Model Predictive Collision-Free Path Following Control for Nonholonomic Mobile Robots

Thai Thanh Hiep
Faculty of Transportation Engineering,
Ho Chi Minh City University of Technology
(HCMUT), VNU-HCM, Ho Chi Minh City,
Vietnam

Vo Duy Cong
Industrial Maintenance Training Center,
Ho Chi Minh City University of Technology
(HCMUT), VNU-HCM, Ho Chi Minh City,
Vietnam

Le Hoai Phuong
Industrial Maintenance Training Center,
Ho Chi Minh City University of Technology
(HCMUT), VNU-HCM, Ho Chi Minh City,
Vietnam

In this research, a model predictive collision-free path following controller is developed and applied for an omnidirectional mobile robot (OMR). The mobile robot is controlled to track a reference path while avoiding collision with obstacles. The path-following problem is reformulated into the regulation problem of an extended plant by introducing a virtual degree of freedom, the path parameter of a geometric reference curve. Then a Model Predictive Controller (MPC) is then applied to steer the mobile robot. The optimization cost function is established from the difference between the state of the robot and the parameter path. The solution of MPC can be obtained by repeatedly solving an optimal control problem (OCP) to reduce the optimization cost function to a minimum value, making the robot state as close to the state of the path as possible. Obstacle avoidance is considered by adding terms as a function of the gap between the mobile robot and the objects in front of the robot. Constraints on the states and inputs of the system are also easily considered in the optimal control problem of MPC. This makes the control inputs not exceed the allowable limits of the robot. Simulations are carried out to reveal the controller's efficiency and show how to choose the right parameters to synchronize path tracking and obstacle avoidance tasks.

Keywords: *Path-following problem, Model predictive control, Obstacle avoidance, Optimal Control Problem, Omnidirectional mobile robot (OMR).*

1. INTRODUCTION

Over the last two decades, nonholonomic mobile robots have gained more and more research interest from engineers and scientists. The advantages of an omni–directional mobile robot (OMR) are that it can sync–hronize the rotation motion (steering) and transla–tion motion (linear) in any direction on the ground, greatly improving the robot's flexibility to attain rapid target tracking and obstacle avoidance. So, it meets the growing requirement for high flexibility, high perfor–mance, and safety of various applications in practice, such as heal–thcare assistance [1], workshop assistance [2], home assistance [3], and domestic [4]. Omni–directional mobile robots are built using un-steered Omni wheels or mecanum wheels. In [5,6], two omni–directional mobile structures with mecanum wheels were developed.

In many automation applications, mobile robots must be able to move autonomously in a plant, labo–ratory, home,... (without any human assistance) [7]. So, the motion control problem is essential in many mobile robotic systems. Control problems are divided into three basic problems: set-point stabilization, trajectory trac–king, and path-following. The path-following problem is the more general problem; the other two problems can be considered as specific cases of it. In the path-following problem, the robots are driven to follow a predefined geometric curve while satisfying dynamic constraints along the path. Many studies have proposed solutions for the path-following problems to apply in many applications (applied for a two-link robot manipulator in [8], for a KUKA LWR IV robot in [9], for an n trailer vehicle in [10], for aircraft and marine vehicles in [11]). Many studies have also investigated the motion planning problems for omnidirectional mobile robots [12–14]. And in recent years, Model Predictive Control (MPC) has been applied for path-following tasks with many successful results [15-16].

MPC is also known as receding horizon control, in which a finite-horizon optimal control problem (OCP) is solved online at every control cycle. The optimal solution's first control action is used as the control input for the real system [20-23]. The main advantage of the MPC algorithm is that constraints on inputs and states are considered in the optimal problem. This helps to avoid exceeding the limits of the system (e.g., workspace) and control signals (e.g. velocity, acce–leration, etc.), and collision avoidance constraints [24-26]. Normally, obstacles must be considered in path-following applications, and the controller must synchronize the path-following and obstacle avoidance.

The problems of following a parameterized curve have been introduced in many studies that use the nonlinear MPC strategy to solve them. In [17], Yu et al. presented a nonlinear MPC scheme for the path-following problem by converting the problem into a

Received: November 2022, Accepted: March 2023
Correspondence to: Vo Duy Cong
Industrial Maintenance Training Center,
Ho Chi Minh City University of Technology, Vietnam
E-mail: congvd@hcmut.edu.vn
doi: 10.5937/fme2302192H

regulation problem. A Polytopic Linear Differential Inclusion Problem (PLDI) is used to choose a suitable terminal constraint and cost. An acar-like mobile robot is used in the simulation to confirm the control performance. Faulwasser and Findeisen [18] provided a general framework for solving constrained output path-following problems by designing a continuous-time predictive control. The transverse normal forms are analyzed and used to compute stabilizing terminal regions and end penalties. Two different cases of the path-following problem are investigated. In the first case, the velocity assignment for the reference evolution is not specified. The other case is extended with a velocity assignment. In [19], a path-following controller based on the kinematics model of underactuated vehicles is developed by combining the MPC controller with a nonlinear auxiliary control law. By assuming that the terminal set can be neglected in the case of unconstrained inputs, MPC controllers provide a global solution to the addressed constrained motion control problems. In [20], a real-time nonlinear model pre–dictive path-following controller is developed for a labo–ratory tower crane to move a load along a predefined geometric path. The MPC is adopted for an extended system where the time evolution along the path is an extra degree of freedom to be determined by the controller. In previous studies, the selection of the controller coefficients in the case of with and without obstacles has yet to be presented. When it is required to follow the path and avoid obstacles simultaneously, the coefficients in the cost function must be adjusted so that the robot does not stop when approaching the obstacle.

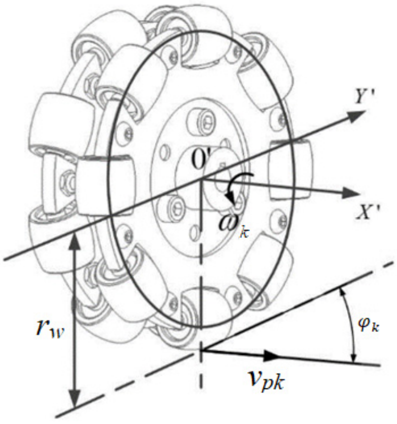


Figure 1. The structural principle of a Mecanum wheel [28]

In this work, a Model Predictive Collision-Free Path Following Control (MPCPFC) is developed to control an OMR robot. The mobile robot is driven to move along the path while avoiding collisions with obstacles. The parametrized path introduces a new virtual degree of freedom into the controller. The path-following problem is formulated into the regulation problem of the augmented plant. The input and state constraints are directly considered in the control design. Moreover, obstacle avoidance is achieved by adding constraints and penalty terms in the optimal function that depend on the relative position of the obstacles and the mobile robot. Simulation results are performed, thereby giving the correlation between the coefficients in the controller when considering collision avoidance.

The remainders of this paper are structured as fol–lows: Section 2 presents the kinematic of the omni–directional mobile robot; In section 3, The path follo–wing problem is reformulated as the regulation problem, and the model predictive control strategy is applied to solve the regulation problem by optimizing the cost function established from the error of the robot state and the state of the parameter path, obstacle avoidance is also simultaneously addressed by adding components related to the position of the obstacle in the cost function; Section 4 shows the simulation results to demonstrate the efficiency of the proposed method; Finally, Section 5 is the conclusion.

2. THE KINEMATICS OF OMNIDIRECTIONAL MOBILE ROBOT

In Figure 1, a mecanum wheel is constructed by a wheel connected to a motor and some passive rollers. A co–ordinate frame $x'O_ky'$ is attached to the wheel centroid. The X-axis has the same direction as the wheel's axis of rotation, the Y-axis is parallel to the ground. The wheel’s radius is denoted by r_w, the motor’s angular velocity is denoted by ωk, the velocity of the passive roller is denoted by υpk, and the υ_{pk} is perpendicular to the rotation axis of the rollers. The velocity of the mecanum wheels is a combination of the velocities of the motor and the rollers. Let $[\upsilon'_{kx}, \upsilon'_{ky}, \omega'_{\mathrm{k}}]^T$denote the velocity of the mecanum wheel’s centroid relative to $x'O_ky'$. It can be obtained from the relationship [25]:

$$\begin{bmatrix} v'_{kx} \\ v'_{ky} \end{bmatrix} = \begin{bmatrix} 0 & \sin\varphi_k \\ r_w & \cos\varphi_k \end{bmatrix} \begin{bmatrix} \omega_k \\ v_{pk} \end{bmatrix} \tag{1}$$

where φ_k is the deflection angle of each roller, which is the angle between the velocity υ_{pk} and the vector O_ky', k, represents the order of the wheels.

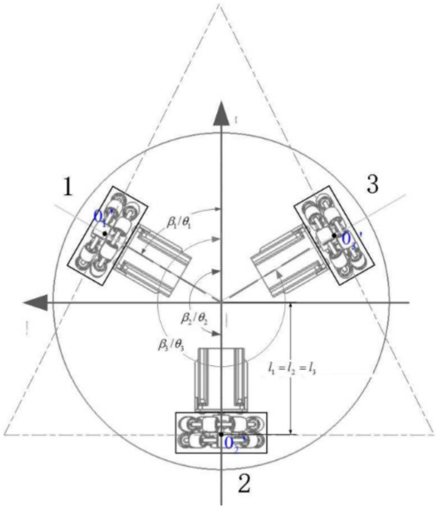


Figure 2. The Three-wheels mobile robot [28]

Usually, to create omnidirectional motion on the ground, omnidirectional mobile robots are usually equ–ipped with three or four mecanum wheels. A three-wheel mobile robot is utilized in this work in which three mecanum wheels are evenly distributed in a 360° circumference (Figure 2). The axes intersect at the frame's centroid, and the angles between two neigh–

boring wheels are the same (equal to 120^0). In Figure 3, the coordinate frame XOY is attached to the centroid of the robot platform in which the centreline OO'_k is collinear with the axis $O'x_i$. The angle between two axes OO'_k and OX is equal to the angle between O_kx and OX, denoted by β_k.

Let [v_{kx}, v_{ky}] ωkT denote the centroid velocity of the mecanum wheels relative to the frame XOY. The relationships between velocities in two frames $x'O_ky'$ and XOY are [28]:

$$\begin{bmatrix} v_{kx} \\ v_{ky} \end{bmatrix} = \begin{bmatrix} \cos\beta_k & -\sin\beta_k \\ \sin\beta_k & \cos\beta_k \end{bmatrix} \begin{bmatrix} v'_{kx} \\ v'_{ky} \end{bmatrix} \tag{2}$$

Substituteequation (1) into equation (2):

$$\begin{bmatrix} \dot{x}_g \\ \dot{y}_g \\ \dot{\theta}_g \end{bmatrix} = \begin{bmatrix} \cos\theta_g & -\sin\theta_g & 0 \\ \sin\theta_g & \cos\theta_g & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{3}$$

with $\delta_k = \beta_k$ - φ_k

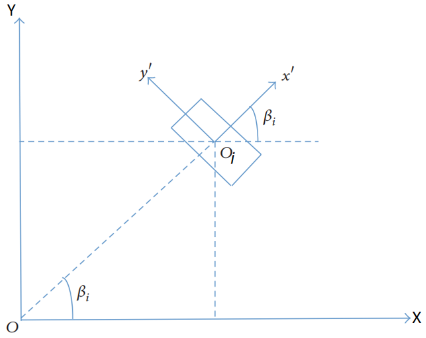


Figure 3. The Coordinate Frames Relationship

Denote $[v_x,\ v_y,\ \omega]^T$ is the velocity vector of the centroid platform relative to the ground plane. According to the geometric relationship [28]:

$$\begin{bmatrix} v_{kx} \\ v_{ky} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -d_{ky} \\ 0 & 1 & d_{kx} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{4}$$

where (d_{kx}; d_{ky})is the coordinate of the kth wheel's mass point in XOY coordinate system and can be expressed as:

$$\begin{aligned} d_{kx} &= d_i \cos\beta k \\ d_{ky} &= d_i \sin\beta k \end{aligned} \tag{5}$$

Substitute Equations (3) into (4), the inverse equation to determine the velocity of each mecanum wheel can be determined by the equation:

$$\begin{aligned} \begin{bmatrix} \omega_k \\ v_{pk} \end{bmatrix} &= \begin{bmatrix} r_w \sin\beta_k & -\sin\delta_k \\ r_w \sin\beta_k & \cos\delta_k \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & -d_{iy} \\ 0 & 1 & d_{ix} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \\ &= -\frac{1}{r_w \sin\varphi_k} \begin{bmatrix} \cos\delta_k & \sin\delta_k & -d_k \sin\varphi_k \\ r_w \cos\beta_k & -r_w \sin\beta_k & 0 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \end{aligned} \tag{6}$$

Define:

$$A = \begin{bmatrix} \dfrac{\cos\delta_1}{\sin\varphi_1} & \dfrac{\sin\delta_1}{\sin\varphi_1} & -d_1 \\ \dfrac{\cos\delta_2}{\sin\varphi_2} & \dfrac{\sin\delta 2}{\sin\varphi_2} & -d_2 \\ \dfrac{\cos\delta_3}{\sin\varphi_3} & \dfrac{\sin\delta_3}{\sin\varphi_3} & -d_3 \end{bmatrix} \tag{7}$$

The angular velocities of three mecanum wheels are calculated:

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \frac{1}{r_w} A \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{8}$$

In practice, the state of the mobile robot is deter–mined in a global coordinate system $x_gO_gy_g$. Let (x_g,y_g,θ_g) denote the robot's state in the global frame. The velocity relationships are defined as:

$$\begin{bmatrix} \dot{x}_g \\ \dot{y}_g \\ \dot{\theta}_g \end{bmatrix} = \begin{bmatrix} \cos\theta_g & -\sin\theta_g & 0 \\ \sin\theta_g & \cos\theta_g & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{9}$$

From Figure 2, the values of angles are $\beta_1 = \varphi_1 = 60°$, $\beta_2 = \varphi_2 = 180°$, $\beta_3 = \varphi_3 = 300°$, combining with (7), (8) and (9) to get equation (10):

$$S = r_w \begin{bmatrix} -\frac{\sqrt{3}}{3} c\theta_g - \frac{1}{3} s\theta_g & \frac{2}{3} s\theta_g & \frac{\sqrt{3}}{3} c\theta_g - \frac{1}{3} s\theta_g \\ \frac{1}{3} c\theta_g - \frac{\sqrt{3}}{3} s\theta_g & -\frac{2}{3} s\theta_g & \frac{1}{3} c\theta_g - \frac{\sqrt{3}}{3} s\theta_g \\ \frac{d}{3} & \frac{d}{3} & \frac{l}{3} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{10}$$

with

$$c\theta_g = \cos\theta_g, s\theta_g = \sin\theta_g, d_1 = d_2 = d_3 = d \tag{10}$$

Define $x = [x_g,y_g,\theta_g]^T$ as the state vector of the robot in frame $x_gO_gy_g$, $u = [\omega_1,\omega_2,\omega_3]^T$ as the control input vector, the state space equation of the robot is:

$$\dot{x}(t) = S(t), u(t) = f\left(x(t), u(t)\right) \tag{11}$$

where:

$$S = r_w \begin{bmatrix} -\frac{\sqrt{3}}{3} c\theta_g - \frac{1}{3} s\theta_g & \frac{2}{3} s\theta_g & \frac{\sqrt{3}}{3} c\theta_g - \frac{1}{3} s\theta_g \\ \frac{1}{3} c\theta_g - \frac{\sqrt{3}}{3} s\theta_g & -\frac{2}{3} s\theta_g & \frac{1}{3} c\theta_g - \frac{\sqrt{3}}{3} s\theta_g \\ \frac{d}{3} & \frac{d}{3} & \frac{l}{3} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{12}$$

3. MODEL PREDICTIVE COLLISION-FREE PATH FOLLOWING CONTROL FOR MOBILE ROBOT

3.1 Problem of following a parameter path

Consider a geometric reference curve given by [26]:

$$\mathbb{C} = \left\{ s \in [\hat{s}, 0] \subset \mathbb{R} \rightarrow \rho(s) \in \mathbb{R}^3 \right\} \tag{12}$$

where the s is a scalar value and called the path parameter, and $p(s)$ is a parametrization function:

$$p(s) = \left(\rho_x(s), \rho(s), a\tan\left(\frac{\partial \rho_y}{\partial \rho_x} \right) \right) \tag{13}$$

where $\rho_x(s)$, ρys are differentiable at least to the second order[20]. The path-following problem is defined as designing a controller so that the robot state $x(t)$ approaches the path $\mathcal{C}$and moves along $\mathcal{C}$[30]:

$$\lim_{t\to\infty} \left\| x(t) - p(s(t)) \right\| = 0 \tag{14}$$

In practice, exact path tracking is difficult due to system constraints. For the path-following problems, the mobile robot is not required to track the curve $\mathcal{C}$exactly. Instead, the control input and the timing $s(t)$ are to be chosen to steer the mobile robot as close as possible to the path $\mathcal{C}$ while still satisfying the state and control constraints. The timing law of $s(t)$ is not defined in advance; the controller determines it. The time evolution $t \rightarrow s(t)$ and the control input must be optimized to allow the robot to follow the curve and ensure the state and control constraints ($x \in \boldsymbol{X}$ and $u \in \boldsymbol{U}$**)**. Furthermore, to keep the robot moving on the path, it is required that the state of the path must satisfy the state constraints, $\mathcal{C} \subset \boldsymbol{X}$ for all $s \in [\hat{s}, 0]$.

Because the time evolution of s is was not specified prior, a virtual control input $\omega \in \mathbb{R}$ obtained in the controller is applied to control the path parameter s throughout a '*timing law'*, and the path parameter s is considered as a virtual state of the system. So, the controller is extended with an extra degree of freedom. Although complex timing laws can be considered, a simple integrator is used in this work:

$$\dot{s}(t) = \omega(t), s(0) = s_0 \tag{15}$$

where $s_0 \in [\hat{s}, 0]$ is the initial state of the path, it must be predefined (assumed to be known in advance). The virtual control input ω is assumed to be piecewise continuous and bounded ω(t) $\in O \subset \mathbb{R}_{>0}$. The positive value of ω is to ensure that the robot does not move backward along the path.

Define the extended state $\chi = [x^T, s]^T \in \mathbb{R}^4$ that contains the robot state and the virtual state, the extended input $\eta = [u^T, \omega]^T \in \mathbb{R}^4$ consists of the control input and the virtual control variable. The constraint set of the augmented state χ is $\Omega = \boldsymbol{\chi} \times [\hat{s}, 0]$. The input constraint of η is $V = \boldsymbol{U} \times O$. The path-following prob–lem is analyzed via the following augmented system:

$$\chi(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{s}(t) \end{bmatrix} = \begin{bmatrix} f(x(t), u(t)) \\ \omega(t) \end{bmatrix} = f_\chi(\chi(t), \eta(t)) \tag{16}$$

The output of the system is:

$$\begin{bmatrix} e_{pf}(t) \\ s \end{bmatrix} = \begin{bmatrix} x(t) - p(s(t)) \\ s(t) \end{bmatrix}$$

It can be seen that the output path-following requires that the error e_{pf} converges to zero while the path parameter s converges to the origin. Where epft the path following error, which is the difference between the state of the robot and the path state. So, the path-following problem is reduced to consider as the point-stabilization problem and can be solved by using a Nonlinear Model Predictive Control.

3.2 Model Predictive Path-Following Control (MPFC)

Assuming that sampled-data state information is available to the controller, the output path-following problem is solved using a continuous-time sampled-data nonlinear model predictive control (NMPC) scheme, called model predictive path following control (MPFC). The solution of MPFC can be obtained by repeatedly solving an optimal control problem (OCP). Let x and u denote the predicted states and outputs to distinguish the actual value from the predicted. At each sampling instance $t_k = k\tau$, with $k = 0,1,2\ldots$; τ is the sampling time, the cost function to be minimized in the OCP is [27]:

$$J(\bar{\chi}, \bar{\eta}) = \int_{t_k}^{t_k+T} F(\bar{\chi}, \bar{\eta}) d\tau + E(\bar{\chi}(t_k + T)) \tag{17}$$

with $T > 0$ is the prediction horizon. The OCP solved repetitively in the controller is [30]:

$$\min_{\eta \in V} J(\bar{\chi}, \bar{\eta}) \tag{18}$$

subject to the constraints:

$$\begin{aligned} &\dot{\bar{\chi}}(\tau) = f_\chi(\bar{\chi}(\tau), \bar{\eta}(\tau)) \\ &\bar{\chi}(t_k) = \chi(t_k) \\ &\bar{\chi} \in \Omega, \bar{\eta} \in V \end{aligned} \tag{19}$$

The terms of the cost function are defined as [25]:

$$\begin{aligned} F(\bar{\chi}, \bar{\eta}) &= \frac{1}{2}\bar{e}_{pf}(\tau)\mathbf{Q}\bar{e}_{pf}^T(\tau) + \frac{1}{2}\dot{\bar{e}}_{pf}(\tau)\mathbf{Q_d}\dot{\bar{e}}_{pf}^T(\tau) \\ &+ \frac{1}{2}\bar{u}(\tau)\mathbf{R}\bar{u}^T(\tau) + \frac{1}{2}qs(\tau)^2 + \frac{1}{2}r\omega(\tau)^2 \\ E(\bar{\chi}) &= \frac{1}{2}\bar{e}_{pf}(\tau)\mathbf{R_e}\bar{e}_{pf}^T(\tau) + \frac{1}{2}r_s s(\tau)^2 \end{aligned} \tag{20}$$

with $q \geq 0$, $r \geq 0$, $r_s \geq 0$, and $\boldsymbol{Q}, \boldsymbol{Q_d}, \boldsymbol{R}$ Re are the positive semidefinite diagonal matrices.

The OCP is solved by applying the direct simul–taneous approach with the support of CasADI software. The OCP is reformulated as Nonlinear programming (NLP) using the 4th order Runge-Kutta (RK4). Denote $\chi_k = \chi(t_k)$, $\eta_k = \eta(t_k)$, The state of the system is solved by RK4 equation [31]:

$$\chi_{k+1} = \chi + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4) = G(\chi, \eta) \tag{21}$$

with

$$\begin{aligned} &K_1 = f_\chi(\bar{\chi}_k, \bar{\eta}_k) \\ &K_2 = f_\chi(\bar{\chi}_k + K_1/2, \bar{\eta}_k) \end{aligned}$$

$$
\begin{aligned}
K_3 &= f_\chi(\bar{\chi}_k + K_2/2, \bar{\eta}_k) \\
K_4 &= f_\chi(\bar{\chi}_k + K_3, \bar{\eta}_k)
\end{aligned}
\tag{22}
$$

The cost function $J(\bar{\chi},\bar{\eta})$ is discretized as [8]:

$$
J(\bar{\chi},\bar{\eta}) = \Phi(q) = \sum_{j=k}^{k+N_T} \tau F(\bar{\chi},\bar{\eta}) + E\left(\bar{\chi}_{k+N_T}\right) \tag{23}
$$

where $q = \left(\bar{\chi}_k^T, \bar{\eta}_k^T, \ldots, \bar{\chi}_{k+N_T}^T, \bar{\eta}_{k+N_T}^T\right)^T, N_T = T/\tau$ is the prediction horizon length.

The resulting NLP is as follows [8]:

$$
\min_q \Phi(q) \tag{24}
$$

subject to:

$$
\begin{aligned}
& f_q(q) = 0 \\
& q_{2k-1} \in \Omega, q_{2k} \in V, k = 1 \div N_T
\end{aligned}
\tag{25}
$$

With

$$
f_q(q) = \begin{bmatrix} \bar{\chi}_k - \chi(t_k) \\ \bar{\chi}_{k+1} - G(\bar{\chi}_k, \bar{\eta}_k) \\ \vdots \\ \bar{\chi}_{k+N_T} - G\left(\bar{\chi}_{k+N_T-1}, \bar{\eta}_{k+N_T-1}\right) \end{bmatrix} \tag{26}
$$

3.3 Obstacle avoidance

To avoid collision with obstacles when following the path, the cost function is extended in order to push the mobile robot away from obstacles. At the same time, the mobile robot is controlled to follow the path as closely as possible. The distances between the centroid of the robot and the obstacles are calculated:

$$
d_i^2 = \left\| x_{0i} - x \right\|^2 \tag{27}
$$

where x0i $\in \square^3$ is the coordinate of the ith obstacle point. Suppose that the positions of obstacles are known in advance and are available to the controller at each iteration. The cost function is augmented by a term that penalizes the squared distances between the mobile robot and the obstacle centroid, as used by Angelika in [25]:

$$
\begin{aligned}
F_0(x) &= \sum_{i=1}^{N_0} \frac{m_i}{d_i(x)^2 - d_{\min}^2} \\
E_0(x) &= c_0 F_0(x)
\end{aligned}
\tag{28}
$$

with $m_i \geq 0$, $c_0 \geq 0$ are constant coefficients, d_{min} is the minimum distance between the robot center and obstacles, and N_0 is the number of obstacles. It can be seen that if $d_i^2 \to d_{\min}^2$, ratio $m_i / \left(d_i^2 \to d_{\min}^2\right) \to \infty$. So, adding F_0 to the objective function helps the mobile robot move away from the obstacles. However, there is no guarantee that the mobile robot does not collide with the obstacles. For this reason, the following constraints must be considered:

$$
d_i\left(x(k)\right)^2 - d_{\min}^2, k = 1 \div N_T, i = 1 \div N_0 \tag{29}
$$

These constraints ensure that the distances between the centroid of the obstacles centroid and the mobile robot are always to be greater safety distance.

4. SIMULATION RESULTS

In this part, simulations are conducted to confirm the correctness of the kinematic equation of the OMR and the effectiveness of the model's predictive collision-free path following the control algorithm. Two reference paths chosen for simulation are:

(1) a circle of radius 0.8m, the centroid at (0.5 m; 1 m), no obstacle, the initial coordinate of the mobile robot is (1.3m; 1m; π/4 rad) (robot do not lie on the reference path at the beginning).

(2) a sine path, $p_x = s/2$, $p_y = \sin(s)$, $s \in [-4\pi, 0]$, two obstacles at $x_{01} = (-4,-1)$ and $x_{02} = (-2.6, -1)$. The radius of obstacles is 0.1 m. So, the minimum distance between the robot and obstacles is $d_{min} = 0.3$m. the initial position of the mobile robot is (-6.48m; 0.1m; 1.1 rad).

In simulation 1, all coefficients for the MPFC are as follows.

Q = diagonal(4000, 4000, 800)
Q_p = diagonal (40,40,10)
R = diagonal(0.01, 0.01, 0.01)
R_e = diagonal ($4{\cdot}10^5$, $4{\cdot}10^5$, $8{\cdot}10^4$)
$Q = 0.1$, $r = 0$, $r_s = 1$
$N_T = 10$, $T = 0.05$s

To simulate the real state of the mobile robot, noise with a mean of zero and a variance of 5mm is included in the output state. The results are shown in Figure 4. Plot a of Figure 4 is the comparison of the reference and real path of the robot. The reference path is represented by the green dashed line, and the red solid line marks the movement of the mobile robot. Plot b of Figure 4 shows the error on the x and y-axes and the direction angle error. Plot c of Figure 4 shows the control input applied to the mobile robot. From Figure 4.a, it can be seen that robot is controlled from the initial position to approach the reference path and follow exactly the reference. The maximum error at the initial stage is (15mm, 200 mm, 0.084 rad). When the robot has approached the reference, the maximum tracking error is about 4mm. The control inputs are bound in the constraints with a maximum 1 rad/s.It can be seen that the robot tries to move to the first point of the path before moving along the path. To ensure small tracking errors, the coefficients of the matrix Q have very large values. In contrast, the value of r_S is very small. In this case, the robot needs to track the path accurately rather than move to the target or overcome an obstacle.

The parameters for the MPFC in Simulation Case 2 are chosen:

Q = diagonal(40, 40, 40)
Q_p = diagonal (40,40,10)
R = diagonal(0.01, 0.01, 0.01)
R_e = diagonal ($4{\cdot}10^5$, $4{\cdot}10^5$, $8{\cdot}10^4$)
$Q = 2$, $r = 0$, $r_s = 20$
$N_T = 10$, $T = 0.05$s

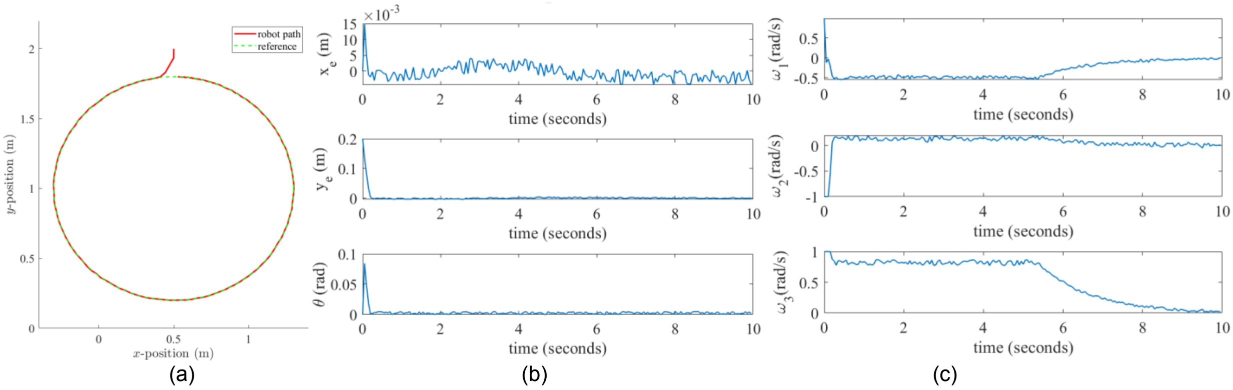


Figure 4. The simulation result of the circular path: (a) robot path and reference in x-y plane, (b) the pose error, (c) the angular velocity of wheels

In this case, the robot is controlled to follow the path and avoid obstacles simultaneously. Obstacle avoidance task has a higher priority to follow the path. So, the weight matrix must be changed to achieve the targets. If the parameters are kept the same as in the previous task with high coefficients ofpath-following error, the robot tries to follow the path exactly. So, it does not leave the path to pass through the obstacle and halts in front of the obstacle, as shown in Figure 5.

For this reason, the coefficients of matrix Q and R_e in the cost function are reduced, and the coefficients q and r in the cost of the path parameter are increased. The results are shown in Figure 6. The MPFC can synchronize the path following the task and avoiding a collision. However, in the early stages, the robot takes longer to approach the reference path. Then, the robot follows the path exactly until it encounters an obstacle. The MPFC steers the robot as close to the path as possible while avoiding collisions with obstacles. In Figure 6.b, it can be seen that the maximum position error is equal to the distance d_{min}. Due to using the cost $F_0(x)$, the trajectory of the robot is smooth. If only using the constraints about the minimum distance to the obstacle, the robot's path would be the same as the outline of the obstacles, and it may not be smooth.

The parameters in test case 2 are used to simulate the path defined by the equation:

$$p_x = 1.8\sin(s),\ p_y = 1.2\sin(2s)$$

and avoid colliding with three obstacles located at positions (0,0), (-1.9, 0.5), and (1.4, -1.2), respectively. The results in Figure 7 show that the robot can track the path and avoid colliding with obstacles. The selected set of parameters can be used for other path types.

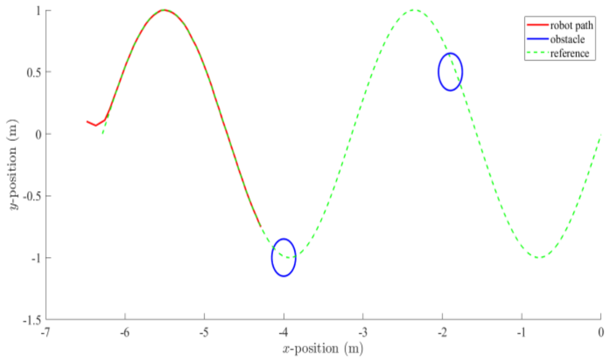


Figure 5. The result with high coefficients of the path following error.

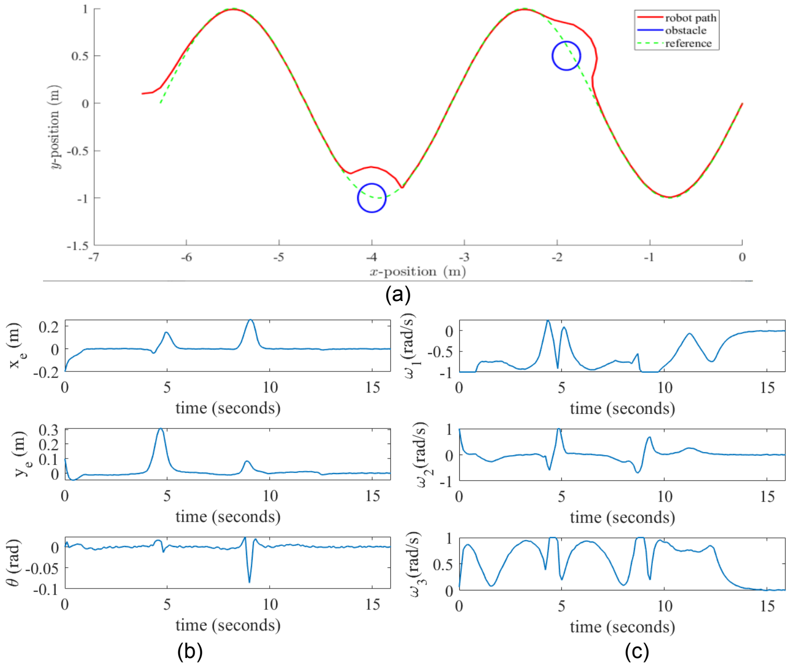
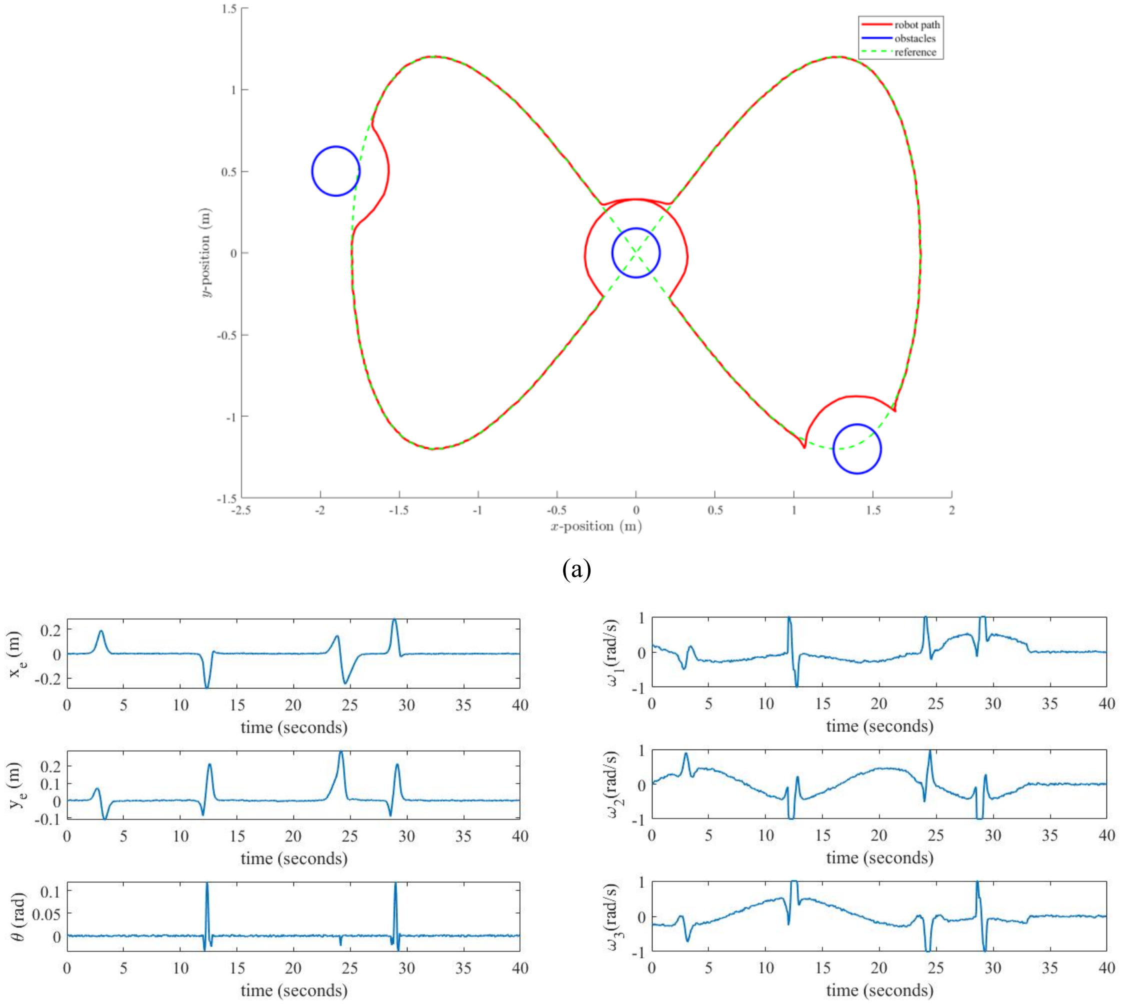


Figure 6. The result of the simulation of the sine path: (a) robot path and reference in the xy plane, (b) pose error, and(c) angular velocity of the wheel



(a)

(b)

(c)

Figure 7. The result of the simulation of the eight-shaped curve: (a) robot path and reference in the xy plane, (b) pose error, and(c) angular velocity of the wheel

From the simulation results, it can be seen that the choice of parameters in the cost function is very important. Depending on which tasks need to be prio–ritized, we choose the appropriate parameters. If accu–racy is the preferred metric, then the coefficients related to tracking error (matrix Q and Q_P) must be large. The coefficients of the matrices R and R_S or the values of q and r will be related to the speed of travel. If the matrices R and R_S have elements with large values, then the control signal u will have a small value, so the robot will move more slowly. If collision avoidance is also consi–dered, then the elements of the matrix Q and Q_P must have small values and the coefficients q and r must have large values. When encountering an obstacle, the robot can overcome it without considering the tracking error.

5. CONCLUSION

This work has presented a model predictive path for the following controller for an omnidirectional mobile robot. The kinematic equation of the OMR is obtained and used to estimate the next state of the robot in the controller. The extended system with the output is the path-following error, the virtual state is established, and the path-following problem is reformulated into the regulation problem. Obstacle avoidance is obtained by adding a penalty into the cost function that depends on the gap between the mobile robot and the obstacles.

The simulations are conducted, and the results reveal that the proposed controller can steer the robot to follow the path and avoid colliding with obstacles simulta–neously. If only a reference path is required to follow exactly, the parameters related to the path following the error in the cost function are chosen with high values. If obstacles are considered, these values should be reduced so that the controller can synchronize the path-following task and collision avoidance. In the test case of the path-following task, the maximum tracking error is only 4mm, and the robot quickly approaches the path (in 0.1s). If obstacle avoidance is considered, tracking error is not a priority metric, so the robot slowly approaches the path. It takes about 1s for the robot to approach the path. The tracking error is large at the beginning of the path.

In future work, we will develop a real robot proto–type to conduct the experiment test. The real robot will be equipped with sensors to perceive the surrounding environment and measure the distance to obstacles. Robots can be used for purposes such as transporting goods in warehouses; cleaning and disinfection in hos–pitals or healthcare centers; delivering food, drinks, and goods to guests in restaurants or in communal areas…

ACKNOWLEDGMENT

We acknowledge Ho Chi Minh City University of tech–nology (HCMUT), VNU-HCM for supporting this study.

REFERENCES

[1] Saadatzi, M.N., Abubakar, S., Das, S.K., Saadatzi, M.H., Popa, D.: Neuroadaptive Controller for Physical Interaction With an Omni-Directional Mobile Nurse Assistant Robot, In Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Virtual, Online, 17–19 August 2020.

[2] Levratti, A., De Vuono, A., Fantuzzi, C., Secchi, C.: TIREBOT: A Novel Tire Workshop Assistant Robot, In Proceedings of the International Conference on Advanced Intelligent Mechatronics (AIM), Banff, AB, Canada, 12–15 July 2016.

[3] Tagliavini, L., Botta, A., Cavallone, P., Carbonari, L., Quaglia, G.: On the Suspension Design of Paquitop, a Novel Service Robot for Home Assistance Applications, Machines, Vol.9, No.3, 2021.

[4] Bogue, R.: Domestic Robots: Has their time finally come?, Ind. Robot. Int. J., Vol. 44, pp. 129–136, 2017.

[5] Mohd Salih, J.E., Rizon, M., Yaacob, S., Adom, A.H.: Mamat, M.R. Designing omni-directional mobile robot with mecanum wheel, Am. J. Appl. Sci., Vol. 3, pp. 1831–1835, 2006.

[6] Asama, H., Sato, M., Bogoni, L., Kaetsu, H.: Development of an omni-directional mobile robot with 3 DOF decoupling drive mechanism, In Proceedings of the IEEE International Conference on Robotics and Automation, Nagoya, Japan, 21–27 May 1995; Volume 2, pp. 1925–1930.

[7] Dario, A., Khurshid, A.: Intelligent Energy Management for Mobile Manipulators Using Machine Learning, FME Transactions, Vol. 50, pp. 752-761, 2022. ø

[8] Arbo, M.H, Grøtli, E. I., Gravdahl, J. T.: On model predictive path following and trajectory tracking for industrial robots, 2017 13th IEEE Conference on Automation Science and Engineering (CASE), pp. 100-105, 2017.

[9] Faulwasser, T. Weber, T., Zometa, P. and Findeisen, R.: Implementation of Nonlinear Model Predictive Path-Following Control for an Industrial Robot, IEEE Transactions on Control Systems Technology, pp. 1–7, 2016.

[10] Altafini, C.: Following a path of varying curvature as an output regulation problem, IEEE Trans. on Automatic Control, Vol. 47, No. 9, pp. 1551-1556, Sep. 2002.

[11] Al-Hiddabi, S.A. and McClamroch, N.H.: Tracking and maneuver regulation control for nonlinear non-minimum phase systems: application to flight control, IEEE Trans. on Control Systems Technology, Vol. 10, No. 6, pp. 780-792, 2002.

[12] Kalmár-Nagy, T., D'Andrea, R., Ganguly, P.: Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle, robot. Auton. Syst., Vol. 46, pp. 47–64, 2004.

[13] Huang, H.C., Tsai, C.C.: Adaptive Trajectory Tracking and Stabilization for Omnidirectional Mobile Robot with Dynamic Effect and Uncertainties, IFAC Proc. Vol. 41, pp. 5383–5388, 2008.

[14] Kanjanawanishkul, K., Zell, A.: Path following for an omnidirectional mobile robot based on Model predictive control, In Proceedings of the IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 3341–3346.

[15] Nascimento, T., Dorea, C.E.T., Gon' calves, L.: Nonholonomic ˙ mobile robots' trajectory tracking model predictive control: a survey. Robotica, Vol. 36, No. 5, 2018.

[16] Nascimento, T.P., Saska, M.: Position and attitude control of multi-rotor aerial vehicles: a survey. Ann. Rev. Control, Vol. 48, pp. 129–146, 2019.

[17] Yu, S., Li, X., Chen, H., Allgower, F.: Nonlinear model predictive control for path following problems. Int. J. Robust Nonlinear Control, Vol. 25, No. 8, pp. 1168–1182, 2015.

[18] Faulwasser, T. and Findeisen, R.: Nonlinear Model predictive control for constrained output path following. IEEE Trans. Automat. Control, Vol. 61, No. 4, pp. 1026-1039, 2016.

[19] Alessandretti, A., Aguiar, A.P., Jones, C.N.: Trajectory-tracking and path-following controllers for constrained underactuated vehicles using model predictive control. In: 2013 European Control Conference (ECC), pp. 1371–1376. IEEE (2013).

[20] Bock, M., Kugi, A.: Real-time nonlinear model predictive path-following control of a laboratory tower crane. IEEE Trans. Control Syst. Technol. Vol. 22, No. 4, pp. 1461–1473, 2013.

[21] Matschek, J., Bathge, T., Faulwasser, T., Findeisen, R.: Nonlinear Predictive Control for Trajectory Tracking and Path Following: An Introduction and Perspective, Springer International Publishing, Cham, pp. 169–198, 2019.

[22] Ammar, A., Abdelhakim, C., Halim, M., Yazid L.D.L.: Nonlinear Model Predictive Control of a Class of Continuum Robots Using Kinematic and Dynamic Models, FME Transactions, Vol. 50, pp. 339-350, 2022.

[23] Fawaz, F.A.B., Salwan, O.W.K., Hasan, H.A.: Adaptive Model Predictive Control for a Magnetic Suspension System under Initial Position Dispersions and Voltage Disturbances, FME Transactions, Vol. 50, pp. 211-222, 2022.

[24] Funke, J., Brown, M., Erlien, S.M., Gerdes, J.C.: Collision avoidance and stabilization for autono–mous vehicles in emergency scenarios, IEEE Trans. Control Syst. Technol. Vol. 25, No.4, pp. 1204–1216, 2017.

[25] Zube, A.: Cartesian nonlinear model predictive control of redundant manipulators considering obstacles. In: 2015 IEEE International Conference on Industrial Technology (ICIT), pp. 137–142, 2015.

[26] Rubagotti, M., Taunyazov, T., Omarali, B., Shintemirov, A.: Semi-autonomous robot teleoperation with obstacle avoidance via Model predictive control. IEEE Robot. Autom. Lett. Vol. 4, No. 3, pp. 2746–2753, 2019.

[27] Sánchez, I., D'Jorge, A., Raffo, G.V. et al.: Nonlinear Model Predictive Path Following Controller with Obstacle Avoidance, J Intell Robot Syst, Vol. 102, No. 16, 2021.

[28] Wang C.C: et al.: Trajectory Tracking of an Omni-Directional Wheeled Mobile Robot Using a Model Predictive Control Strategy, Appl. Sci., Vol. 8, 2018.

[29] Yu S., Li X., Chen H., Allgöwer F.: Nonlinear model predictive control for path following problems, International Journal of Robust and Nonlinear Control Vol. 25, No. 8, pp. 1169-1182, 2013

[30] Faulwasser T., Kern B., Findeisen R.: Model predictive path-following for constrained nonlinear systems, Proceedings of the IEEE Conference on Decision and Control, pp. 8642-8647, 2009.

[31] Egeland O. and Gravdahl, J.T.: Modeling and Simulation for Automatic Control. Trondheim: Marine Cybernetics, 2003.

NOMENCLATURE

x	The state vector of the robot
$p(s)$	The parametrization function
s	The path parameter
$\boldsymbol{X}$	The state constraint
$\boldsymbol{U}$	The control constraint
ω	The virtual control input
χ	The extended state
η	The extended input
Ω	The constraint set of the augmented state
V	The augmented control constraint
e_{pf}	The path follows error
J	The cost function
F	The penalty term of the error and input
E	The terminal penalty term
K_i	Coefficients of RK4 function
N_T	The prediction horizon length
x_{0i}	The centroid of the obstacles
d_i	The centroid distance
F_0	The term that penalizes the centroid distance
E_0	The terminal term that penalizes the centroid distance
$\boldsymbol{Q},\boldsymbol{Qp}$	Positive semidefinite diagonal matrix for error
$\boldsymbol{R},\boldsymbol{R_e}$	Positive semidefinite diagonal matrix for control
q,r,r_s	Positive semidefinite coefficients
T	Sample time

Abbreviations

c	cosine
s	sine
MPC	Model Predictive Control
NMPC	Nonlinear Model Predictive Control
OCP	Optimal Control Problem
MPFC	Model Predictive Path Following Control
MPCPFC	Model Predictive Collision-Free Path Following Control
OMR	Omnidirectional Mobile Robot
NLP	Nonlinear programming
RK4	the 4th order Runge-Kutta
PLDI	Polytopic Linear Differential Inclusion

МОДЕЛ ПРЕДИКТИВНЕ КОНТРОЛЕ ПУТА БЕЗ СУДАРА ЗА НЕХОЛОНОМСКЕ МОБИЛНЕ РОБОТЕ

Т.Т. Хиеп, В.Д. Конг, Л.Х. Фуонг

У овом истраживању развијен је модел предиктивног контролора пута без судара који је примењен за омнидирекциони мобилни робот (ОМР). Мобилни робот се контролише тако да прати референтну путању док избегава судар са препрекама. Проблем праћења путање је преформулисан у проблем регулације проширеног постројења увођењем виртуелног степена слободе, параметра путање геометријске референтне криве. Затим се примењује моделски предиктивни контролер (МПЦ) за управљање мобилним роботом. Функција трошкова оптимизације се успоставља из разлике између стања робота и путање параметра. Решење МПЦ-а се може добити узастопним решавањем проблема оптималне контроле (ОЦП) да би се функција трошкова оптимизације свела на минималну вредност, чинећи стање робота што ближе стању путање. Избегавање препрека се разматра додавањем појмова као функције јаза између мобилног робота и објеката испред робота. Ограничења на стања и улазе система се такође лако разматрају у проблему оптималног управљања МПЦ. Ово чини да контролни улази не прелазе дозвољене границе робота. Симулације се спроводе да би се открила ефикасност контролера и показало како одабрати праве параметре за синхронизацију задатака праћења путање и избегавања препрека.