MPC-Based Path Following Control of an Omnidirectional Mobile Robot with Consideration of Robot Constraints

Kiattisin KANJANAWANISHKUL

Mechatronics Research Unit, Faculty of Engineering, Mahasarakham University, Kantarawichai District, Khamriang Sub-District, Maha Sarakham, 44150, Thailand

kiattisin_k@hotmail.com

DOI: 10.15598/aeee.v13i1.1228

Abstract. *In this paper, the path following problem of an omnidirectional mobile robot (OMR) has been studied. Unlike nonholonomic mobile robots, translational and rotational movements of OMRs can be controlled simultaneously and independently. However the constraints of translational and rotational velocities are coupled through the OMR's orientation angle. Therefore, a combination of a virtual-vehicle concept and a model predictive control (MPC) strategy is proposed in this work to handle both robot constraints and the path following problem. Our proposed control scheme allows the OMR to follow the reference path successfully and safely, as illustrated in simulation experiments. The forward velocity is close to the desired one and the desired orientation angle is achieved at a given point on the path, while the robot's wheel velocities are maintained within boundaries.*

Keywords

Model predictive control, omnidirectional mobile robots, path following control, robot constraints, virtual vehicle.

1. Introduction

In this paper, we address the path following problem where a mobile robot is forced to follow a desired spatial path without consideration in temporal specifications [1]. The solution of this problem offers several remarkable advantages over trajectory tracking [2]. For example, the time dependence of the trajectory tracking problem is eliminated, convergence to the path is achieved smoothly, and saturation of control signals is less likely reached. Original research on this area can be found in [3].

In general, the path following controller determines the robot's moving direction that can bring it to the path, while the robot's forward velocity tracks a desired velocity profile. In the literature, there are two control strategies for path parameterization [4], i.e., the Frenet frame with an orthogonal projection of a robot on the given path [3], [5], [6], [7] and the Frenet frame with a non-orthogonal projection of a robot on the given path [1], [8], [9], [10], [11], [12]. In the first method, the position of the virtual vehicle to be followed by a real one is defined by the orthogonal projection of the robot on the path. However, this method can be used only when the initial position of the robot is near the path. In the other method which can overcome the initial condition problem of the first method, a desired geometric path is parametrized by the curvilinear abscissa $s \in \mathbb{R}$ and the velocity of the virtual vehicle ($\dot{s}(t)$) can be controlled explicitly.

Although the path following problem has been solved for different types of robots over the past decade, omnidirectional mobile robots (OMRs) have been considered in this work since they have some distinct advantages over nonholonomic mobile robots. They can move instantly in any direction without reorientation [13]. They become increasingly popular in mobile robot applications as seen from a large number of publications dealing with OMRs. However, research study on path following control of OMRs is still rare. Some related work is as follows: Vazquez et al. [14] adapted computed torque control usually used in robot manipulators to the path following control problem of the OMR. Conceicao et al. [6] proposed a nonlinear model predictive control for an OMR. The cost function includes the robot pose error and the control effort. They

followed the concept that the position of the virtual vehicle is defined by the orthogonal projection of the robot on the path. Huang and Tsai [15] applied an adaptive robust control method to the path following problem for an OMR with consideration of actuators' uncertainties in polar coordinates. Kanjanawanishkul and Zell [16] used model predictive control to generate an optimal velocity of the virtual vehicle. Recently, Oftadeh et al. [11] proposed a new solution to the path following problem where speed of the robot can be determined analytically to keep the steering and driving velocities of the wheels under predetermined values.

In this work, a model predictive control (MPC) approach for solving the path following problem of an OMR is designed. The proposed idea is that the velocity of the virtual vehicle can be controlled explicitly through the MPC scheme. Although the translational velocity and the rotational velocity of an OMR can be controlled separately, their boundaries are coupled via the robot's orientation angle. Thus, both velocities are included into the objective function, while the robot's wheel velocity constraints and other robot constraints are satisfied in the constrained minimization problem of the MPC scheme that is online solved at each sampling time.

The rest of the paper is structured as follows: in Section 2., the robot kinematics is derived. The path following control problem is described and the path following controller based on the MPC strategy with consideration of robot constraints are developed in Section 3. Then, simulation experiments are conducted in Section 4. to show the effectiveness of our proposed controller. Finally, our conclusions are given in Section 5.

2. Kinematic Modeling

From Fig. 1, the kinematic model of an OMR can be given by:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \mathbf{R}_z(\theta)\begin{bmatrix}u\\ v\\ \omega\end{bmatrix} = \begin{bmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}\cdot\begin{bmatrix}u\\ v\\ \omega\end{bmatrix}, \quad (1)$$

where $\mathbf{R}_z(\theta)$ is the rotation matrix that transforms the robot velocities with respect to the body frame (X_m, Y_m) to the world frame (X_w, Y_w). $\vec{x}(t) = [x,\ y,\ \theta]^T$ is the state vector of the robot in the world frame and θ denotes the angle of the robot's orientation, u and v are the translational velocities observed in the body frame, while ω is the rotational velocity.

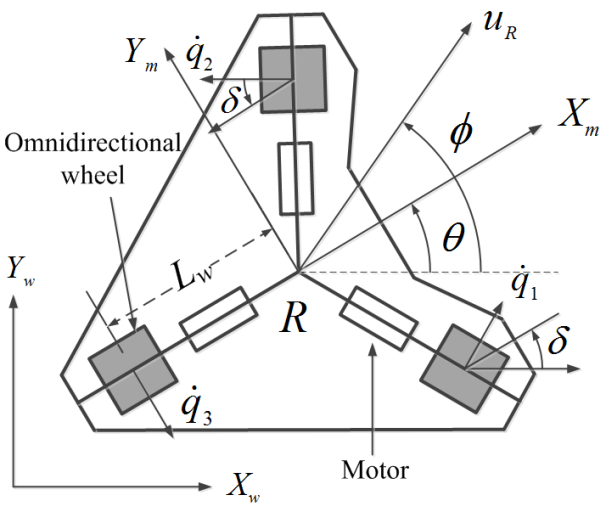

Fig. 1: Coordinate frames of an OMR.

Equation (1) can be rewritten by decoupling translation and rotation as follows:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\phi}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}u_R\cos\phi\\ u_R\sin\phi\\ \psi\\ \omega\end{bmatrix}, \quad (2)$$

where ϕ denotes the angle of the robot's moving direction in the world frame and u_R is the forward linear velocity of the robot. Furthermore, the robot translational velocities can be determined by:

$$\begin{bmatrix}u\\ v\end{bmatrix} = \begin{bmatrix}u_R\cos(\phi-\theta)\\ u_R\sin(\phi-\theta)\end{bmatrix}. \quad (3)$$

When the wheel velocities are taken into account, the following lower level kinematic model with respect to the body frame is given:

$$\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\\ \dot{q}_3\end{bmatrix} = \begin{bmatrix}\cos\delta & \sin\delta & L_w\\ -\cos\delta & \sin\delta & L_w\\ 0 & -1 & L_w\end{bmatrix}\cdot\begin{bmatrix}u\\ v\\ \omega\end{bmatrix}, \quad (4)$$

where $\vec{q}(t) = [\dot{q}_1,\ \dot{q}_2,\ \dot{q}_3]^T$ is the vector of linear velocities of the wheel. It is equal to the wheel's radius multiplied by the wheel's angular velocity. L_w denotes the distance from the platform center to the wheel center (see Fig. 1) to the center of wheel. δ refers to the angle of the wheel orientation in the body frame. Since the translational and rotational velocities of an OMR can be separately controlled, the wheel velocities can be divided into two components:

$$\vec{q}_t = \begin{bmatrix}\cos\delta & \sin\delta\\ -\cos\delta & \sin\delta\\ 0 & -1\end{bmatrix}\cdot\begin{bmatrix}u\\ v\end{bmatrix}, \quad \vec{q}_r = \begin{bmatrix}L_w\\ L_w\\ L_w\end{bmatrix}\omega, \quad (5)$$

where $\vec{q}_t$ and $\vec{q}_r$ are translational and rotational components for each wheel velocity, respectively.

As the motor's voltage and current are limited, the summation of these two components is bounded by $\dot{q}_{\max}$, i.e., $\|\vec{q}_t + \vec{q}_r\|_\infty \leq \dot{q}_{\max}$.

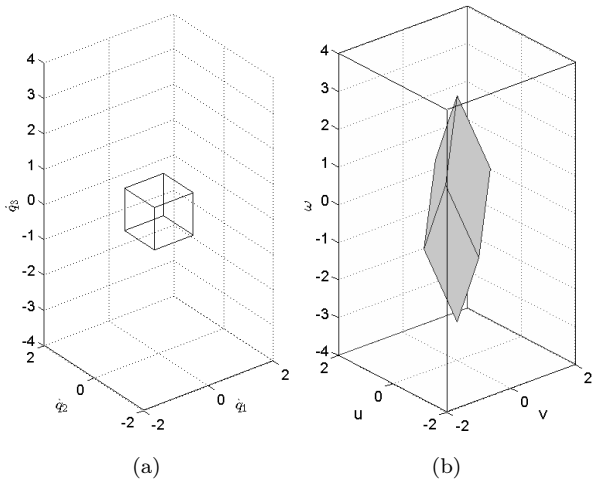

Fig. 2: The relationship between wheel velocities and robot velocities with respect to the body frame: (a) the cube defined by $\mathcal{Q}(t) = \{\vec{q}(t) \mid |\dot{q}_i(t)| \leq \dot{q}_{i,\max}\}$. (b) the tilted cuboid $\mathbf{P}(0)\mathcal{Q}(t)$.

The relationship between wheel velocities and robot velocities with respect to the world frame can be calculated by substituting Eq. (4) into Eq. (1), resulting in:

$$\dot{\mathbf{x}}(t) = \mathbf{R}_z(\theta)\mathbf{P}(0)\vec{q}(t) \ , \tag{6}$$

where:

$$\mathbf{P}(0) = \begin{bmatrix} \sqrt{3}/3 & -\sqrt{3}/3 & 0 \\ 1/3 & 1/3 & -2/3 \\ 1/(3L_w) & 1/(3L_w) & 1/(3L_w) \end{bmatrix}, \tag{7}$$

with $\delta = \frac{\pi}{6}$ rad. As an example, the linear transformation $\mathbf{P}(0)$ maps the cube $\mathcal{Q}(t) = \{\vec{q}(t) \mid |\dot{q}_i(t)| \leq \dot{q}_{\max}\}$, where $i = 1, 2, 3$, (see Fig. 2(a)) into the tilted cuboid $\mathbf{P}(0)\mathcal{Q}(t)$ (see Fig. 2(b)) with $L_w = 0.2$ m and $|\dot{q}_i(t)| \leq 0.6$ m·s^{-1}. The transformation $\mathbf{R}_z(\theta)$ then rotates this cuboid about the ω axis. As seen in Fig. 2(b), the boundary of translational and rotational velocities are coupled via θ. Furthermore, for a given θ, the translational velocity may decrease in order that the allowable rotational velocity can increase, while all wheel velocities are kept within the cube $\mathcal{Q}(t)$. This concept is useful when the large rotational velocity is required, e.g., during converging the robot to the path or moving along a sharp turning. One simple solution is to scale the translational and rotational velocities, as proposed in [17]. However, this solution does not utilize the full capacity of the wheel's maximum velocity. In this work, an MPC-based method with consideration of robot constraints is proposed to ensure that the path following control is attained and the robot constraints are within boundaries.

3. The Path Following Control Problem and Controller Design

The Frenet frame plays the role of the body frame of the virtual vehicle moving long the reference path. In general, the forward velocity u_R tracks a desired velocity profile, while the velocity of the virtual vehicle $\dot{s}$ converges to u_R. However, to utilize the full capacity of wheel velocities and to keep the wheel velocities within boundaries, the forward velocity cannot be fixed. Thus, an acceleration control input $a_m = \dot{u}_m$, where u_m is the robot's actual forward velocity, is introduced. Then, we obtain $\eta_e = u_m - u_R$ and $\dot{\eta}_e = a_m - \dot{u}_R$. From Fig. 3, the error state vector $\vec{x}_e = [x_e,\ y_e,\ \phi_e,\ \theta_e,\ \eta_e]^T$ between the state vector of the robot and that of the virtual vehicle can be expressed in the Frenet frame as follows:

$$\begin{bmatrix} x_e \\ y_e \\ \phi_e \\ \theta_e \\ \eta_e \end{bmatrix} = \tag{8}$$

$$= \begin{bmatrix} \cos\phi_d & \sin\phi_d & 0 & 0 & 0 \\ -\sin\phi_d & \cos\phi_d & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x - x_d \\ y - y_d \\ \phi - \phi_d \\ \theta - \theta_d \\ u_m - u_R \end{bmatrix},$$

where θ_d is the desired orientation angle, and ϕ_d is the tangent angle to the path. Note that, in this work, the desired orientation angles of the robot are predetermined on the path. This setting is very useful when the robot is required to orient itself to a specific direction.

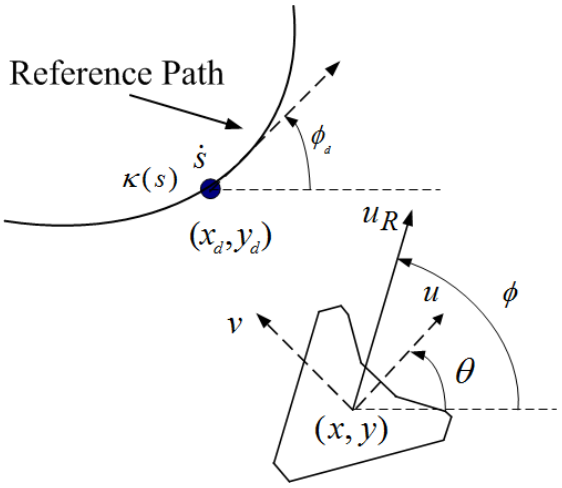

Fig. 3: The path following problem where a small black dot represents the location of the virtual vehicle.

The error state dynamic model chosen in the Frenet frame is derived using Eq. (1), Eq. (2) and Eq. (7), resulting in:

$$\dot{x}_e = y_e \kappa(s)\dot{s} - \dot{s} + u_m \cos\phi_e, \tag{9}$$

$$\dot{y}_e = -x_e \kappa(s)\dot{s} + u_m \sin\phi_e, \tag{10}$$

$$\dot{\phi}_e = \psi - \kappa(s)\dot{s}, \tag{11}$$

$$\dot{\theta}_e = \omega - \frac{\partial\theta_d}{\partial s}\dot{s}, \tag{12}$$

$$\dot{\eta}_e = a_m - \dot{u}_R, \tag{13}$$

where $\kappa(s)$ is the path's curvature. From Eq. (9), the following system control inputs are redefined:

$$\vec{u}_e = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} -\dot{s} + u_m \cos\phi_e \\ \psi - \kappa(s)\dot{s} \\ \omega - \frac{\partial\theta_d}{\partial s}\dot{s} \\ a_m - \dot{u}_R \end{bmatrix}. \tag{14}$$

The error dynamic model defined in Eq. (9), Eq. (10), Eq. (11), Eq. (12), Eq. (13) and Eq. (14) is used in the MPC framework designed in the next subsection. Moreover, to show the effectiveness of our control scheme, a comparison with the feedback control laws proposed by Oftadeh et al. [11] has been conducted (see Subsection 3.2.).

3.1. MPC Design

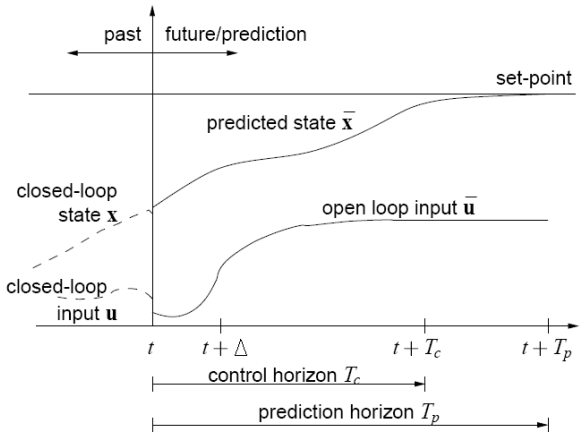

Fig. 4: Principle of model predictive control [18].

MPC is one of the most successful control techniques used in industry. It is based on a minimization of predicted tracking errors and control effort with constraints on the control inputs and the state variables over a finite horizon. At each sampling time, it generates an optimal control input sequence after the minimization problem is solved. The first element of this control input sequence is applied to the system. The problem is then solved again at the next sampling time with the updated process measurements and a shifted horizon. The conceptual structure of MPC is depicted in Fig. 4. The reader is referred to [18] and [19] for more details.

Although MPC is obviously not a new control method according to a review paper on MPC for a mobile robot [20], there is a small number of publications dealing with MPC for path following problems. Thus, the aim of this paper is to achieve path following control for an OMR in such a way that the robot's forward velocity is close to the desired one and the path following control is attained. Furthermore, robot constraints can be handled straightforwardly as hard constraints in the optimization problem so that the robot can travel safely without constraint violations.

The control input applied to the system is obtained by solving the following finite horizon open-loop optimal control problem at each sampling time:

$$\min_{\bar{\mathbf{u}}(\cdot)} \int_t^{t+T_p} F(\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau))\, d\tau, \tag{15}$$

subject to:

$$\dot{\bar{\mathbf{x}}}(\tau) = \mathbf{f}(\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau)), \tag{16}$$

$$\bar{\mathbf{u}}(\tau) \in \mathcal{U} \quad \forall\tau \in [t, t+T_c], \tag{17}$$

$$\bar{\mathbf{x}}(\tau) \in \mathcal{X} \quad \forall\tau \in [t, t+T_p], \tag{18}$$

$$\|\bar{\mathbf{x}}(t+T_p)\|_P \leq \beta \|\bar{\mathbf{x}}(t)\|_P \quad \beta \in [0,1), \tag{19}$$

where $F(\bar{\mathbf{x}}, \bar{\mathbf{u}}) = \bar{\mathbf{x}}^T \mathbf{Q} \bar{\mathbf{x}} + \bar{\mathbf{u}}^T \mathbf{R} \bar{\mathbf{u}}$. The bar denotes an internal controller variable. At the beginning of each sampling time, $\bar{\mathbf{x}} = \mathbf{x}_e, \bar{\mathbf{u}} = \mathbf{u}_e$. T_p represents the length of the prediction horizon, and T_c denotes the length of the control horizon ($T_c \leq T_p$). The deviations from the desired values are weighted by the positive definite matrices $\mathbf{Q}$, and $\mathbf{R}$, where:

- $\mathbf{Q} = \mathrm{diag}\{q_{11}, q_{22}, q_{33}, q_{44}, q_{55}\}$,
- $\mathbf{R} = \mathrm{diag}\{r_{11}, r_{22}, r_{33}, r_{44}\}$.

Besides wheel velocities maintained within boundaries (i.e., $\dot{q}_i(t) \leq \dot{q}_{\max}$, where $i = 1, 2, 3$), the constraints in Eq. (17) denote bounded control inputs as follows:

$$\begin{bmatrix} 0 \\ \psi_{\min} \\ \omega_{\min} \\ a_{m,\min} \\ \alpha_{\min} \end{bmatrix} \leq \begin{bmatrix} \dot{s} \\ \psi \\ \omega \\ a_m \\ \alpha \end{bmatrix} \leq \begin{bmatrix} \dot{s}_{\max} \\ \psi_{\max} \\ \omega_{\max} \\ a_{m,\max} \\ \alpha_{\max} \end{bmatrix}, \tag{20}$$

where α is angular acceleration, i.e., $\alpha = \dot{\omega}$.

A so-called contractive constraint [21] is defined in the last inequality end constraint Eq. (19). It requires that, at time t, the system states at the

end of the prediction horizon, i.e., $\bar{\mathbf{x}}(t+T_p)$ are contracted in norm with respect to the states at the beginning of the prediction, $\bar{\mathbf{x}}(t)$. $\beta \in [0,1)$ and the positive definite matrix $\mathbf{P}$ are two parameters that determine how much contraction is required. The reader is referred to [21] for stability analysis.

3.2. Feedback Control Laws with Low-Level Constraint Handling

The following feedback control laws for $\dot{s}$, ω and ϕ taken from [11] with slight modification is given by:

$$\dot{s} = (k_1 x_e + \cos(\sigma(y_e)))u_R = k_s u_R, \tag{21}$$

$$\omega = (-k_3\theta_e + \frac{\partial \theta_d}{\partial s}k_s)u_R, \tag{22}$$

$$\phi = \phi_d - \sigma(y_e), \tag{23}$$

where:

$$\begin{gathered}\sigma(y_e) = \arcsin\frac{k_2 y_e}{|y_e| + \epsilon}, \\ k_s = k_1 x_e + \cos(\sigma(y_e)), \\ 0 < k_1 \leq 1, \epsilon > 0, \\ k_1, k_3 > 0,\end{gathered} \tag{24}$$

The proof for semi-global exponential stability can be found in [11] with $u_R > 0$. To handle the robot constraints, a conventional approach shown in Alg. 1 has been used. However, this solution does not utilize the full capacity of the wheel's maximum velocity as shown in the simulation results. In Alg. 1, each component of the wheel velocities is scaled down such that no components are out of acceptable bounds. Function max() returns the maximum value in the array.

Algorithm 1 Velocity scaling.

INPUT: $\vec{q}$ and $\dot{q}_{\max}$
OUTPUT: $\vec{q}$
1: $factor \leftarrow 1$
2: $maxSpd \leftarrow \max(|\dot{q}_1|, |\dot{q}_2|, |\dot{q}_3|)$
3: **if** $maxSpd > \dot{q}_{\max}$ **then**
4: $\quad factor \leftarrow \dot{q}_{\max}/maxSpd$
5: **end if**
6: $\vec{q} \leftarrow \vec{q} \cdot factor$

4. Simulation Experiments

The control strategy proposed in this work was evaluated through simulation experiments. The following eight-shaped path was considered as a desired reference path:

$$x_d(t) = 1.8\sin(t), \tag{25}$$

$$y_d(t) = 1.2\sin(2t). \tag{26}$$

This reference path was numerically parameterized by the curvilinear abscissa s, while the robot's desired orientation angles were given as follows: $\theta_d(s) = \pi s \cdot 2^{-1}$, 0, and $\pi s \cdot 2^{-1}$ at $s = 0.0 - 4.0$ m, $s = 4.0 - 8.0$ m, and $s = 8.0 - 12.0$ m, respectively. All snapshots shown in the simulation results were taken at every 2 s (except for the last one).

Three simulation experiments were conducted and the results were compared. In each simulation, the initial conditions for the OMR were given as:

$$\begin{bmatrix} x \\ y \\ \phi \\ \theta \\ u_m \end{bmatrix} = \begin{bmatrix} -0.6 \\ -0.25 \\ -\pi/4 \\ 0 \\ 0 \end{bmatrix}. \tag{27}$$

The forward velocity u_R was 0.6 m·s^{-1}, and a total traveling distance was 10 m. The feedback control laws without consideration of actuator constraints were first implemented. The control parameters were set as follows: $k_1 = 1.1$ m^{-1}, $k_2 = 0.5$, $k_3 = 10$ m^{-1}. As seen in Fig. 5(a) and Fig. 5(b), although the OMR followed almost exactly the reference path, this control scheme cannot be used in practice due to constraint violation.

In the second simulation experiment, the feedback control laws with low-level constraint handling described in Subsection 3.2. were implemented with the same values of control parameters. As seen in Fig. 6(a), there were some deviations from the desired reference path at sharp corners, which means poor performance for path following control. However, the wheels' velocity constraints were not violated, as required.

In the last simulation experiment, our proposed MPC-based control strategy was evaluated. It was carried out by using a set of the following parameters:

- $\mathbf{Q} = \text{diag}(200, 1000, 5, 0.1, 20)$,
- $\mathbf{R} = \text{diag}(1, 0.1, 0.01, 0.01)$,
- $\mathbf{P} = \text{diag}(1, 1, 0.1, 0.1, 0.1)$, $\beta = 0.99$,
- $\dot{s}_{\max} = 1.2$ m·s^{-1},
- $\psi_{\max} = 2$ rad·s^{-1}, $\psi_{\min} = -2$ rad·s^{-1},
- $\omega_{\max} = 2$ rad·s^{-1}, $\omega_{\min} = -2$ rad·s^{-1},
- $a_{\text{m,max}} = 1$ m·s^{-2}, $a_{\text{m,min}} = -1$ m·s^{-2},
- $\alpha_{\max} = 2$ rad·s^{-2}, $\alpha_{\min} = -2$ rad·s^{-2},
- $T_p = T_c = 5\Delta$, $\Delta = 0.05$ s,

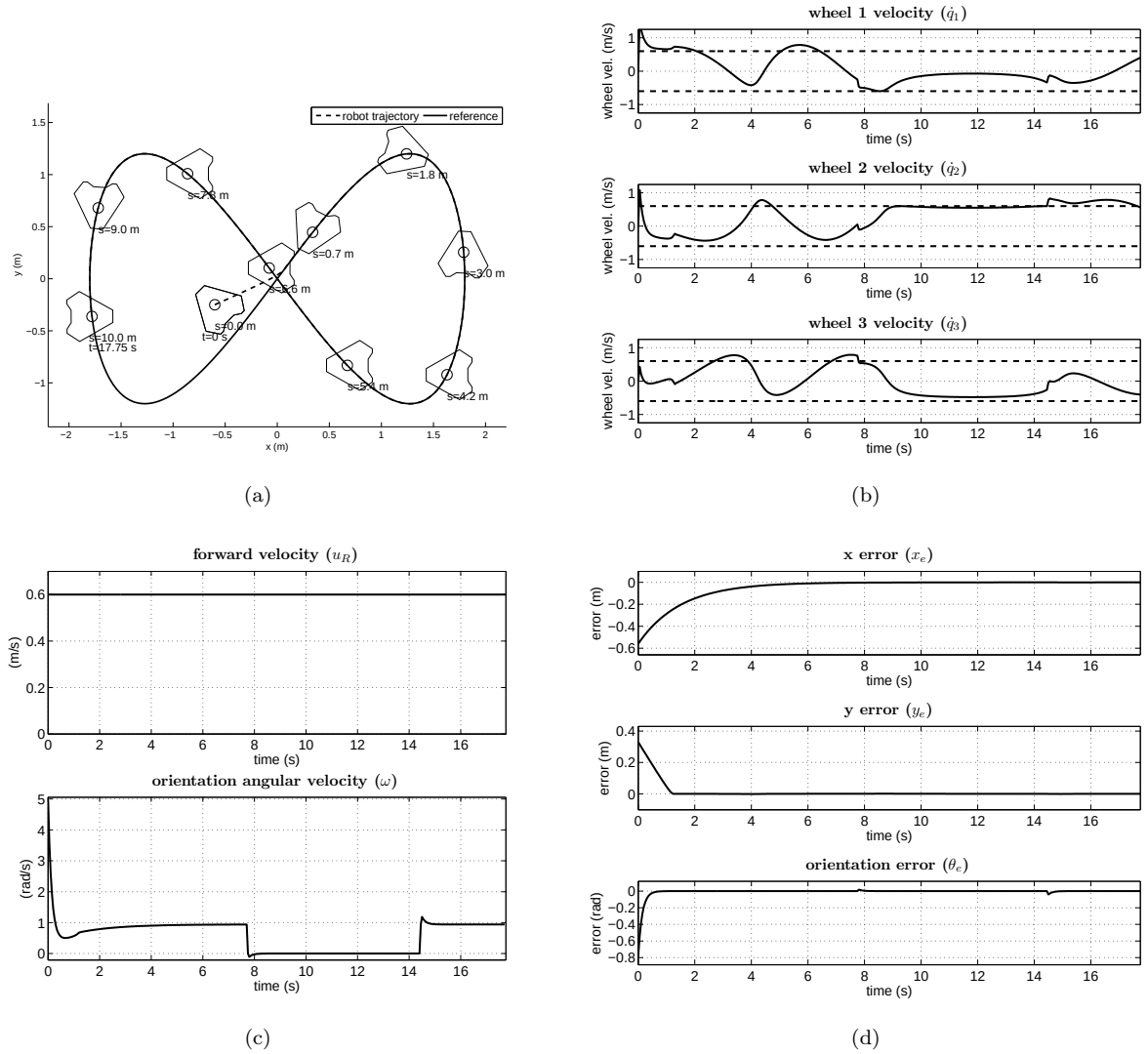

Fig. 5: The simulation results when the feedback control laws [11] without constraint handling were implemented: (a) superimposed snapshots, (b) wheel velocities, (c) robot velocities with respect to the body frame and (d) robot state errors with respect to the path coordinate.

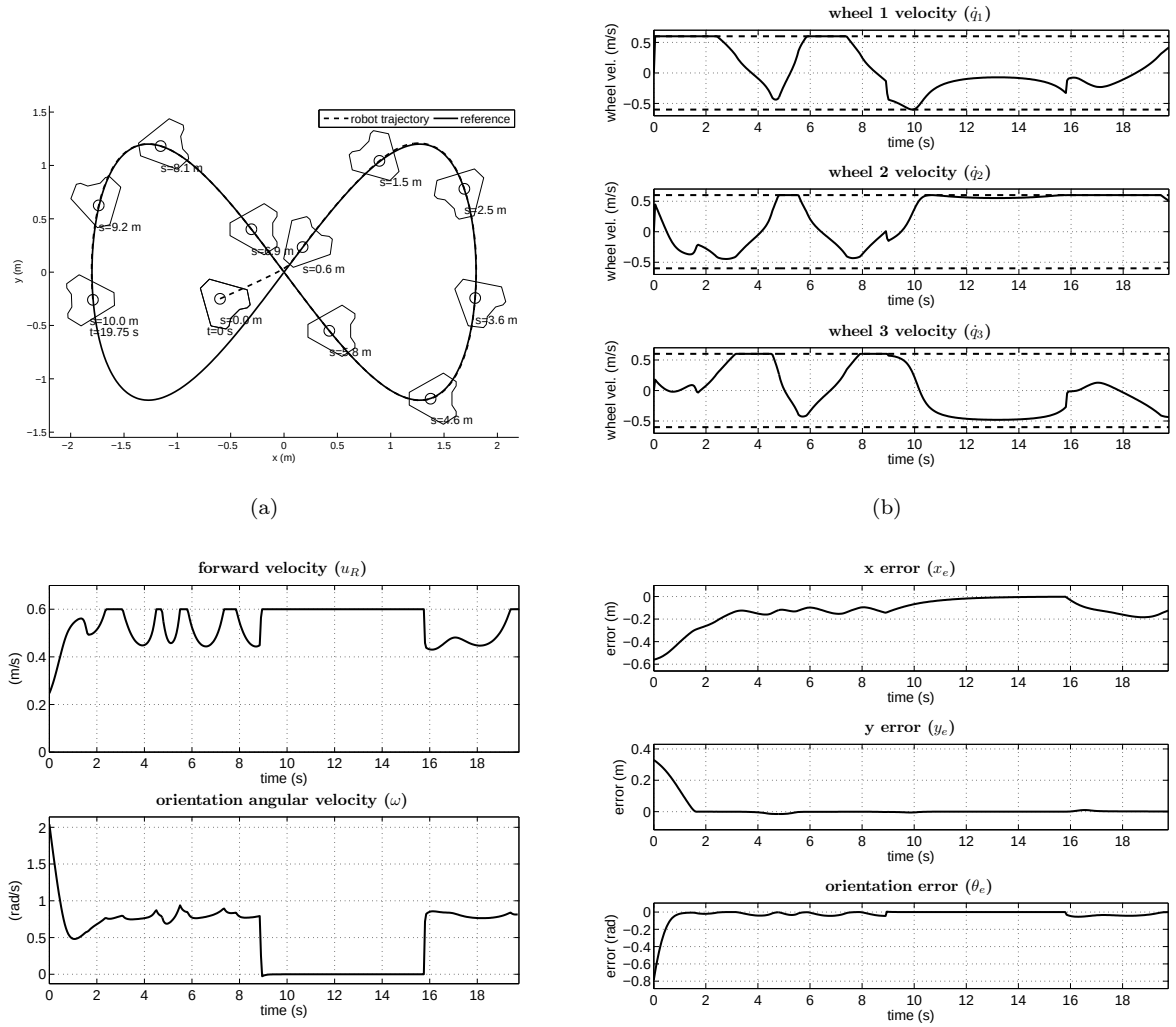

Fig. 6: The simulation results when the feedback control laws [11] with low-level constraint handling were employed: (a) superimposed snapshots, (b) wheel velocities, (c) robot velocities with respect to the body frame and (d) robot state errors with respect to the path coordinate.

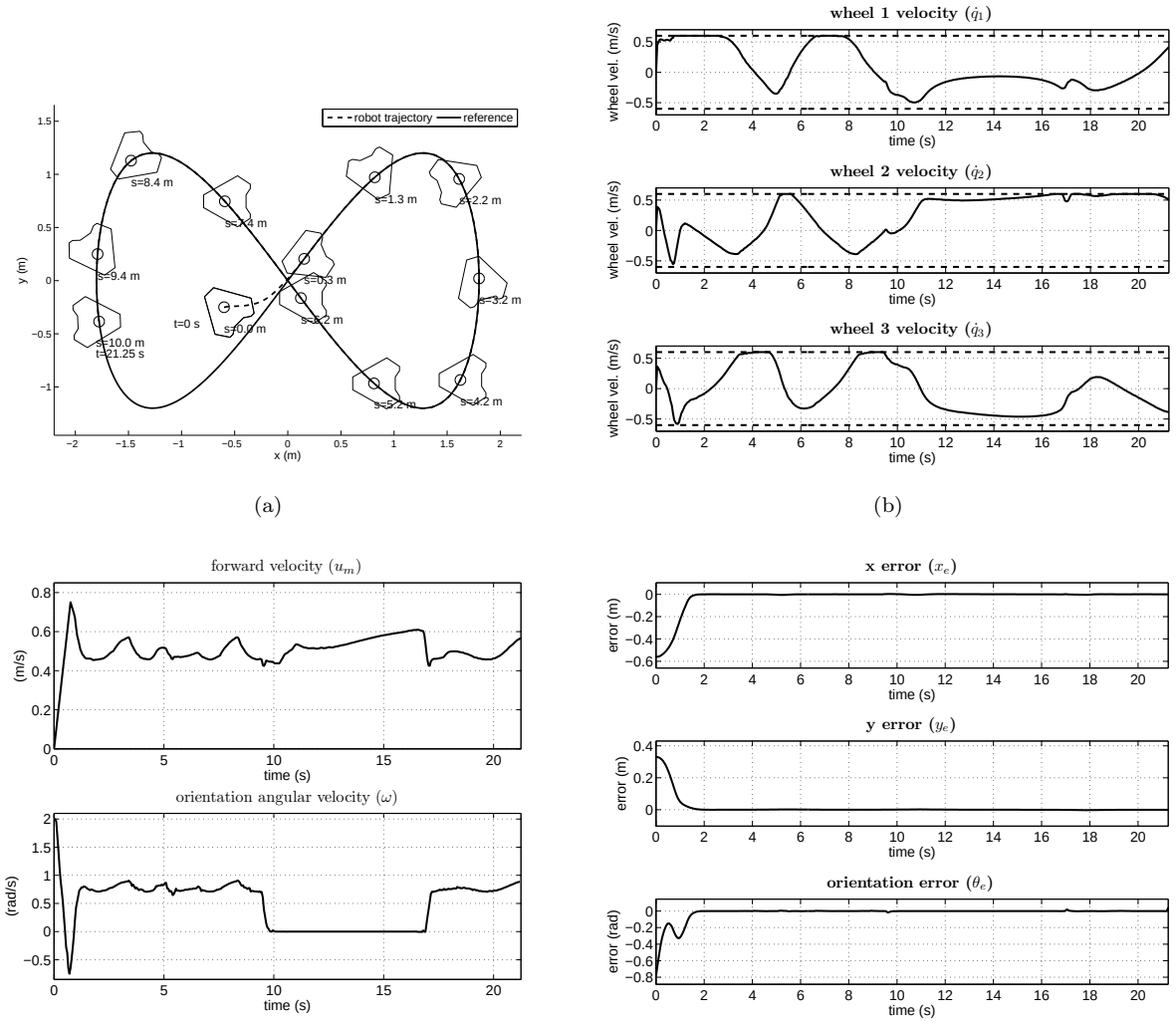

Fig. 7: The simulation results when the MPC-based control strategy was implemented: (a) superimposed snapshots, (b) wheel velocities, (c) robot velocities with respect to the body frame and (d) robot state errors with respect to the path coordinate.

where Δ is the sampling time.

Figure 7 shows the simulation results when our proposed control strategy was implemented. The OMR followed almost exactly the reference path (see Fig. 7(a)), i.e., the position errors and orientation errors defined in Eq. (8) converged to zero as depicted in Fig. 7(d). However, the forward velocity was less than 0.6 $\mathrm{m{\cdot}s^{-1}}$ during $s = 0.0 - 4.0$ m since the large rotational velocity was required. When the requirement of the rotational velocity became lower, the forward velocity became closer to the desired one (see Fig. 7(c)), while each wheel's velocity was bounded (see Fig. 7(b)).

5. Conclusions

An OMR with no nonholonomic constraints has remarkable advantages over more common design platforms like car-like robots and differential drive robots. In particular, it can move in any direction regardless of current pose and, at the same time, it can attain any desired orientation. Thus, this kind of maneuverability is especially preferred for congested applications.

In this paper, we presented an MPC scheme to solve the path following control problem of an OMR. The proposed MPC controller can handle robot constraints straightforwardly. Thus, the OMR can follow a reference path safely. Moreover, the forward velocity was optimized in the sense that it decreased when the large rotational velocity was required and it was close to the desired one when the capacity of the wheel's velocity was available.

Acknowledgment

The author gratefully acknowledges the financial support from National Science and Technology Development Agency (NSTDA), Thailand.

References

[1] SOETANTO, D., L. LAPIERRE and A. PASCOAL. Adaptive, non-singular path-following control of dynamic wheeled robots. In: *42nd IEEE International Conference on Decision and Control*. Piscataway: IEEE, 2003, pp. 1765–1770. ISBN 0-7803-7924-1. DOI: 10.1109/CDC.2003.1272868.

[2] AGUIAR, A. P., D. B. DACIC, J. P. HESPANHA and P. KOKOTOVIC. Path-following or reference-tracking? An answer relaxing the limits to performance. In: *Proceedings of the IFAC/EURON Symposium on Intelligent Autonomous Vehicles*. Lisbon: Elsevier, 2004, pp. 1–6. ISBN 008-044237-4.

[3] MICAELLI A. and C. SAMSON. Trajectory-tracking for unicycle-type and two-steering-wheels mobile robots. In: *HAL-Inria* [online]. 1993. Available at: `https://hal.inria.fr/inria-00074575/document`.

[4] PLASKONKA, J. Different Kinematic Path Following Controllers for a Wheeled Mobile Robot of (2,0) Type. *Journal of Intelligent*. 2013, vol. 71, iss. 3, pp. 1–18. ISSN 0921-0296. DOI: 10.1007/s10846-013-9879-6.

[5] ALTAFINI, C. Following a path of varying curvature as an output regulation problem. *IEEE Transactions on Automatic Control*. 2002, vol 47, iss. 9. pp. 1551–1556. ISSN 0018–9286. DOI: 10.1109/TAC.2002.802750.

[6] CONCEICAO, A., H. OLIVEIRA, A. SILVA, D. OLIVEIRA and A. MOREIRA. A nonlinear model predictive control of an omni-directional mobile robot. In: *Proceedings of the IEEE International Symposium on Industrial Electronics*. Vigo: IEEE, 2007, pp. 2161–2166. ISBN 978-1-4244-0754-5. DOI: 10.1109/ISIE.2007.4374943.

[7] PETROV, P. and L. DIMITROV. Nonlinear path control for a differential-drive mobile robot. *RECENT Journal*. 2010, vol. 11, iss. 1. pp. 41–45. ISSN 1582-0246.

[8] AICARDI, M., G. CASALINO, A. BICCHI and A. BALESTRINO. Closed loop steering of unicycle like vehicles via Lyapunov techniques. *IEEE Robotics and Automation Magazine*. 1995, vol. 2, iss. 1, pp. 27–35. ISSN 1070-9932. DOI: 10.1109/100.388294.

[9] FAULWASSER, T., B. KERM and R. FINDEISEN. Model predictive path-following for constrained nonlinear systems. In: *Proceedings of the 48th IEEE Conference on Decision and Control*. Shanghai: IEEE, 2009, pp. 8642–8647. ISBN 978-1-4244-3871-6. DOI: 10.1109/CDC.2009.5399744.

[10] KANJANAWANISHKUL, K. Path following control of a mobile robot using contractive model predictive control. *Applied Mechanics and Materials*. 2013, vol. 397–400, iss. 1, pp. 1366–1372. ISSN 1662–7482. DOI: 10.4028/www.scientific.net/AMM.397-400.1366.

[11] OFTADEH, R., R. GHABCHELOO and J. MATTILA. A novel time optimal path following controller with bounded velocities for mobile robots

with independently steerable wheels. In: *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Tokyo: IEEE, 2013, pp. 4845–4851. ISBN 978-146736358-7. DOI: 10.1109/IROS.2013.6697055.

[12] YU, S., X. LI, H. CHEN and F. ALLGOWER. Nonlinear model predictive control for path following problems. In: *Proceedings of the 4th IFAC Nonlinear Model Predictive Control Conference*. Noordwikkerhout: IFAC, 2012, pp. 145–150. ISBN 978-390282307-6. DOI: 10.3182/20120823-5-NL-3013.00003.

[13] CAMPION, G., G. BASTIN, B. DANDREA-NOVEL and Y. SHUYOU. Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation*. 2013, vol. 12, iss. 1. pp. 47–62. ISSN 1042-296X. DOI: 10.1109/70.481750.

[14] VAZQUEZ, A. and M. VELASCO-VILLA. Path-tracking dynamic model based control of an omnidirectional mobile robot. In: *Proceedings of the World Congress*. Seoul: IFAC, 2008, pp. 5365–5370. ISBN 978-1-1234-7890-2. DOI: 10.3182/20080706-5-KR-1001.00904.

[15] HUANG, H. and C. TSAI. Adaptive robust control of an omnidirectional mobile platform for autonomous service robots in polar coordinates. *Journal of Intelligent and Robotic Systems*. 2008, vol. 51, iss. 4. pp. 439–460. ISSN 0921-0296. DOI: 10.1007/s10846-007-9196-z.

[16] KANJANAWANISHKUL, K. and A. ZELL. Path following for an omnidirectional mobile robot based on model predictive control. In: *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA 2009)*. Kobe: IEEE, 2009, pp. 3341–3346. ISBN 978-1-4244-2788-8. DOI: 10.1109/ROBOT.2009.5152217.

[17] ORIOLO, G., A. DE LUCA and M. VENDITTELLI. WMR control via dynamic feedback linearization: design, implementation and experimental validation. *IEEE Transactions on Control Systems Technology*.2002, vol. 10, iss. 6, pp. 835–852. ISSN 1063-6536. DOI: 10.1109/TCST.2002.804116.

[18] ALLGOWER, F., R. FINDEISEN and Z. K. NAGY. Nonlinear model predictive control: from theory to application. *Journal of the Chinese Institute of Chemical Engineers*. 2004, vol. 35, no. 3, pp. 299–315. ISSN 0368-1653.

[19] MAYNE, D. Q., J. B. RAWLINGS, C. V. RAO and P. O. M. SCOKAERT. Constrained model predictive control: Stability and optimality. *Automatica*. 2000, vol. 36, iss. 6. pp. 789–814. ISSN 0005-1098. DOI: 10.1016/S0005-1098(99)00214-9.

[20] KANJANAWANISHKUL, K. Motion control of a wheeled mobile robot using model predictive control: A survey. *KKU Research Journal*. 2012, vol. 17, iss. 5. pp. 811–837. ISSN 0859-3957.

[21] KOTHARE, S. L. D. and M. MORARI. Contractive model predictive control for constrained nonlinear systems. *IEEE Transactions on Automatic Control*. 2000, vol. 45, iss. 6, pp. 1053–1071. ISSN 0018-9286. DOI: 10.1109/9.863592.

About Authors

Kiattisin KANJANAWANISHKUL was born in Trang, Thailand, in 1977. He received the B.Eng. in Electrical Engineering from Prince of Songkla University, Thailand in 2000. He received the M.Sc. in Mechatronics from University of Siegen, Germany in 2006. He received his Ph.D. in Computer Science from University of Tuebingen, Germany in 2010. Since 2010, he has been employed at the Faculty of Engineering, University of Mahasarakham, Thailand. His research interests include cooperative and distributed control, model predictive control, intelligent control, multi-robot systems, and robotic motion control.