New Trajectory Generation Methods for Nonholonomic Mobile Robots

Wenjie Dong and Yi Guo
Department of Electrical and Computer Engineering
University of Central Florida, Orlando, FL 32816, USA
Email: dongwj12@yahoo.com and yguo@ee.ucf.edu

ABSTRACT

We consider the problem of trajectory generation of nonholonomic mobile robots. We propose two trajectory generation algorithms, one uses a differential flatness based method and the other uses a polynomial input based method. Simulation results are shown for the proposed trajectory generation algorithms.

KEYWORDS: Trajectory generation, closed-loop control, mobile robot, nonholonomic system.

1. INTRODUCTION

Trajectory generation of nonholonomic mobile robots has been widely studied in the last decades [6][8][5]. Due to the nonholonomic constraints, trajectory generation of nonholonomic systems is not an easy task. Many well-developed trajectory generation methods for holonomic systems cannot be directly used to plan the motion of nonholonomic systems. New different methods have been proposed for the trajectory generation problem of nonholonomic systems, which includes differential geometric and differential algebra techniques, geometric phase, control input parameterization, and optimal control approach. The idea behind differential geometric and differential algebra techniques is to generate motions in the directions of iterated Lie brackets by employing typical inputs [10]. Monaco and Normand-Cyrot first proposed to use piece-wise constant inputs in trajectory generation in [9]. In [10], sinusoids are used as inputs in the trajectory generation. In [11][12], a trajectory generation algorithm is proposed for nonholonomic systems based on the concept of differential flatness. For differential flat nonlinear systems, the trajectory generation problem is equivalent to finding the output functions which satisfy the boundary conditions posed on the initial and final states. For nonholonomic Chaplygin systems, various techniques based on the different geometric phase were proposed in [1][3][7]. In the geometric phase approach, the trajectory generation problem is reduced to find an appropriate base space path to produce the desired geometric phase. In [2], the optimal trajectory generation is discussed.

In this paper, we consider the trajectory generation problem of mobile robots. We propose two trajectory generation methods. The first one is based on the differential flatness of nonholonomic mobile robots. We use polynomials as flat outputs. A second order and a fifth order polynomials are proposed for the two flat outputs. The second method is based on the well-known chained form and the polynomial input method. The control inputs are a constant and a second order polynomial, respectively. By integrating control inputs, the coefficients of the polynomials are determined by boundary conditions.

This paper is organized as follows. In Section 2, the problem discussed in this paper is defined. In Sections 3 and 4, a flatness based and a polynomial input based trajectory generation algorithms are proposed, respectively. In Section 5, simulation results are presented to show the effectiveness of the proposed algorithms. The last section concludes this paper.

2. MOBILE ROBOT MODEL AND PROBLEM STATEMENT

Consider a car-like mobile robot shown in Figure 1. The front wheels of the mobile robot are steering wheels and the rear wheels are driving wheels with a fixed forward orientation. The kinematic model of the mobile robot can be written as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ \tan\phi/l \\ 0 \end{bmatrix} \rho u_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_2 \qquad (1)$$

where $q = [x, y, \theta, \phi]^T$ is the system state, (x, y) represents the Cartesian coordinates of the middle point of the rear wheel axle, θ is the orientation of the robot body with respect to the X-axis, ϕ is the steering angle; l is the distance between the front and rear wheel-axle centers, ρ is the radius of rear driving wheel; u_1 is the angular velocity of the driving wheels, and u_2 is the steering velocity of the front wheels. $\phi \in (-\pi/2, \pi/2)$ due to the structure constraint of the robot.

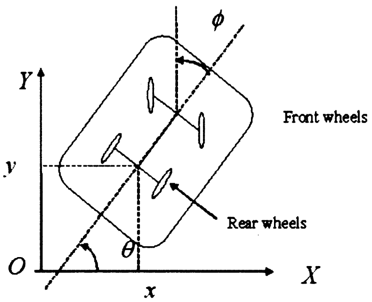

Figure 1. A Car-Like Robot

This paper considers the trajectory generation problem of the mobile robot from an initial point to a final point. Given an initial state $q(0)$ at time $t = 0$ and a final state $q(T)$ at time $t = T$, the problem discussed in this paper is as follows.

Trajectory Generation: Generate a feasible trajectory for the state q from the initial state $q(0)$ at time $t = 0$ to the final state $q(T)$ at time $t = T$ accounting for nonholonomic kinematics.

In the following sections, we propose two trajectory generation methods with the aid of different techniques.

3. TRAJECTORY GENERATION BASED ON DIFFERENTIAL FLATNESS

The concept of differential flatness was introduced by Fliess *et al.* in [4]. An important property of flat systems is that we can find a set of outputs such that all states and inputs can be expressed in terms of these outputs and their derivatives. Differential flatness has been widely used in trajectory generation, controller design and other areas. In this section, we use the differential flatness concept to our trajectory generation.

From (1), we have

$$\frac{dy}{dx} = \tan\theta, \qquad \frac{d^2y}{dx^2} = \frac{\tan\phi}{l\cos^3\theta}, \tag{2}$$

Therefore, θ and ϕ can be calculated from dy/dx and d^2y/dx^2, which means that system (1) is differentially flat [4]. For the flat outputs x and y, we have the following boundary conditions

$$x(0) = x_0, x(T) = x_f, \tag{3}$$

$$y(0) = y_0, \frac{dy}{dx}|_{t=0} = \tan\theta_0, \frac{d^2y}{dx^2}|_{t=0} = \frac{\tan\phi_0}{l\cos^3\theta_0}, \tag{4}$$

$$y(T) = y_f, \frac{dy}{dx}|_{t=T} = \tan\theta_f, \frac{d^2y}{dx^2}|_{t=T} = \frac{\tan\phi_f}{l\cos^3\theta_f}. \tag{5}$$

Next, we propose an algorithm to generate the trajectory of system (1) from the initial state $(x_0, y_0, \theta_0, \phi_0)$ at time $t = 0$ to the final state $(x_f, y_f, \theta_f, \phi_f)$ at time $t = T$ and such that $|\dot{x}(t)| \geq \epsilon(> 0)$. For x, noting the boundary condition (3), we choose

$$x = x_0(T-t)/T + x_f t/T + at(t-T) \tag{6}$$

where a is a constant determined by

$$|(x_f - x_0)/T + a(2t - T)| \geq \epsilon(> 0), \forall t \in (0, T). \tag{7}$$

One pair of a and ϵ which satisfy (7) is

$$a = \frac{|x_f - x_0|}{2T^2}, \epsilon = \frac{|x_f - x_0|}{2T}. \tag{8}$$

Therefore, the trajectory generated for x is

$$x = \frac{x_0(T-t)}{T} + \frac{x_f t}{T} + \frac{|x_f - x_0|}{2T^2}t(t-T). \tag{9}$$

For y, noting the boundary conditions (4) on the initial point, we choose

$$y = y_0 + \alpha_1\tan\theta_0 t + \frac{\alpha_2\tan\phi_0}{2l\cos^3\theta_0}t^2 + b_1t^3 + b_2t^4 + b_3t^5 \tag{10}$$

where

$$\alpha_1 = \frac{2(x_f - x_0) - |x_f - x_0|}{2T}, \alpha_2 = \frac{|x_f - x_0|}{T^2},$$

$b = [b_1, b_2, b_3]^T$ is a constant vector determined by the boundary conditions. By (5), we have

$$b_1T^3 + b_2T^4 + b_3T^5 = y_f - y_0 - \alpha_1\tan\theta_0 T - \frac{\alpha_2\tan\phi_0}{2l\cos^3\theta_0}T^2,$$

$$3b_1T^2 + 4b_2T^3 + 5b_3T^4 = \alpha_3\tan\theta_f - \alpha_1\tan\theta_0 - \frac{\alpha_2\tan\phi_0}{l\cos^3\theta_0}T,$$

$$6b_1T + 12b_2T^3 + 20b_3T^3 = \frac{\alpha_2\tan\phi_f}{l\cos^3\theta_f} - \frac{\alpha_2\tan\phi_0}{l\cos^3\theta_0},$$

where $\alpha_3 = (2(x_f - x_0) + |x_f - x_0|)/(2T)$. Let

$$A = \begin{bmatrix} T^3 & T^4 & T^5 \\ 3T^2 & 4T^3 & 5T^4 \\ 6T & 12T^2 & 20T^3 \end{bmatrix},$$

$$c = \begin{bmatrix} y_f - y_0 - \alpha_1\tan\theta_0 T - \dfrac{\alpha_2\tan\phi_0}{2l\cos^3\theta_0}T^2 \\ \alpha_3\tan\theta_f - \alpha_1\tan\theta_0 - \dfrac{\alpha_2\tan\phi_0}{l\cos^3\theta_0}T \\ \dfrac{\alpha_2\tan\phi_f}{l\cos^3\theta_f} - \dfrac{\alpha_2\tan\phi_0}{l\cos^3\theta_0} \end{bmatrix}$$

Since $T \neq 0$, A is nonsingular. Therefore, b has a unique solution, i.e.,

$$b = A^{-1}c. \tag{11}$$

By (2) , we have

$$\theta = \tan^{-1}(2T^2(\alpha_1 \tan\theta_0 + \frac{\alpha_2 \tan\phi_0}{l\cos^2\theta_0} + 3b_1t^2 + 4b_2t^3$$
$$+5b_3t^4)/(2T(x_f - x_0) - T|x_f - x_0| + 2|x_f - x_0|t)), \quad (12)$$

$$\phi = \tan^{-1}\left(\frac{T^2(\frac{\alpha_2 \tan\phi_0}{l\cos^2\theta_0} + 6b_1t + 12b_2t^2 + 20b_3t^3)}{|x_f - x_0|}\right). \quad (13)$$

Since

$$|(2T(x_f - x_0) - T|x_f - x_0| + 2|x_f - x_0|t)| \geq \epsilon(\forall t \in [0, T]),$$

θ is well-defined. Up to now, the trajectory of state (x, y, θ, ϕ) is generated from the initial state $(x_0, y_0, \theta_0, \phi_0)$ at time $t = 0$ to the final state $(x_f, y_f, \theta_f, \phi_f)$ at time $t = T$. In summary, we propose the following trajectory generation algorithm.

Trajectory Generation Algorithm 1:

Step 1: Using boundary conditions, generate trajectory of x according to (9);

Step 2: Calculate b and generate trajectory y according to (10);

Step 3: Generate trajectories of θ and ϕ according to (12) and (13), respectively.

From the planned motion, we can obtain u_1 and u_2 over interval $[0, T]$ as follows.

$$\begin{aligned} u_1(t) &= \left(\frac{2(x_f - x_0) - |x_f - x_0|}{2T\rho} + \frac{|x_f - x_0|}{T^2}\right)\cos\theta \\ &\quad +(\alpha_1 \tan\theta_0 + \frac{\alpha_2 \tan\phi_0}{l\cos^3\theta_0}t + 3b_1t^2 + 4b_2t^3 \\ &\quad +5b_3t^4)\sin\theta/\rho \quad (14) \\ u_2(t) &= \dot{\phi}. \quad (15) \end{aligned}$$

where θ and ϕ are functions of time t. In order to make the system robust to the initial state errors and disturbances during the motion, closed-loop controller can be used. However, this paper will not discuss this problem.

Flatness based trajectory generation method has been proposed in several papers [4][11][12][14][13]. However, our trajectory generation is different from them. In our trajectory generation, we assume the two flat outputs are all functions of time t. While in the existing literature one output is assumed to be a function of another. Since the outputs are all polynomials of time, the orders of the polynomials with respect to time are lower than those in [14][13].

4. TRAJECTORY GENERATION BASED ON POLYNOMIAL INPUTS

In this section, we propose another method. In order to make the trajectory generation problem of system (1) easier, we convert (1) into the chained form. Let the state transformation

$$z_1 = x, z_2 = \frac{\tan\phi}{l\cos^3\theta}, z_3 = \tan\theta, z_4 = y \quad (16)$$

and the input transformation

$$v_1 = u_1\rho\cos\theta, v_2 = \frac{u_2 l\cos^2\theta + 3\sin\theta\sin^2\phi v_1}{l^2\cos^5\theta\cos^2\phi}, \quad (17)$$

system (1) is transformed into

$$\dot{z}_1 = v_1, \dot{z}_2 = v_2, \dot{z}_3 = z_2v_1, \dot{z}_4 = z_3v_1. \quad (18)$$

System (18) is the chained form which is introduced first in [10]. Since system (1) is equivalent to system (18) when $\theta \in (-\pi/2, \pi/2)$ and $\phi \in (-\pi/2, \pi/2)$, the trajectory generation problem of system (1) is equivalent to that of system (18). Therefore, we only consider the trajectory generation problem of system (18). Noting the transformation in (16), that system (1) moves from the initial state $(x_0, y_0, \phi_0, \theta_0)$ at time $t = 0$ to the final state $(x_f, y_f, \phi_f, \theta_f)$ at time $t = T$ is equivalent to that system (18) moves from the initial state $(z_{10}, z_{20}, z_{30}, z_{40})$ at time $t = 0$ to the final state $(z_{1.f}, z_{2.f}, z_{3.f}, z_{4.f})$ at time $t = T$ where

$$z_{10} = x_0, z_{1.f} = x_f, \quad (19)$$

$$z_{20} = \frac{\tan\phi_0}{l\cos^3\theta_0}, z_{30} = \tan\theta_0, z_{40} = y_0, \quad (20)$$

$$z_{2.f} = \frac{\tan\phi_f}{l\cos^3\theta_f}, z_{3.f} = \tan\theta_f, z_{4.f} = y_f. \quad (21)$$

We choose the inputs

$$v_1 = a_0, \qquad v_2 = b_0 + b_1t + b_2t^2, \quad (22)$$

by (18), we have

$$\begin{aligned} z_1(t) &= z_{10} + a_0t, \quad (23) \\ z_2(t) &= z_{20} + b_0t + \frac{1}{2}b_1t^2 + \frac{1}{3}b_2t^3, \quad (24) \\ z_3(t) &= z_{30} + a_0z_{20}t + \frac{1}{2}a_0b_0t^2 + \frac{1}{6}a_0b_1t^3 \\ &\quad +\frac{1}{12}a_0b_2t^4, \quad (25) \\ z_4(t) &= z_{40} + a_0z_{30}t + \frac{1}{2}a_0^2z_{20}t^2 + \frac{1}{6}a_0^2b_0t^3 \\ &\quad +\frac{1}{24}a_0^2b_1t^4 + \frac{1}{60}a_0^2b_2t^5. \quad (26) \end{aligned}$$

For z_1, by the boundary conditions (19), we have

$$a_0 = (z_{1.f} - z_{10})/T. \quad (27)$$

For z_2, z_3 and z_4, by the boundary conditions (21), we have

$$b = M^{-1}d \quad (28)$$

where $b = [b_0, b_1, b_2]^T$,

$$M = \begin{bmatrix} T & \frac{1}{2}T^2 & \frac{1}{3}T^3 \\ \frac{1}{2}a_0T^2 & \frac{1}{6}a_0T^3 & \frac{1}{12}a_0T^4 \\ \frac{1}{6}a_0^2T^3 & \frac{1}{24}a_0^2T^4 & \frac{1}{60}a_0^2T^5 \end{bmatrix},$$

$$d = \begin{bmatrix} z_{2,f} - z_{20} \\ z_{3,f} - z_{30} - a_0z_{20}T \\ z_{4,f} - z_{40} - a_0z_{30}T - \frac{1}{2}a_0^2z_{20}T^2 \end{bmatrix},$$

M is a nonsingular if $a_0 \neq 0$ and $T \neq 0$. Once a_i and b_i are obtained, the steering control and the motion of the state are determined. By the inverse transformations, we can obtain the open-loop steering control and motions of the original states which are omitted here for space limit. To sum up, the trajectory generation algorithm is as follows.

Trajectory Generation Algorithm 2:

Step 1: Transfer system (1) into chained form (18) with the transformation (16)-(17);

Step 2: Calculate a_0 and b according to (27) and (28), respectively;

Step 3: Generate the trajectory z according to (23)-(26);

Step 4: Generate trajectory q by the inverse transformation of (16)-(17).

It should be noted that the flatness based trajectory generation and the polynomial input based trajectory generation are different. In the flatness based trajectory generation, the flat outputs are polynomials, while in the polynomial input based trajectory generation the steering control inputs are polynomials.

In Algorithm 2 x is a linear function of t while x is a polynomial function of t with second order in Algorithm 1. In both algorithms, y is a polynomial with five order. In both algorithms, the initial and final values of θ and ϕ should not be $\pi/2$.

5. SIMULATION

To show effectiveness of the proposed trajectory generation methods, simulation results are presented in this section. For the mobile robot, we assume $l = 1m$ and $\rho = 0.4m$. Given the initial state $(0, 0, 0, 0)$, the final state $(5, 5, \pi/4, \pi/6)$, and $T = 5$, we generate trajectory of system (1) with the proposed methods.

With the given initial state and the final state, the trajectory of system (1) can be generated by the proposed two algorithms. Figure 2 shows the path of the robot in X-Y plane generated by Algorithm 1. Figure 3 and Figure 4 show trajectories θ and ϕ generated by Algorithm 1, respectively. These figures show that the proposed Algorithm 1 is effective.

For the second trajectory generation method, Figures 5-7 show the path of $x - y$, θ, and ϕ. The simulation results show that Algorithm 2 is effective. Figure 8 shows the pathes in Algorithms 1 and 2. From this figure, it can be seen that the path with Algorithm 1 is shorter than that with Algorithm 2.

For the initial state $(0, 0, 0, 0)$ and the final state $(5, 5, 0, 0)$, Figures 9-15 show the simulation results with Algorithm 1 and Algorithm 2. From these figures, the effectiveness of the algorithms are also shown.

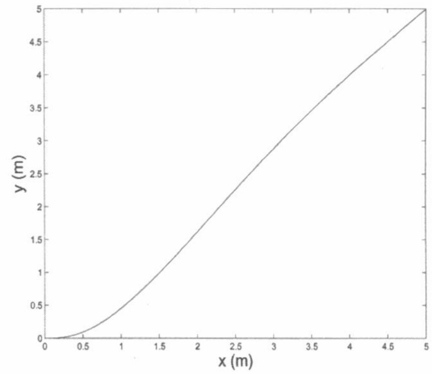

Figure 2. Path In X-Y Plane With Algorithm 1

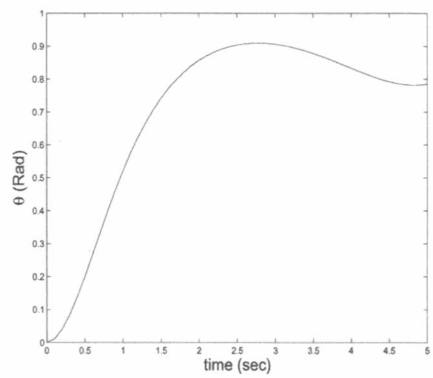

Figure 3. Trajectory θ With Algorithm 1

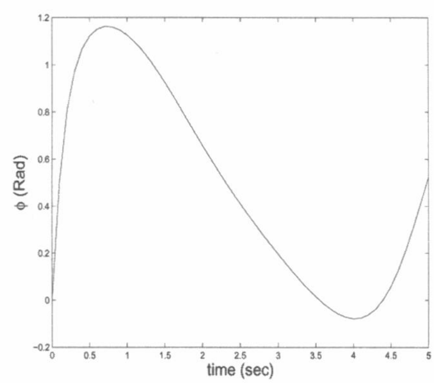

Figure 4. Trajectory ϕ With Algorithm 1

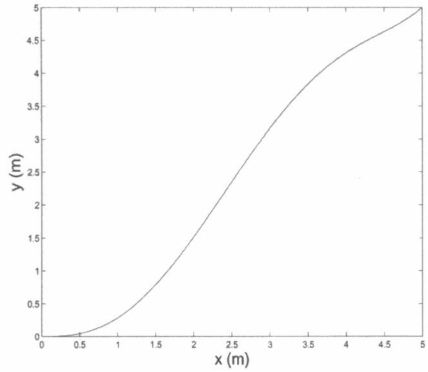

Figure 5. Path In X-Y Plane With Algorithm 2

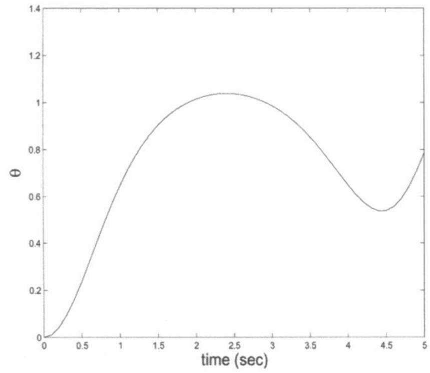

Figure 6. Trajectory Of θ With Algorithm 2

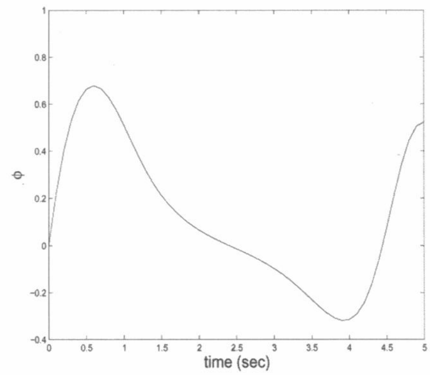

Figure 7. Trajectory Of ϕ With Algorithm 2

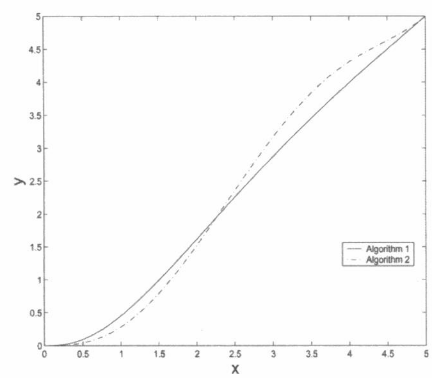

Figure 8. Pathes With Algorithm 1 And Algorithm 2

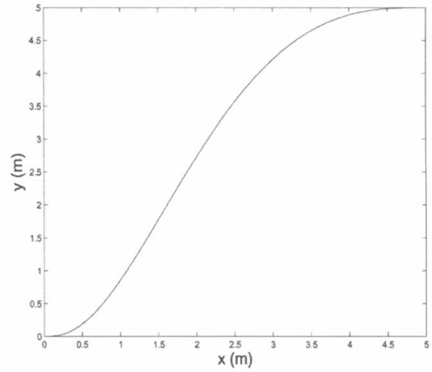

Figure 9. Path In X-Y Plane With Algorithm 1

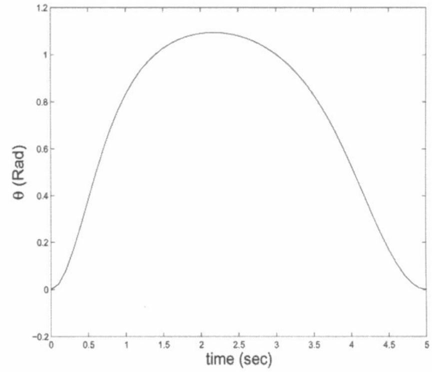

Figure 10. Trajectory θ With Algorithm 1

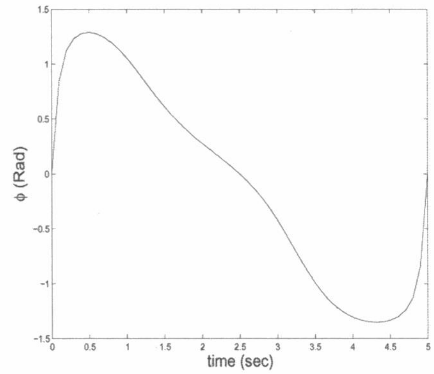

Figure 11. Trajectory ϕ With Algorithm 1

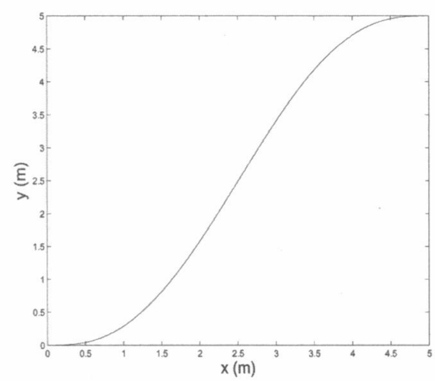

Figure 12. Path In X-Y Plane With Algorithm 2

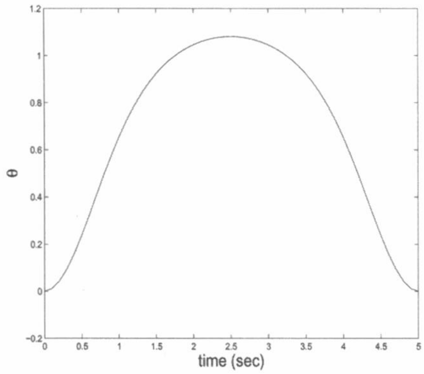

Figure 13. Trajectory Of θ With Algorithm 2

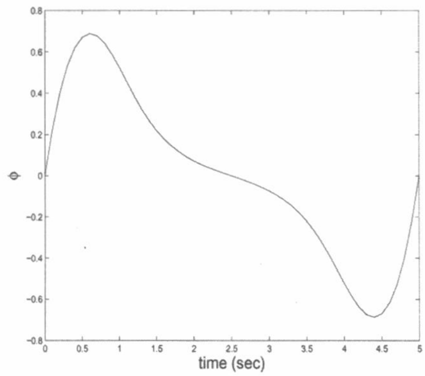

Figure 14. Trajectory Of ϕ With Algorithm 2

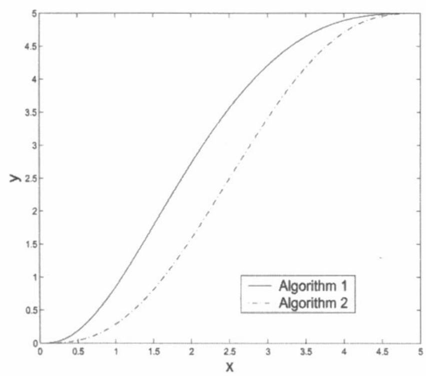

Figure 15. Pathes With Algorithm 1 And Algorithm 2

6. CONCLUSION

We considered the problem of trajectory generation of nonholonomic mobile robots. Flatness based and polynomial input based trajectory generation methods are proposed. Simulation results show effectiveness of the proposed trajectory generation methods.

REFERENCES

[1] Bloch, A., M. Reyhanoglu, and N.H. McClamroch, "Control and stabilization of nonholonomic dynamic system," *IEEE Trans. on Automatic Control,* Vol.37, No.11, 1992, pp.1746-1757.

[2] Brocket, R.W., and L. Dai, "Nonholonomic kinematics and the role of elliptic functions in constructive controllability," In *Nonholonomic Motion Planning*, Z. Li and J.F. Canny Eds, Kluwer, 1992, pp.1-22.

[3] Curvitz, L., and Z. Li, "Smooth time-periodic feedback solutions for nonholonomic motion planning," in *Nonholonomic Motion Planning,* Z.Li and J.F. Canny Eds, Kluwer, 1993, pp.53-108.

[4] Fliess, M,. J. Levine, P. Martin, and P. Rouchon, "Flatness and Defect of Nonlinear systems: Introductory Theoy and Examples," *Int. J. of Control,* Vol.61, No.6, 1995, pp.135-150.

[5] Kolmanovsky, I., and N. McClamroch, "Developments in nonholonomic control problems," *IEEE Control Systems Magazine,* Vol.15, No.6, 1995, pp.21-36.

[6] Laumond, J.-P., *Robot Motion Planning and Control,* Springer-Verlag, London, 1998.

[7] Mukherjee, R., and D.P. Anderson, "A surface integral approach to the motion planning of nonholonomic systems," *ASME Trans. JDSMC,* Vol.116, 1994, pp.315-364.

[8] Murray, R.M., Z. Li, and S.S. Sastry, *A Mathematical Introduction to Robotic Manipulation,* CRC Press, Inc., Boca Raton, 1994.

[9] Monaco, S., and D. Normand-Cyrot, "An introduction to motion planning under multirate digital control," *Proc. of Conf. on Decision and Control,* Tucson, AZ, 1992, pp.1780-1785.

[10] Murray, R., and S. Sastry, "Nonholonomic Motion Planning: Steering Using Sinusoids", *IEEE Trans. on Automatic Control,* Vol.38, 1993, pp.700-716.

[11] Rouchon, P., M. Fliess, J. Levin, and P. Martin, "Flatness and motion planning: the car with n trailers," *Proc. of the European Control Conference*, Groninger, Netherlands, 1993, pp.1518-1522.

[12] Rouchon, P., M. Fliess, J. Levin, and P. Martin, "Flatness, motion planning and trailer systems," *Proc. of the 32nd IEEE Conference on Decision and Control,* 1993, pp.2700-2705.

[13] Xu, W.L, B.L. Ma, and S.K. Tso, "Curve fitting Approach to Motion Planning of Chained Systems," *Proc. of IEEE Conf. on Robotics and Automation,* Michigan, 1999, pp.811-816.

[14] Wang, Y., "Nonholonomic Motion Planning: A Polynomial Approach," *Proc. of IEEE Conf. on Robotics and Automation,* Minneapolis, 1996, pp.2956-2961.