

Optimal Trajectory Generation and Flatness Tracking Control for a Mobile Robot

Amine Abadi, Hassen Mekki, Anis Ben Hadj Brahim, Adnen El Amraoui and Nacim Ramdani

Abstract—This work proposes an optimal trajectory generation and a controller design for a mobile robot. The objective consists of creating a new performance guidance module for the mobile robot in a closed environment. The reference trajectory can be calculated by solving an optimization problem. The properties of the differential flatness, the B-spline and the direct collocation method are exploited to convert the constrained optimization problem into a nonlinear programming one that can be easily resolved by a classic solver. After that, the obtained optimal reference trajectory is used in the kinematic mobile robot model. A flatness-based controller is designed to ensure the trajectory tracking. The numerical simulations are done in order to evaluate the proposed optimal trajectory generation approach and the tracking strategy.

I. INTRODUCTION

Nowadays, mobile wheeled robots have become an interesting area of research because they possess a lot of practical applications in hospital tasks, freight transport, agriculture, military, and mine clearance. To efficiently complete the different practical tasks, a highly performance guidance scheme needs to be developed for the wheeled mobile robot. Thus, the optimal trajectory generation and the tracking control have become an interesting problem in the recent years. In the literature, a lot of numerical methods [1] have been used for solving the optimal trajectory problems of mobile robot which are classified into two kinds: indirect and direct. In [2], an indirect solution of the optimization trajectory problem method was used to generate the trajectory with minimum energy for mobile robots. This approach consisted in solving the problem of the two-point limit value arising from the Pontryagins maximum principle. In the same context, Aditya [3] utilized the leapfrog algorithm and the necessary conditions of the Pontryagins maximum to determine the optimal trajectory. Direct methods are developed while avoiding the disadvantages of the two-point limit value resolution, by converting the path optimization problem to a Non Linear Programming NLP One. Hence, the state and control systems are based

on their discretization and implicit integration. Therefore, the optimization problem is solved using one of the NLP solvers such as Snopt, Fmincon, etc. The applications of the direct method for mobile robot were discussed in [4]-[6]. The disadvantage of direct and indirect methods is the large number of variables utilized during the optimization problem, which complicates the resolution task for solvers. To solve this problem, another approach was created to solve the optimization trajectory problem of quadrotor based on the differential flatness [7], where the problem dimensions were decreased to have a more computational performance. On the other hand, many researchers dedicated their efforts in an attempt to solve the tracking problem. In [8], Ibari designed a backstepping approach to ensure an autonomous trajectory tracking for the mobile robot. In [9], a feedback linearization method was developed in order to follow the reference trajectory for the mobile robot under the existence of longitudinal and lateral slips. In [10], Zhijun Li combined the predictive control and the neural network to ensure an online optimal tracking for the nonholonomic mobile robots. In [11], Ji-Chul Ryu developed a robust flatness tracking controller for the mobile robots despite the presence of slipping. In [12], Guillaume Allibert combined the predictive and flatness control to create a visual servoing for mobile robots. The role of this latter was to track references and avoid obstacles.

In this paper, we focus on the creation of a performance guidance module for the mobile robot in closed environment. Thus, two objectives are necessary. The first objective is to generate a smooth and feasible trajectory that allows avoiding static obstacles. In fact, an optimization problem will be created where the obstacles are defined as constraints and the jerk as a cost function. Thereby, the optimization problem will be complicated due to the number of constraints, and the problem resolution time will increase. Then, an approach based on the theory of the differential flatness, the B-spline function and the collocation algorithm is utilized to solve the optimization problem more easily. Secondly, a flatness-based tracking controller is designed for the robot mobile system to track the obtained optimal reference trajectories in spite of the existence of model uncertainty.

This document is structured as follows. In section II, we present the mobile robot model. In section III, we define the optimization trajectory problem. In section IV, we define the algorithm of optimal trajectory generation. Section V is devoted to the design of the tracking controller, and section VI deals with the simulation results.

A. Abadi, is with National Engineering School Of Sousse, University of Sousse, Networked Objects Control and Communication Systems Laboratory, Tunisia. amineabadi@hotmail.fr

H. Mekki is with National Engineering School Of Sousse, University of Sousse, Networked Objects Control and Communication Systems Laboratory, Tunisia. mekki.hassen@gmail.com

A. Ben Hadj Brahim is with National Engineering School Of Sousse, University of Sousse, bacha.anis@gmail.com

A. El Amraoui is with Univ. Orléans, INSA CVL, PRISME EA 4229, F45072 Orléans, France. adnen4@gmail.com

N. Ramdani is with Univ. Orléans, INSA-CVL, PRISME, EA 4229, F45072, Orléans, France. nacim.ramdani@univ-Orléans.fr

II. ROBOT MOBILE MODEL

The system used in our application is a unicycle mobile robot (Figure.1). It is composed of two driving wheels in the middle of the main body of the robot.

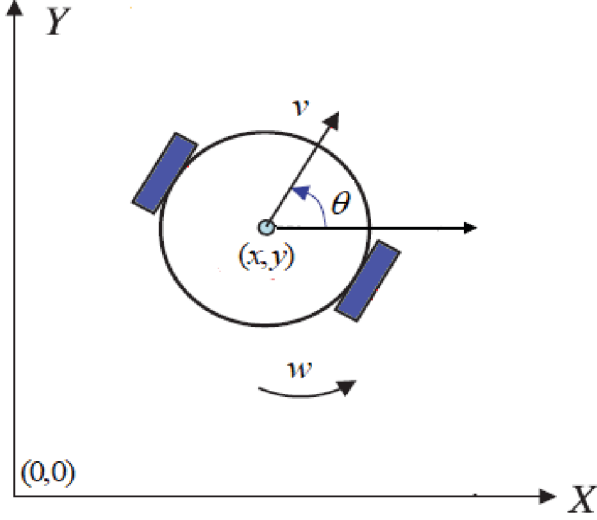


Figure.1 A Two-wheeled mobile robot

The system's configuration is defined in the cartesian coordinates by (x, y) which represent the center position of the robot and the orientation θ . According to the non slip condition, the non-holonomic constraint [13] is defined as follows:

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0 \quad (1)$$

$$\dot{x}\cos\theta + \dot{y}\sin\theta = r_m \quad (2)$$

where r_m is the wheel radius of the mobile robot and ϕ represent the wheel's angle measured from the vertical. Assuming that the linear velocity $v = r_m\dot{\phi}$ and the angular velocity $w = \dot{\theta}$, the kinematic model of the mobile robot is defined as follows:

$$\begin{cases} \dot{x}(t) = \cos(\theta)v \\ \dot{y}(t) = \sin(\theta)v \\ \dot{\theta} = w \end{cases} \quad (3)$$

where the state system of the mobile robot is $q = [x, y, \theta]^T$ and the control input is defined by v and w .

III. OPTIMAL TRAJECTORY GENERATION PROBLEM

Our target is to determine a smooth desired trajectory that allows the mobile robot to avoid obstacles in a closed environment. This problem is considered as an optimal control problem defined by the state constraints, the actuator constraints(control), the obstacle avoidance constraints and the cost function

A. State Constraint

The boundary conditions of our optimization problem are defined by $X(t_0) = [x_0, y_0, \theta_0]^T$ and $X(t_f) = [x_f, y_f, \theta_f]^T$. In order to improve the performance of the guidance module of the mobile robot, specific constraints are integrated in the model. These restrictions are represented in bounds on state variables, as follows:

$$\begin{cases} x_{min} \leq x \leq x_{max} \\ y_{min} \leq y \leq y_{max} \\ \theta_{min} \leq \theta \leq \theta_{max} \end{cases} \quad (4)$$

B. Obstacle constraint

A lot of solutions for the problem of avoiding an obstacle exist in the literature. Among these solutions is the integration of obstacles in an optimization problem. All the functions and equations used to formulate the optimization problem must be continuous and differentiable. Consequently, it is necessary to describe obstacles with smooth functions. Most work in the literature algebraically defines the obstacles by utilizing the p-norm [14] to design generic shapes such as: rectangles, circles, squares and ellipses. Figure.2 demonstrates how the p-norm can be used to design these obstacle shapes.

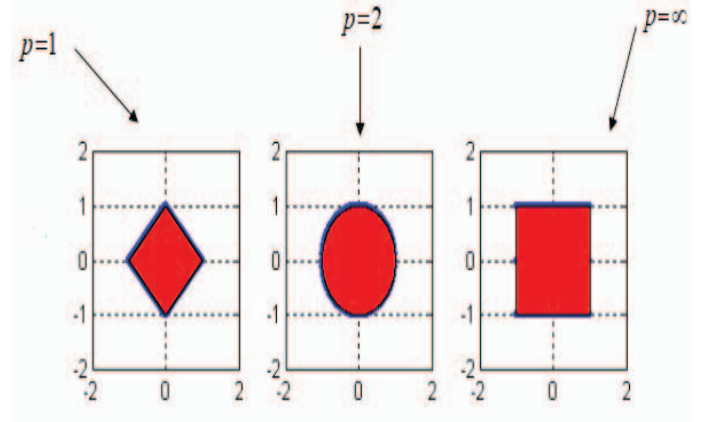


Figure.2 Unit p-norms

The equation $h(x, y)$ shows the general form used to create the obstacles' model

$$h(x, y) = \left(\frac{x - x_{ob}}{a}\right)^p + \left(\frac{y - y_{ob}}{b}\right)^p - 1 \quad (5)$$

where x_{ob} and y_{ob} represent the center of each obstacle. a represents the distance from the center to the obstacle boundary in the x-direction, and b represents the distance from the center to the obstacle boundary in the y-direction. When choosing $p = \infty$ the optimization problem will become unresolved computationally. For this reason, p is limited to 10. The constraint which means that the mobile robot avoids the obstacle is defined as $h(x, y) > 0$. The formula of $h(x, y)$ used previously does not take into account the size of the mobile robot, because it considers it as a point which is not practical. Hence, the new formulation

of $h(x, y)$ means that the mobile robot obstacle avoidance in a real case is defined as follows:

$$\left(\frac{x - x_{ob}}{a}\right)^p + \left(\frac{y - y_{ob}}{b}\right)^p - 1 > R_s \quad (6)$$

The radius R_s integrated in the obstacle representation allows additional safety around the obstacle zone.

C. Cost

Choosing a good cost function is important for the definition of the optimal problem. This choice depends usually on our needs. Based on the results of Hogan [15], generating smooth trajectories necessitates minimizing of the jerk, which is defined as follows:

$$J = \int_{t_0}^{t_f} (\ddot{x}^2 + \ddot{y}^2)^2 dt \quad (7)$$

IV. OPTIMAL TRAJECTORY GENERATION APPROACH

In the direct collocation method, an immense number of optimization variables is used due to the rich discretization on trajectory optimization problems. These optimization variables result in a long solution time. Thus, we should find an approach to reduce these variables so as to decrease the solution time without any lessening in solution accuracy. In this section, an approach is presented for the rapid solving of optimal trajectory problems. This approach is based on the simultaneous use of differential flatness to decrease the dimensional space of optimal problems, B-spline curves to approximate trajectories and a collocation method to transform an optimal problem in an NLP one.

A. Optimal problem in flat output space

The differential flatness introduced by Fliess [16] reduces the dimension of the optimization problem defining the flat variables. The following nonlinear system

$$\dot{x} = f(x, u), x \in \mathbb{R}^n, u \in \mathbb{R}^m \quad (8)$$

is differentially flat if we can find the following outputs :

$$z = \xi(x, u, \dot{u}, \dots, u^{(r-1)}) \quad (9)$$

and

$$x = \gamma(z, \dot{z}, \ddot{z}, \dots, z^{(\alpha)}) \quad (10)$$

$$u = \gamma(z, \dot{z}, \ddot{z}, \dots, z^{(\alpha)}) \quad (11)$$

where α and r are finite multi-indices and ξ, γ are two smooth vector functions .

The first step of the approach is to transform the optimal trajectory problem to the flat output space, so let consider the following optimal trajectory problem as follows:

$$\min_{x, u} J(x(t), u(t)) \begin{cases} \dot{x} = f(x, u) \\ x(t_0) = x_0, x(t_f) = x_f \\ u(t_0) = u_0, u(t_f) = u_f \\ L \leq \lambda(x(t), u(t)) \leq M \end{cases} \quad (12)$$

where $\dot{x} = f(x, u)$ represents the flat non linear system, $\lambda(x(t), u(t))$ represents the trajectory constraint, and L and M are the lower and upper bounds of constraints. The other constraints represent the initial and final conditions. Using the flatness property (10) and (11), problem (12) is transformed into a flat optimal problem of finding the flat output $t \rightarrow z_i(t)$ such that:

$$\min_{\bar{z}_i} J(\bar{z}_i) \begin{cases} \bar{z}_i(t_0) = \bar{z}_{i0}, \\ \bar{z}_i(t_f) = \bar{z}_{if} \\ L \leq \lambda(\bar{z}_i) \leq M \end{cases} \quad (13)$$

with $\bar{z}_i = [z_i, \dot{z}_i, \ddot{z}_i, \dots, z_i^{(r)}]$, $i=1..n$, where n is the number of flat outputs and $\lambda(\bar{z}_i)$ is the trajectory constraint in the flat outputs.

B. Approximation of flat variables

To get feasible trajectories, we need to set all the flat outputs as appropriate functions. In addition to selecting a good function for the flat output with a reasonable number of parameters, it is necessary to define the level of continuity C^k and to estimate the effect of each parameter in order to quickly find a curve corresponding to the right solution. Several curves can be used to specify the flat output (Fourier series, polynomials, ...). Actually, it is possible to express the flat output in terms of a finite-dimensional B-spline. The B-spline formalism [17] proposes a suitable solution for defining piecewise polynomial trajectories and offering good elasticity due to the small number of parameters. Therefore, the B-spline curves are defined in the following way:

$$z_i(t) = \sum_{k=1}^{N_c} B_{k,r}(t) \alpha_{ik}, i = 1, \dots, n \quad (14)$$

where α_{ik} is the control parameters, N_c represents the number of control parameters which is defined by $N_c = N * (r - s) + s$, N is the number of intervals or the number of polynomial pieces, r is the degree of the polynomial pieces, s is the smoothness condition at breakpoints, and $B_{k,r}(t)$ is the basis function. After the approximation of the flat outputs, α_{ik} becomes the decision variables for the flat optimal problem (13)

C. Transferring flat optimal problem to non linear programming problem

The third step consists in transforming problem (13) into an NLP problem using the direct collocation method, which is to break the time domain into smaller intervals: $[t_0, \dots, t_f = t_N]$. According to the evaluation rules from the study of Mark B. Milam [18], the B-splines are defined in

equation (15) when evaluated at the collocation points t_j :

$$\begin{bmatrix} z_i(t_0) \\ \dot{z}_i(t_0) \\ \vdots \\ z_i^{T_i}(t_0) \\ \vdots \\ z_i(t_N) \\ \dot{z}_i(t_N) \\ \vdots \\ z_i^{T_i}(t_N) \end{bmatrix} = B_i \begin{bmatrix} \alpha_{i1} \\ \vdots \\ \alpha_{iN_c} \end{bmatrix} \quad (15)$$

where B_i is the collocation matrix based on the B-spline basis functions, so the constraints will be enforced to a finite number of points t_j called collocation points. The choice of collocation points is then a degree of freedom in the process of optimal trajectory generation, which needs a certain technical expertise in the field of technological constraints. Finally, the NLP problem obtained according to the control parameter is:

$$\min_{\bar{\alpha}} J(\bar{\alpha}) \begin{cases} \alpha_{i1} = z_{i0}, \\ \alpha_{iN_c} = z_{if}L \leq A(\bar{\alpha}) \leq M \\ B(\bar{\alpha}) = 0 \end{cases} \quad (16)$$

with

$$\bar{\alpha} = (\alpha_{11}, \dots, \alpha_{1N_c}, \alpha_{21}, \dots, \alpha_{2N_c}, \dots, \alpha_{n1}, \dots, \alpha_{nN_c})$$

where $A(\bar{\alpha})$ and $B(\bar{\alpha})$ are equality and inequality constraints in function of control parameters. Finally, the last step is to use some solvers to resolve the NLP problem (16) such as FMINCON and NPSOL.

V. FLATNESS-BASED TRACKING CONTROL

A flatness controller is designed so that the mobile robot system can follow the optimal reference trajectories obtained by the approach explained in the previous section. For the robot mobile model (3), we choose $z_1 = x$, $z_2 = y$ as flat outputs.

An input prolongation is utilized to design a diffeomorphism between the states and the flat outputs and their derivatives. Assuming that v is an additional state, the new systems are defined as follows:

$$\begin{cases} \dot{x}(t) = \cos(\theta)v \\ \dot{y}(t) = \sin(\theta)v \\ \dot{v} = u_1 \\ \dot{\theta} = u_2 \end{cases} \quad (17)$$

where the state system of the mobile robot is $X = [x, y, v, \theta]^T$ and the new control input is defined by $u_1 = \dot{v}$ and $u_2 = \dot{\theta}$. According to equation (17), we can express the state and the input in terms of flat output and their derivatives.

$$\theta = \arctan \frac{\dot{z}_2}{\dot{z}_1} \quad (18)$$

$$v = \sqrt{\dot{z}_1^2 + \dot{z}_2^2} \quad (19)$$

$$u_1 = \frac{\dot{z}_1 \ddot{z}_1 + \dot{z}_2 \ddot{z}_2}{\sqrt{\dot{z}_1^2 + \dot{z}_2^2}} \quad (20)$$

$$u_2 = \frac{\dot{z}_1 \ddot{z}_2 - \dot{z}_2 \ddot{z}_1}{\dot{z}_1^2 + \dot{z}_2^2} \quad (21)$$

Let z_{1ref} and z_{2ref} be the reference trajectories for the flat outputs z_1 and z_2 . Since our objective is to track references, the flatness control will be designed by substituting the flat output by their reference flat output as follows:

$$u_1 = \frac{\dot{z}_{1ref} \ddot{z}_{1ref} + \dot{z}_{2ref} \ddot{z}_{2ref}}{\sqrt{\dot{z}_{1ref}^2 + \dot{z}_{2ref}^2}} \quad (22)$$

$$u_2 = \frac{\dot{z}_{1ref} \ddot{z}_{2ref} - \dot{z}_{2ref} \ddot{z}_{1ref}}{\dot{z}_{1ref}^2 + \dot{z}_{2ref}^2} \quad (23)$$

The flatness open-loop control defined by equations (22) and (23) ensures the tracking of the optimal trajectory provided that the model is exact, which is not the case in real applications. Indeed, the mobile robot can contain uncertainty parameters due to error modelization. Consequently the performance of guidance law will be degraded. Hence, a feedback control is necessary to be designed in order to ensure the asymptotically converge of $e_i = z_{iref} - z_i$ ($i = 1, \dots, 2$) to zero despite the existence of uncertainty parameters. From the differential parameterization (22) and (23), a flatness-based closed-loop control can be obtained when replacing \dot{z}_{1ref} and \dot{z}_{2ref} by the auxiliary control inputs \bar{v}_1 \bar{v}_2 respectively in u_1 and u_2 as follows:

$$u_1 = \frac{\dot{z}_{1ref} \bar{v}_1 + \dot{z}_{2ref} \bar{v}_2}{\sqrt{\dot{z}_{1ref}^2 + \dot{z}_{2ref}^2}} \quad (24)$$

$$u_2 = \frac{\dot{z}_{1ref} \bar{v}_2 - \bar{v}_1 \dot{z}_{2ref}}{\dot{z}_{1ref}^2 + \dot{z}_{2ref}^2} \quad (25)$$

With

$$\begin{aligned} \bar{v}_1 &= \dot{z}_{1ref} + K_{12}(\dot{z}_{1ref} - \dot{z}_1) + K_{11}(z_{1ref} - z_1) \\ \bar{v}_2 &= \dot{z}_{2ref} + K_{22}(\dot{z}_{2ref} - \dot{z}_2) + K_{21}(z_{2ref} - z_2) \end{aligned}$$

The gains K_{ij} can be determined by using pole-placement techniques

VI. SIMULATION AND RESULTS

In this section, we present the results simulation of the control and optimization method based on flatness. As an example of application, it is desired to generate the trajectory which allows the mobile robot to move from an initial state $X(0) = [0, 0, 0, 45]^T$ to a final state $X(10) = [8, 10, 0, 60]^T$ in a room full of obstacles. The obtained trajectory must be minimal in Jerk and can avoid static obstacles. In addition, the constraints of states must be respected as follows:

$$\begin{aligned} 0 &\leq x \leq 10 \\ 0 &\leq y \leq 10 \\ -90^\circ &\leq \theta \leq 90^\circ \end{aligned} \quad (26)$$

Three circles and rectangular blocks are considered as static obstacles. The representation model of these are defined as follows:

$$h_1(x, y) = (x - 4)^2 + (y - 4)^2 - 1 \quad (27)$$

$$h_2(x, y) = (x - 7)^2 + (y - 6)^2 - 1 \quad (28)$$

$$h_3(x, y) = (x - 6)^2 + (y - 9)^2 - 1 \quad (29)$$

$$h_4(x, y) = \left(\frac{x-2.5}{2.5}\right)^{10} + (y - 7)^{10} - 1 \quad (30)$$

Let consider $R_s = 0.1 m$. According to equation (6), (27), (28), (29) and (30) the obstacle constraints integrated in the optimisation problem are as follows:

$$(x - 4)^2 + (y - 4)^2 - 1 > 0.1 \quad (31)$$

$$(x - 7)^2 + (y - 6)^2 - 1 > 0.1 \quad (32)$$

$$(x - 6)^2 + (y - 9)^2 - 1 > 0.1 \quad (33)$$

$$\left(\frac{x-2.5}{2.5}\right)^{10} + (y - 7)^{10} - 1 > 0.1 \quad (34)$$

After formulating the optimization problem in terms of flat output, we choose for each flat output z_1 and z_2 , the curve B-spline that is defined by the number of intervals $n = 1$, the smoothness condition $s = 11$ and the order of the polynomial $r = 12$. Thirty points of collocation are chosen to discretize the interval $[0, \dots, 10]$ by the trapezoidal method. Figure.3 and Figure.4 depict the output response of the system and the flatness control.

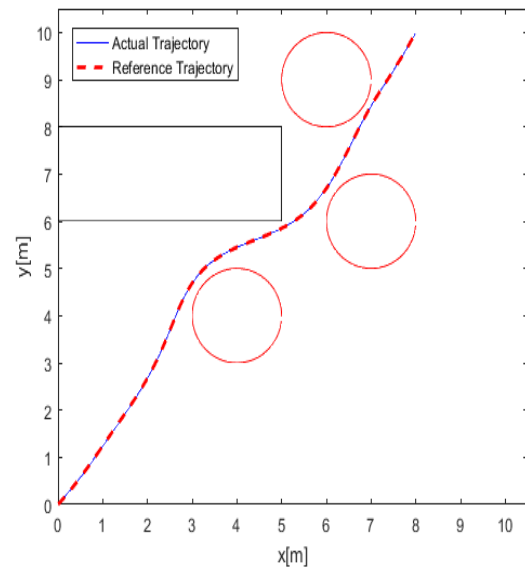
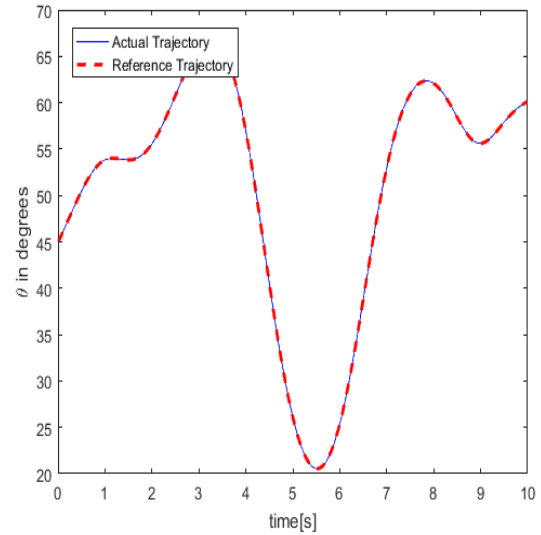
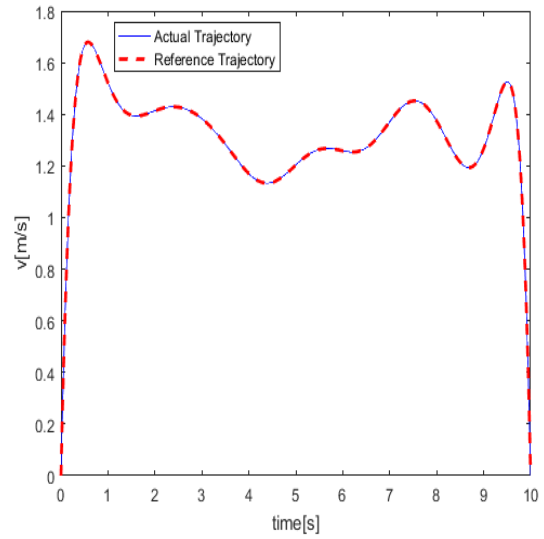
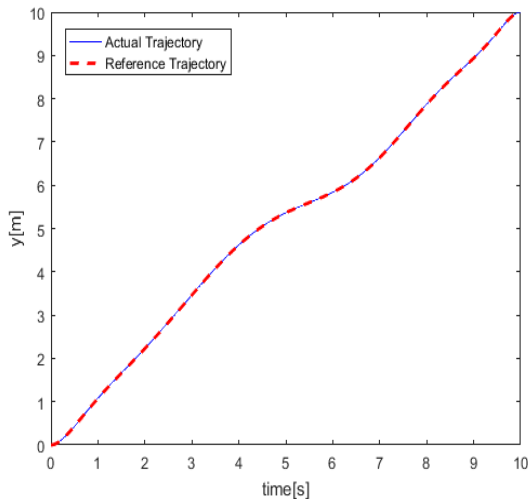
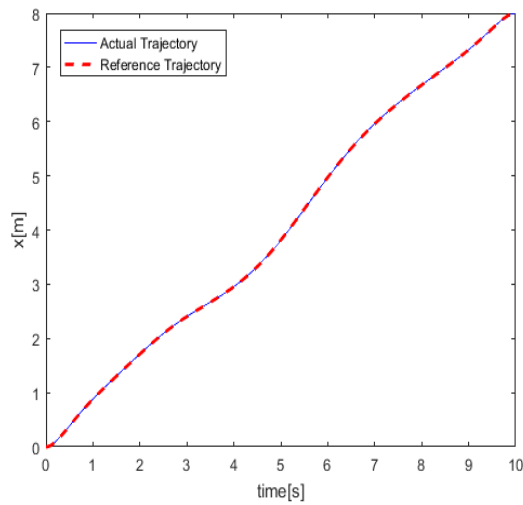


Figure.3 Results of generation and tracking optimal trajectory

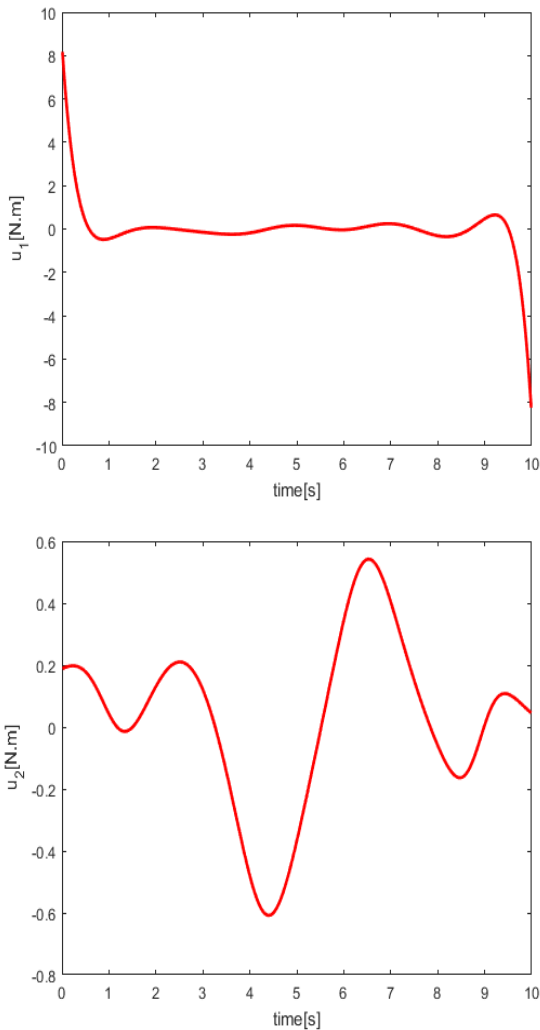


Figure.4 Flatness Control

Figure.3 shows that the trajectory defined for the states satisfies the optimization problem constraints. In addition , it can be observed that the flatness control defined by (24) and (25) permits a good tracking of the desired trajectory for the mobile robot. Thus, the flatness property is considered as a powerful tool for designing new performance guidance modules for the mobile robot in closed environment.

VII. CONCLUSION

The problem of generating and tracking optimal trajectories for a mobile robot was been treated in this paper. The simulation results indicate that the optimal trajectory generation algorithm based on the property of flatness and on the direct collocation method is efficient. Choosing the jerk as a cost can greatly decrease the error tracking trajectories and reduce the difficulty of designing control laws. Transforming the problem of obstacle avoidance as an optimization problem facilitates obtaining a trajectory that avoids static obstacles. In the second part, a flatness control has proven its effectiveness for the tracking of the created optimal trajectory. Our future work will include the slipping

and skidding phenomenon for the mobile robot and will focus on the avoidance of dynamic obstacles.

REFERENCES

- [1] A. V. Rao. A survey of numerical methods for optimal control. *Advances in the Astronautical Sciences*, vol. 135, no. 1, p. 497, 2009.
- [2] M. Nazemizadeh , H. N. Rahimi, K. Amini. Planning of mobile robots using indirect solution of optimal control method in generalized point-to-point task. *International Journal of Frontiers of Mechanical Engineering* 2012.
- [3] A Vamsikrishna, Arun D. Mahindrakar and Shaligram Tiwari. Numerical and Experimental Implementation of Leapfrog Algorithm for optimal Control of a Mobile Robot. *Indian Control Conference (ICC)* 2017.
- [4] Dan Wang, Weizhong Zhang. Trajectory Optimization and Tracking Control of Mobile Robots based on DMOC. *Chinese Control Conference July 28-30, 2014, Nanjing, China*.
- [5] R. Palamakumbura, D.H.S. Maithripala, C. F. Martin. Minimum Jerk Trajectory Generation for Differential Wheeled Mobile Robots". *International Conference on Industrial and Information Systems 2013, Sri Lanka*.
- [6] Naser Azim , Ahmad Fakharianb. Direct Optimal Motion Planning for Omni-directional Mobile Robots under Limitation on Velocity and Acceleration. *Journal of Optimization in Industrial Engineering* 2017.
- [7] Jing Yu, Zhihao Cai, Yingxun Wang M. Optimal Trajectory Generation of a Quadrotor Based on the Differential Flatness. *Chinese Control and Decision Conference (CCDC)* 2016.
- [8] Benaoumeur Ibari, Laredj Benchikh, Amar Reda, Hani Elhachimi. Backstepping Approach for Autonomous Mobile Robot Trajectory Tracking. *Indonesian Journal of Electrical Engineering and Computer Science, IAES*, 2016.
- [9] Nguyen V. Tinh, Nguyen T. Linh, Pham T. Cat, Pham M. Tuan, Mai N. Anh, and Nguyen P. T. Anh. Modeling and Feedback Linearization Control of a Nonholonomic Wheeled Mobile Robot with Longitudinal Lateral Slips. *IEEE International Conference on Automation Science and Engineering (CASE)* Fort Worth, TX, USA, August 21-24, 2016.
- [10] Zhijun Li, Jun Deng, Renquan Lu, Yong Xu. Trajectory-Tracking Control of Mobile Robot Systems Incorporating Neural-Dynamic Optimized Model Predictive Approach. *IEEE Transaction on systems , Man ,and Cyberetics Systems* 2016.
- [11] Ji-Chul Ryu and Sunil K. Agrawal . Differential flatness-based robust control of mobile robots in the presence of slip. *The International Journal of Robotics Research* 2011.
- [12] Guillaume Allibert, Estelle Courtial, Youssoufi Tour. A flat model predictive controller for trajectory tracking in image based visual servoing. *International federation of automatic control* 2007.
- [13] Benaoumeur Ibari, Laredj Benchikh, Amar Reda Hani Elhachimi, Zoubir Ahmed-Foiti. Backstepping Approach for Autonomous Mobile Robot Trajectory Tracking . *Indonesian Journal of Electrical Engineering and Computer Science*, 2016.
- [14] Masanori Harada, Hideyuki Nagatay. Optimal Trajectory Generation and Tracking Control of a Single Coaxial Rotor UAV. *Ninth International Conference on Natural Computation (ICNC)* 2013.
- [15] Hogan, N. An organizing principle for a class of voluntary movements. *The Journal of Neuroscience* 4.11 (1984): 2745-2754.
- [16] Fliess, J. Levine, P. Martin et P. Rouchon. A Lie Backlund approach to equivalence and flatness of nonlinear systems. *IEEE Transactions on Automatic Control*, 44(5) : 922-937, Mai 1999.
- [17] C. de Boor. A pratical guide to spline. in *Applied Mathematical Sciences*, vol. 27. Springer, 1978.
- [18] Milam Mark B. Real-time optimal trajectory generation for constrained dynamical systems. *California Institute of Technology*, 2003.