Comparative Study of PI Controller and Model Based Predictive Control for Mobile Robot

Mendili Manel1,*, Bouani Faouzi2

1National Engineering School of Tunis, University of Tunis El Manar, Tunisia
2Laboratory of Analysis, Design and Control of Systems, LR11ES20, Tunis, Tunisia

Abstract This paper focuses on resolving the trajectory tracking problem of two wheeled mobile robot. We begin by presenting the kinematic model of the robot which is the base of the control law then we present a PI controller and a model predictive controller to solve the problem of trajectory tracking. We performed a comparison between the performances of the classical PI controller and the predictive controller which is an interesting approach that considers an explicit performance criterion and minimizes it during the computation of the control law. Simulation results are provided in order to show the effectiveness of model predictive control in the resolution of trajectory tracking problem.

Keywords Two Wheeled Mobile Robot, Trajectory Tracking, PI controller, Model-based Predictive Control

1. Introduction

The traditional control problems of trajectory-tracking and regulation have been extensively studied in the field of mobile robotics [1], [2]. It is common in mobile robotics to address control problems taking into account only the kinematic model. In [3], authors give the main reasons to choose the kinematic model instead of the dynamic one. In [4] kinematic models for diverse types of mobile robots are presented.

Using the kinematic model, the problem of trajectory-tracking has been solved in [5] using a proportion differential coefficient and a second step differential coefficient in the control algorithm. In [6] the remote control of the omnidirectional mobile robot is developed using a discrete-time strategy assuming a time-lag model of the robot. The problem of trajectory-tracking was solved in [7] by using a fuzzy control of trajectory after demonstrating the kinematic model and the minimal dynamic model which are used to develop the control law. Authors of [8] used an estimation strategy that predicts the future values of the system’s output based on the model of the robot.

There were many additional control techniques of the wheeled mobile robot which have limits and do not present good performances; they usually present undesirable oscillatory motions as results of the closed-loop trajectory. In addition, choosing the controller parameters in order to obtain good performances is very difficult. Therefore, model based predictive control appeared as an interesting approach to overcome the limits of traditional techniques particularly the problem of choosing parameters which is directly related to a cost function which is minimized to achieve an optimal control sequence.

In [9] a model-based predictive control is used to solve the trajectory-tracking problem using two approaches; nonlinear model-based predictive control and linear model-based predictive control.

Despite the fact that the model-based predictive control is not a new control method, works dealing with model-based predictive control and wheeled robots are few and scattered. The generalized predictive control is used in [10] to solve the trajectory-tracking problem. Although, the control in this work acts only on the angular velocity and the linear velocity is supposed constant. A nonlinear model of wheeled mobile robots is used for trajectory tracking in [11]. The problem is solved using a neural network that considers unknown obstacles in the configuration space. There were also works using nonlinear model of robot and nonlinear model predictive control as in [12] where the authors developed a state space representation of a nonlinear model predictive control and they applied it to the problem of point stabilization and trajectory tracking.

In this paper, we present a full kinematic model and we solve the trajectory-tracking problem of two wheeled mobile robot. We consider a PI controller [7], [13] and a predictive controller [9]. The performance of the proposed control strategies are contrasted through numerical simulations.

This paper is organized as follows: section 2 presents the kinematic model of two wheeled mobile robot. In section 3 a PI controller is presented to solve the problem of trajectory-tracking. In section 4 a predictive controller is
developed for the robot and section 5 presents the performances of the two controllers which are evaluated by means of numerical simulations and compared with each other. In section 6 some conclusions are presented.

2. Model of Two Wheeled Mobile Robot

A. Kinematic Model

To control robot motion, it seems necessary to have a kinematic model of the two wheeled mobile robot, robot scheme can be seen on fig.1 An orthonormal inertial bases \((o, \bar{x}, \bar{y})\) is fixed in the plane of the motion. We define \(P\) as a reference point and an arbitrary basis \(\{\bar{x}_1, \bar{x}_2\}\).

The position of the robot is than specified by three variables \(x, y\) and \(\theta\) which are regrouped in a vector \(X = [x \ y \ \theta]^T\) describing the posture of the robot where \(x\) and \(y\) are the coordinates of \(P\) in the inertial bases which is given by the vector \(\vec{OP}\):

\[
\vec{OP} = x\bar{x} + y\bar{y} \tag{1}
\]

and \(\theta\) is the orientation of the basis \(\{\bar{x}_1, \bar{x}_2\}\) with the respect to the inertial basis \(\{\bar{x}, \bar{y}\}\).

The kinematic model can be written as follows:

\[
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{\theta}
\end{bmatrix} =
\begin{bmatrix}
\cos \theta & 0 \\
\sin \theta & 0 \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
v \\
\omega
\end{bmatrix} \tag{2}
\]

where \(v\) denotes the linear velocity and \(\omega\) denotes the angular velocity and they are regrouped in a vector \(U = [v \ \omega]^T\).

B. Discretisation

As described in next sections, in PI controller and in model predictive controller the control law is computed in discrete time. Thus a discrete time representation of this model becomes a necessity. We consider then a sampling period \(T\), a sampling instant \(k\) and by applying the Euler’s approximate to the kinematic model we obtain these formulae.

\[
\begin{align*}
x(k+1) &= x(k) + Tv(k) \cos \theta(k) \\
y(k+1) &= y(k) + Tv(k) \sin \theta(k) \\
\theta(k+1) &= \theta(k) + To\omega(k)
\end{align*} \tag{3}
\]

The problem of trajectory tracking is stated as to find a control law to obtain

\[
X(k) - X_r(k) = 0 \tag{4}
\]

where \(X_r(k) = [x_r(k) \ y_r(k) \ \theta_r(k)]^T\) is a pre-known reference trajectory. We generally associate to this reference trajectory a virtual robot having the same model than the controlled one; then we obtain in discrete time.

\[
\begin{align*}
x_r(k+1) &= x_r(k) + Tv_r(k) \cos \theta_r(k) \\
y_r(k+1) &= y_r(k) + Tv_r(k) \sin \theta_r(k) \\
\theta_r(k+1) &= \theta_r(k) + To\omega_r(k)
\end{align*} \tag{5}
\]

We then linearize the system model by computing an error model with respect to a reference trajectory and we expand the kinematic model in Taylor series around the point \((X_r, U_r)\) at each sampling time \(k\).

\[
\begin{align*}
x_r + Tv_r \cos \theta_r + (x-x_r) - T(\theta-\theta_r) v_r \sin \theta_r + T(v-v_r) \cos \theta_r \\
y_r + Tv_r \sin \theta_r + (y-y_r) - T(\theta-\theta_r) v_r \cos \theta_r + T(v-v_r) \sin \theta_r \\
\theta_r + To\omega_r + (\theta-\theta_r) + T(\omega-\omega_r)
\end{align*} \tag{6}
\]

By using (5) and (6) we find the following discrete-time system model:

\[
\tilde{X}(k+1) = A(k)\tilde{X}(k) + B(k)\tilde{U}(k) \tag{7}
\]

with

\[
A(k) =
\begin{bmatrix}
0 & -Tv_r(k) \sin \theta_r(k) \\
1 & Tv_r(k) \cos \theta_r(k)
\end{bmatrix}
\]

\[
B(k) =
\begin{bmatrix}
Tv_r(k) \sin \theta_r(k) \\
T \cos \theta_r(k)
\end{bmatrix}
\]

where \(\tilde{X}(k) = X(k) - X_r(k)\) represents the error with respect to the reference car and \(\tilde{U} = U - U_r\) is its associated error control input.

3. PI Controller

PI controller is a simple controller which provides generally adequate results, thus it is the most used controller. It consists of two tuning parameters \((K_p, K_i)\) which represent the gains of the present error and the accumulation of past errors. The sum of these two actions adjusts the process input in order to assume a reduced tracking error.

The goal of the PI controller is to control the motor in such
a way that the robot reaches the desired target \( x_r \) and \( y_r \).

Solving equation (7) gives the following resolved kinematic control:

\[
sin(\theta_r)\frac{\dot{x}_r}{2T} + \cos(\theta_r)\frac{\dot{y}_r}{2T} = \frac{1}{2}\sin(\theta)\frac{\dot{x}}{2T} + \cos(\theta)\frac{\dot{y}}{2T}
\]

Solving equation (7) gives the following resolved kinematic control:

\[
s_i \sin(\theta) + c_i \cos(\theta) = \frac{1}{2} s_i \sin(\theta) + c_i \cos(\theta)
\]

\[
2s_i \cos(\theta) - 2c_i \sin(\theta)
\]

\[
x^* = x + x^*\theta_\alpha(k)
\]

\[
y^* = y + y^*\theta_\alpha(k)
\]

\[
\theta^* = \theta + \theta^*\theta_\alpha(k)
\]

\[
e_i(k) = \hat{x}_i(k) - \tilde{x}_i(k)
\]

\[
e_j(k) = \hat{y}_j(k) - \tilde{y}_j(k)
\]

\[
e_\theta(k) = \hat{\theta}_\theta(k) - \tilde{\theta}_\theta(k)
\]

4. Predictive Control

The predictive control is a method based on using the system model to obtain a control sequence by minimizing an objective function. The model is used to predict the system behavior over a prediction horizon. We minimize the objective function using the future outputs sequence which is calculated on a known horizon. We use for that only the values of inputs for current sampling interval. We do the same process for each sampling interval with the update of measurements [9].

After obtaining a linear model it is possible to develop the optimization problem, however we should suppose that \( U \) is different from 0 so that the linearization becomes controllable because the linearization around a stationary operating point makes the robot not moving then non controllable.

Let’s introduce these vectors:

\[
\bar{X}(k+1) = \begin{bmatrix}
\hat{X}(k+1/k) \\
\hat{X}(k+2/k) \\
\vdots \\
\hat{X}(k+N/k)
\end{bmatrix}, \quad \bar{U}(k) = \begin{bmatrix}
\hat{U}(k) \\
\hat{U}(k+1) \\
\vdots \\
\hat{U}(k+N-1)
\end{bmatrix}
\]

where \( N \) is the prediction horizon. Considering the weighting matrices \( Q \): for the error in the state and \( R \): for the control variables we can write the objective function to be minimized as follows.

\[
\sum_{j=1}^{N} \begin{bmatrix}
\tilde{X}(k+j/k)Q(j)\tilde{X}(k+j/k) + \tilde{U}(k+j/k)R(j)\tilde{U}(k+j/k)
\end{bmatrix}
\]

Introducing the vectors in (17) with the function (18) allows us to write the cost function:

\[
\phi(k) = \bar{X}^T(k+1)Q\bar{X}(k+1) + \bar{U}^T(k)R\bar{U}(k)
\]

\[
\bar{Q} = \text{diag}(Q; \ldots; Q) \quad \text{and} \quad \bar{R} = \text{diag}(R; \ldots; R)
\]

We write from (10)

\[
\bar{X}(k+1) = \bar{A}\bar{X}(k) + \bar{B}\bar{U}(k)
\]

\[
\bar{A} \quad \text{and} \quad \bar{B} \text{ are defined in (22) and (23) with } \alpha(k,j,l) \text{ is given by } \alpha(k,j,l) = \prod_{i=0}^{N-1} A(k+l)
\]

\[
\begin{bmatrix}
A(k) \\
A(k+1)A(k) \\
\vdots \\
\alpha(k,1,1)
\end{bmatrix}
\]

\[
\begin{bmatrix}
B(k) \\
A(k+1)B(k) \\
\vdots \\
\alpha(k,1,1)B(k)
\end{bmatrix}
\]

\[
\begin{bmatrix}
\alpha(k,2,1)B(k) \\
\alpha(k,2,2)B(k+1) \\
\vdots \\
\alpha(k,1,1)B(k) + \alpha(k,1,2)B(k+1)
\end{bmatrix}
\]

\[
\begin{bmatrix}
\alpha(k,2,1)B(k) + \alpha(k,2,2)B(k+1) + \ldots + 0 \\
\alpha(k,1,1)B(k) + \alpha(k,1,2)B(k+1) + \ldots + B(k+N-1)
\end{bmatrix}
\]

From (19) and (20) we can rewrite the objective function in a quadratic form:

\[
\phi(k) = \frac{1}{2} \bar{C}^T(k)H(k)\bar{U}(k) + \bar{f}(k)\bar{C}^T(k) + d(k)
\]

where

\[
H(k) = 2(\bar{B}^T\bar{Q}\bar{B} + \bar{R})
\]

\[
f(k) = 2\bar{B}^T\bar{Q}\bar{A}\bar{X}(k/k)
\]

\[
d(k) = \bar{X}^T(k/k)\bar{A}^T\bar{Q}\bar{A}\bar{X}(k/k)
\]

The matrix \( H(k) \) is a Hessian matrix, and it is always positive definite. It describes the quadratic part of the objective function, and the vector which describes the linear part is \( f(k) \). We note that \( d(k) \) is independent of \( \bar{U}(k) \) and has no influence in the determination of the optimal input thus (24) turns to a standard expression used in QP problems and the optimization problem to be solved at each sampling time.
5. Simulation Results

Predictive control and PI control are verified by means of computer simulations. The simulations are accomplished by tracking a desired position \((x, y, \theta)\) with inverse U, circular and lemniscates trajectories. In the tracking control of the robot the results of simulations are shown in figures 2 to 19 including trajectory tracking, rotation angle and linear velocity of mobile robot. The sampling period was set to \(T = 0.1\) s, in these cases there are no disturbances.

We consider the following values of weighting matrices

\[
\begin{align*}
\bar{Q} &= 0.5I_{(3N, 3N)} \\
\bar{R} &= I_{(2N, 2N)}
\end{align*}
\]

where \(I_{(3N, 3N)}\) and \(I_{(2N, 2N)}\) are identity matrix with dimension \((3N, 3N)\) and \((2N, 2N)\) respectively. The prediction horizon is chosen \(N = 15\).

A. The inverse U shape:

it is composed of three segments. The initial posture of reference trajectory is set \(X_r = [0; 0; 0]\) while the actual initial posture of robot is \(X = [0; 0; 0]\). Although the inverse U shape trajectory changes suddenly in path the predictive controller shows good trajectory tracking and the posture error is zero contrariwise PI controller keeps the posture error different from zero which is clear to deduct from figures 2 and 5. The figures 3 and 6 show that the predictive controller is faster than the PI one to join the variation of the rotation angle of the robot. The control input, linear velocity, in figure 4 swings around the fixed value \(0.1\) m/s despite the predictive control in figure 7 takes less than one second to converge towards the desired value.

---

![Figure 2](image1.png)

**Figure 2.** Trajectory tracking with PI controller

![Figure 3](image2.png)

**Figure 3.** Rotation angle
Figure 4. Control input

Figure 5. Trajectory tracking with Predictive controller

Figure 6. Rotation angle

Figure 7. Control input
B. Lemniscates trajectory:

It seems to be the most challenging trajectory tracking cases which has constant change of rotation radius with constant linear velocity. This trajectory is described by the following equations:

\[
\begin{align*}
x_d(t) &= 0.75 + 0.75 \sin \left( \frac{2\pi t}{50} \right) \\
y_d(t) &= 0.75 \sin \left( \frac{4\pi t}{50} \right)
\end{align*}
\]  

(25)

The robot’s angular varies according to these formulae:

\[
\theta_d(t) = \begin{cases} 
\tan^{-1} \frac{8\cos \left( 2\sin^{-1} \left( \frac{4x}{3} - 1 \right) \right)}{3 \sqrt{1 - \left( \frac{4x}{3} - 1 \right)^2}} & 0 \leq t < 12.5 \\
-\pi - \tan^{-1} \frac{8\cos \left( 2\sin^{-1} \left( \frac{4x}{3} - 1 \right) \right)}{3 \sqrt{1 - \left( \frac{4x}{3} - 1 \right)^2}} & 12.5 \leq t < 37.5 \\
\tan^{-1} \frac{8\cos \left( 2\sin^{-1} \left( \frac{4x}{3} - 1 \right) \right)}{3 \sqrt{1 - \left( \frac{4x}{3} - 1 \right)^2}} & t \geq 37.5
\end{cases}
\]  

(26)

The robot’s model starts from the initial position \( X = [0.75; 0.2; \frac{\pi}{3}] \) as initial condition. The trajectory tracking, the rotation angular and the linear velocity are shown in figures 8 to 13. These figures show the good trajectory tracking of the predictive control unlike the PI controller; the figure 8 shows that the posture error is kept different from zero counter to figure 11 the error is equal to zero. In addition to that the convergence of the control inputs to the fixed value 0.1 is faster in the figures 12 and 13 then those of the PI controller; the linear and the angular velocity are stabilized to the desired value at 0.1s. The predictive control eliminates also the oscillations present in the linear velocity (figure 9).

![Figure 8. Trajectory tracking with predictive controller](image-url)
Figure 9. Control inputs

Figure 10. Control inputs

Figure 11. Trajectory tracking with predictive controller
C. The circular trajectory:

We start from a desired linear velocity $v_r=0.1$ ms$^{-1}$ and angular velocity $\omega_r=0.1$ rds$^{-1}$. The equations that generate this trajectory are:

\begin{align}
\theta_r(t) &= \frac{\pi}{2} + \frac{t}{10} \\
x_r(t) &= 1 + \cos \left( \frac{t}{10} \right) \\
y_r(t) &= \sin \left( \frac{t}{10} \right)
\end{align}

(27)

The initial position of the desired trajectory is $X_r=[2; 0; \frac{\pi}{2}]$ circular trajectory tracking simulation curves are shown in figures 14 to 19. As like the other reference trajectories the predictive controller shows a capacity to track the trajectory without posture error and the control law converges rapidly to the desired values with a response time 0.09s.
Figure 14. Trajectory tracking with PI controller

Figure 15. Control inputs

Figure 16. Control inputs
6. Conclusions

This paper has presented an application of PI and model predictive control to solve problem of mobile robot trajectory tracking. Control algorithm has been applied for three kinds of reference trajectory (i.e. U-shape, infinity and circular) these trajectories performed by using Simulation program. The results show the effectiveness of the predictive control in comparison with the PI one to track the trajectories without posture errors and to converge to the desired control values (linear velocity and angular velocity) rapidly and without oscillations.

REFERENCES

[1] Bétourné, A. & Campion, G. “Dynamic Modelling and Control Design of a Class of Omnidirectional Mobile Robots”, Proceedings of the IEEE Int. Conference on Robotics and Automation, Minneapolis, US pp. 2810-2815, 1996.

[2] Campion, G.; Bastin, G. & D'Andréa-Novel, B. “Structural Properties and Classification of Kinematics and Dynamics Models of Wheeled Mobile Robots”, IEEE Transactions on Robotics and Automation, Vol.12, No. 1, pp. 47-61, 1996.
3. Guéchi, E. “Suivi de trajectoires d’un robot mobile non holonome: approche par modèle flou de Takagi-Sugeno et prise en compte des retards in Magnetism”, PhD, University of Valenciennes and Hainaut cambrésis, 2010.

4. Canudas, C.; Siciliano, B.; Bastin, G.; Brogliato, B.; Campion, G.; D’Andrea-Novel, B.; De Luca, A.; Khalil, W.; Lozano, R.; Ortega, R.; Samson, C. & Tomei, P. Theory of Robot Control. Springer-Verlag, London, 1996.

5. Weibing, Z.; Longjun, W.; Zaixin, L. & Xinqiao, D. “Research on Motion Control Algorithm for Mirobot Soccer Robot”, Proceedings of the IEEE International Conference on Robotics and Biomimetics, Guilin, China, December 2009.

6. Velasco-Villa M.; Alvarez-Aguirre, A. & Rivera-Zago, G. “Discrete-Time control of an omnidirectional mobile robot subject to transport delay” IEEE American Control Conference, New York City, USA, pp. 2171-2176, 2007.

7. Kecskés, I. & Odry, P. “Fuzzy Route Control of Dynamic Model of Four wheeled Mobile Robot”, 4th IEEE International Symposium on Logistics and Industrial Informatics, Smolenice, Slovakia, pp. 437-440, September 2012.

8. Velasco-Villa M.; del-Muro-Cuellar B. & Alvarez-Aguirre, A. “Smith-Predictor compensator for a delayed omnidirectional mobile robot”, 15th Mediterranean Conference on Control and Automation, Athens, Greece, T30-027, 2007.

9. Kühne, F.; Lages, W. & Gomes da Silva, J. “Mobile robot trajectory tracking using predictive control”, 2nd IEEE-RAS Latin American Robotics Symposium. São Luis, September 2005.

10. Ollero, A. & Amidi, O. “Predictive path tracking of mobile robots Application to the CMU Navlab”, Proceedings of the 5th IEEE International Conference on Advanced Robotics, Pisa, Italy, pp. 1081–1086, 1991.

11. Gómez-Ortega, J. & Camacho, E. F. “Mobile robot navigation in a partially structured static environment using neural predictive control”, Control Engineering Practice 4 (12): 1669–1679, 1996.

12. Essen, H. V. & Nijmeijer, H. “Non-linear model predictive control of constrained mobile robots”, European Control Conference, Porto, Portugal, pp. 1157–1162, 2001.

13. Corke, P. “Robotics, Vision and Control”, Springer Tracts in Advance Robotics Volume 73, Berlin Heidelberg, 2011.