lead to division by zero for a certain case. However, it is still possible to pick another pair which avoids numerical problems. From Table (2), which summarizes such singularities, we notice that for an initial angle in the range $\left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$ or $\left[\frac{3\pi}{4}, \frac{5\pi}{4}\right]$, cases 1 and 3 are to be used. Otherwise, cases 2 and 4 are to be used. Regarding to the final orientation, in the range $\left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$ or $\left[\frac{3\pi}{4}, \frac{5\pi}{4}\right]$, cases 1 and 2 are to be used, while cases 3 and 4 apply outside these ranges.

Case 1	$\theta_i = \frac{\pi}{2}$	$\theta_i = \frac{3\pi}{2}$	$\theta_f = 0$	$\theta_f = \pi$
Case 2	$\theta_i = 0$	$\theta_i = \pi$	$\theta_f = 0$	$\theta_f = \pi$
Case 3	$\theta_i = \frac{\pi}{2}$	$\theta_i = \frac{3\pi}{2}$	$\theta_f = \frac{\pi}{2}$	$\theta_f = \frac{3\pi}{2}$
Case 4	$\theta_i = 0$	$\theta_i = \pi$	$\theta_f = \frac{\pi}{2}$	$\theta_f = \frac{3\pi}{2}$

Table 2. Singularities for initial and final orientation.

3. VELOCITY PROFILE

In order to relate the parameterized path to a trajectory in time, we need to establish a relation $\lambda = \lambda(t)$, subject to the constraints of Eq. (30) and (31).

$$\lambda\left(t = 0\right) = 0, \tag{30}$$

and

$$\lambda\left(t = t_f\right) = 1. \tag{31}$$

Once we have $\lambda(t)$, the Eq. (1) and (2) transform to:

$$x\left(t\right) = a_0 + a_1\lambda\left(t\right) + a_2\lambda^2\left(t\right) + a_3\lambda^3\left(t\right) + a_4\lambda^4\left(t\right) + a_5\lambda^5\left(t\right), \tag{32}$$

$$y\left(t\right) = b_0 + b_1\lambda\left(t\right) + b_2\lambda^2\left(t\right) + b_3\lambda^3\left(t\right) + b_4\lambda^4\left(t\right) + b_5\lambda^5\left(t\right), \tag{33}$$

which differentiated with respect to time result in

$$\frac{d}{dt}\left[x\left(t\right)\right]=v_x\left(t\right)=v_x\left(\lambda\right)\left(\frac{d}{dt}\lambda\left(t\right)\right) \tag{34}$$

$$\frac{d}{dt}\left[y\left(t\right)\right]=v_y\left(t\right)=v_y\left(\lambda\right)\left(\frac{d}{dt}\lambda\left(t\right)\right) \tag{35}$$

The Eq. (7) and (9) are also rewritten with explicit time as Eq. (36) and (37):

$$v\left(t\right)=\sqrt{\left(v_x^2\left(\lambda\right)+v_y^2\left(\lambda\right)\right)\left(\frac{d}{dt}\lambda\left(t\right)\right)}=v\left(\lambda\right)\left(\frac{d}{dt}\lambda\left(t\right)\right) \tag{36}$$

and

$$\omega\left(t\right)=\omega\left(\lambda\right)\left(\frac{d}{dt}\lambda\left(t\right)\right). \tag{37}$$

A proper choice of function $\lambda(t)$ allows for null velocities at the initial and final times, just by setting to zero the corresponding velocities with respect to λ.

Two additional conditions might be considered when constructing $\lambda(t)$, even tough they are not mandatory. The first one consists on forbidding the robot to trace back the path. Such condition takes the form:

$$\lambda\left(t\right)<\lambda\left(t+\Delta t\right). \tag{38}$$

The second condition assures that the robot moves only in between the initial and the final positions, and it has the form:

$$0\leq\lambda\left(t\right)\leq 1\Rightarrow t=\left[0,t_f\right] \tag{39}$$

In order to respect the four conditions shown, we take some primitive function $\eta\left(\varepsilon\right)$, which is strictly positive in the range $[0,t_f]$, and apply the following transformation:

$$\rho\left(u\right)=\frac{\int\limits_0^{\frac{u}{t_f}}\eta\left(\varepsilon\right)d\varepsilon}{\int\limits_0^1\eta\left(\varepsilon\right)d\varepsilon} \tag{40}$$

Looking at Eq. (40), we notice that: the constraint of Eq. (30) is satisfied if $\rho\left(u=0\right)=0$, the constraint of Eq. (31) is satisfied if $\rho\left(u=t_f\right)=1$, and the two additional conditions are also satisfied if $\eta\left(\varepsilon\right)$ is strictly positive over the domain of interest.

4. EXAMPLE

As an example, we consider the following initial and final conditions:

$$\left(x_i,y_i,\theta_i,v_i,\omega_i\right)=\left(2m,1m,0,0ms^{-1},0s^{-1}\right) \tag{41}$$

and

$$\left(x_f,y_f,\theta_f,v_f,\omega_f\right)=\left(10m,7m,-\frac{\pi}{4},1ms^{-1},0s^{-1}\right). \tag{42}$$

The velocities with respect to λ, both initial and final, correspond to $1ms^{-1}$. From Eq. (18) and (19) we obtain:

$a_0=2$	$a_1=1$	$a_4=35.293-3a_2-2a_3$	$a_5=-28.293+2a_2+a_3$
$b_0=1$	$b_1=0$	$b_4=30.707-3b_2-2b_3$	$b_5=-24.707+2b_2+b_3$

Table 3. Coefficients determined by Eq. (18) and (19).

The case 1 is to be used, and the coefficients a_3 e b_2 are determined by Eqs. (22) and (23), as functions of the free coefficients a_2 e b_3. Therefore,

$$a_3=134-b_3-3a_2$$
$$b_2=0$$

We choose to parameterize the coordinates as follows:

$$x(\lambda) = 2 + \lambda + a_2\lambda^2 + (134 - b_3 - 3a_2)\lambda^3 + (-232.707 + 3a_2 + 2b_3)\lambda^4 + (105.707 - a_2 - b_3)\lambda^5 \tag{43}$$

and

$$y(\lambda) = 1 + b_3\lambda^3 + (30.707 - 2b_3)\lambda^4 + (-24.707 + b_3)\lambda^5. \tag{44}$$

The two free coefficients are used to optimize the objective function:

$$\min(J) = \min\left[\int_0^{t_f} \left(\tau_r^2(t) + \tau_l^2(t)\right) dt\right], \tag{45}$$

where τ_r and τ_l are the right and left torque in the wheels. The wheels' masses are neglected.

Such optimization problem is numerically solved by the Downhill Simplex Method (Nelder and Mead, 1965), implemented as described by "Numerical Recipes in C" (Press et all, 2002), finding $a_2 = 30$ e $b_3 = 80$, from what results the path shown in Fig. (1).

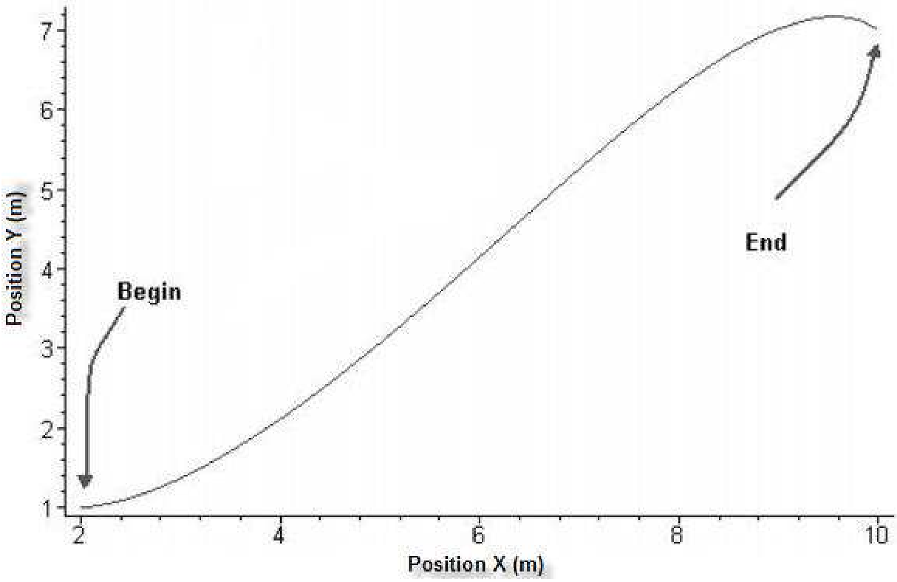

Figure 1. Optimal path

For the velocity profile we choose to use the function $\eta(\varepsilon) = c_0 + c_1\varepsilon + c_2\varepsilon^2$, which is plugged in Eq. (40) and subject to the conditions:

$$\frac{d}{du}(\rho(u=0)) = v_i = 0 \tag{46}$$

and

$$\frac{d}{du}(\rho(u=t_f)) = v_f = 1 \tag{47}$$

where $t_f = 10s$.

5. EXPERIMENT

In order to investigate the viability of the method for practical applications, we set up an experiment consisting of a robot inside a walled rectangular area filled with three obstacles: a box and two other stationary similar robots, as partially shown in Fig. (2), where the moving robot sits in the upper right.

For such set-up, an optimal trajectory was precomputed using the method described, with its cost function augmented by a virtual potential field,in order to account for the obstacles.

Additionally, a controller was designed for the practical implementation of the motion. Such controller is based on a Model-Predictive Control scheme, where the controller takes advantage of the same cost function as the trajectory planner (see [3] for details).

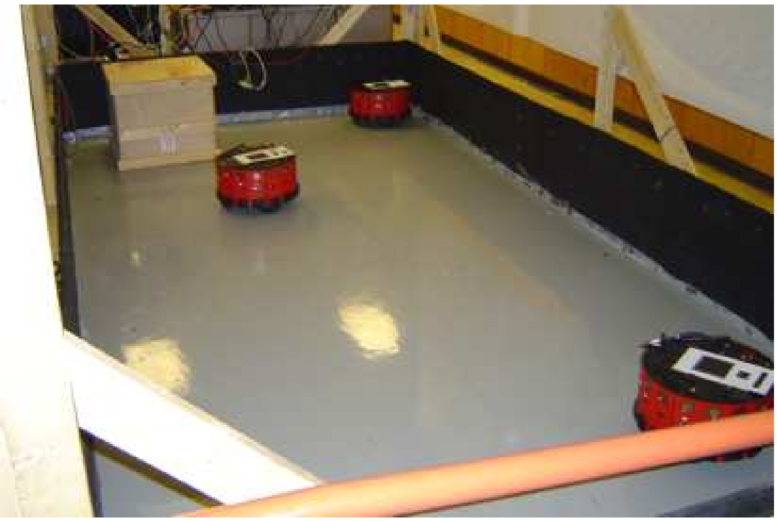

Figure 2. Experimental setup.

The initial and final conditions are as follows:

$$\begin{aligned}&(x_i, y_i, \theta_i, v_i, \omega_i) = \left(0.3m, 0.3m, 0, 0ms^{-1}, 0s^{-1}\right)\\&(x_f, y_f, \theta_f, v_f, \omega_f) = \left(1.5m, 3.5m, \pi, 0ms^{-1}, 0s^{-1}\right).\end{aligned} \tag{48}$$

We used three polynomial segments and determined the free coefficients to minimize the actuators inputs. A Fig. (3) depicts a simulation of the experimental setup. The goal is to reach point 1 heading to point 2. The elements 5, 6 and 7 are the obstacles, as well as the lines 2, 3, 4 and 8.

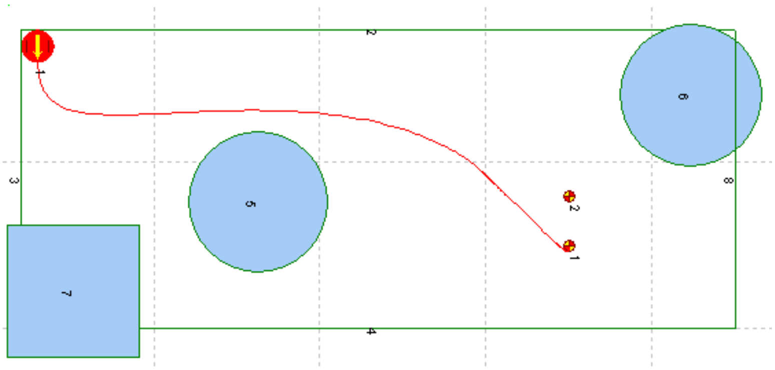

Figure 3. Simulation of the experimental setup.

The results are shown in Fig. (4), where the actual path is depicted in a dashed line and the planned path in a continuous line, demonstrating a good agreement.

6. CONCLUSION

The main contribution of this article is the presentation of a method of generation of trajectories, using parametric polynomials of fifth degree, respecting the non-holonomic constrains of the robot. The gotten polynomials possess two free coefficients, one in each polynomial, which are used to obtain a trajectory with an associated minimum energy cost.

The method allows to specify the initial and final pose of the robot, as well as the longitudinal and angular velocities, both in the initial and in the final point. It guarantees a total compliance with the initial and final states of the robot, as well as with the non-holonomic and dynamic constraints, thus allowing the accomplishment of trajectories of great complexity.

Extensions of this work may take in to account additional costs, such as those relative to the avoidance of obstacles, as exemplified in the experimental results shown, which also confirm the applicability of the proposed method.

7. ACKNOWLEDGMENTS

The authors gratefully thank the support of CNPq and Brazilian Army and appreciate the help on the conducting experiments by Dr. Sunil Kumar Agrawal and Dr. Yong Shing Hao, both from the Mechanical Systems Lab, University

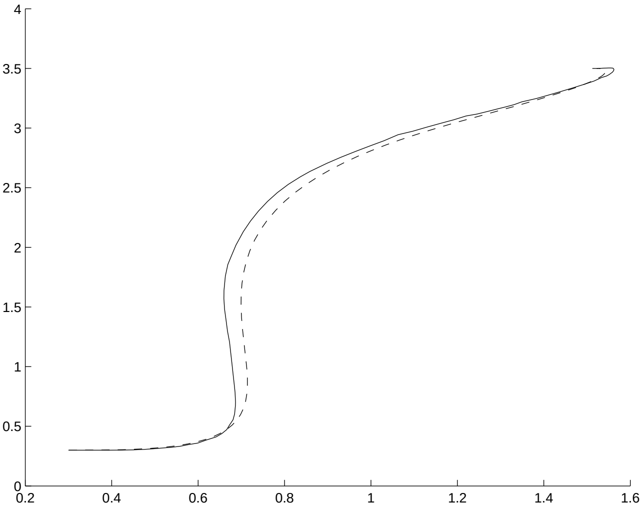

Figure 4. Robot path.

of Delaware.

8. REFERENCES

Burden, R.L. and Faires, D., 2003, "Anlise Numrica", Ed. Thomson, 736p.

Choset, H., 2005, "Principles of Robot Motion: Theory, Algorithms, and Implementations", MIT Press.

Costa, T. A. de A., 2004, "Controle Integrado de um Veculo No-Tripulado",Dissertao de Mestrado, IME, 2004.

Gu, D. and Hu, H., 2002, "Neural Predictive Control for a Car-like Mobile Robot", Robotics and Autonomous Systems.

Latombe, J.C., 1991, "Robot Motion Planning", Kluwer Academic Publishers, 651p.

Nagy, B. and Kelly, A., 2001, "Trajectory Generation for Car-Like Robots Using Cubic Curvature Polynomials", Field and Service Robots, Finland.

Nelder, J. and Mead, R., 1965, "Downhill Simplex Method", Computer Journal.

Pedrosa, D.P.F, Medeiros, A.A.D. and Alsina, P.J., 2002, "Gerao de Caminhos Ponto-a-Ponto Para Robs Mveis com Rodas", Congresso Brasileiro de Automtica, Natal-RN.

Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P., 2002, "Numerical Recipes in C", Cambridge University Press,.

Reeds, J.A. and Shepp, L.A., 1990, "Optimal Paths for a Car That Goes Both Forwards and Backwards", Pacific Journal of Mathematics.

9. Responsibility notice

The authors are the only responsible for the printed material included in this paper