International Journal of Engineering

Journal Homepage: www.ije.ir

Path Following and Velocity Optimizing for an Omnidirectional Mobile Robot

K. Kanjanawanishkul *

Mechatronics Research Unit, Faculty of Engineering, Mahasarakham University, Khamriang, Kantarawichai, Mahasarakham, Thailand

PAPER INFO

Paper history:
Received 06 October 2014
Accepted in revised form 29 January 2015

Keywords:
Path Following Control
Robot Motion
Model Predictive Control
Omnidirectional Mobile Robots
Actuator Constraints

ABSTRACT

In this paper, the path following controller of an omnidirectional mobile robot (OMR) has been extended in such a way that the forward velocity has been optimized and the actuator velocity constraints have been taken into account. Both have been attained through the proposed model predictive control (MPC) framework. The forward velocity has been included into the objective function, while the actuator saturation has been considered as hard constraints. As shown in the simulation results, the OMR can converge to and follow a reference path successfully and safely. The forward velocity of the robot was close to the desired one and the desired orientation angle was achieved at a given point on the path, while the actuator constraints were not violated. Furthermore, to show the effectiveness of our proposed framework, a comparison with conventional approaches used to bound actuator constraints has been conducted. Mean squared error (MSE), integral squared error (ISE), and traveling distance were used as performance indices. As seen in the results, the proposed control strategy outperforms the conventional approaches. The proportion between translational and rotational velocities was optimized, although the limitation of the rotational and translational velocities was coupled via the OMR's orientation angle.

doi: *10.5829/idosi.ije.2015.28.04a.07*

1. INTRODUCTION

Over the past few decades, omnidirectional mobile robots (OMRs) have become increasingly popular in mobile robot applications, since they have some remarkable advantages over nonholonomic mobile robots. In particular, they can move in any direction without the need of reorientation [1-3]. This capability is very desirable for a variety of applications.

Although, there have been a large number of publications dealing with OMRs, path following control is still one of the most active topics in robotic research [4]. It presents some advantages over trajectory tracking [5]. For example, the time dependence of the trajectory tracking problem is not included in the path following problem. The robot converges to the path with less aggression. Control signals are less likely pushed into saturation. Pioneering work in this research area can be found in [6] and the references therein.

*Corresponding Author's Email: kiattisin_k@hotmail.com (K. Kanjanawanishkul)

The aim of the path following problem is that the path following controller determines the robot's moving direction such that the robot is forced to the reference path without any consideration in temporal specifications, while the robot's forward velocity tracks a desired velocity profile. In general, to solve this problem, there are two concepts for path parameterization [7], i.e., the Frenet frame with an orthogonal projection of a robot on the given path [6, 8-11] and the Frenet frame with a non-orthogonal projection of a robot on the given path [4, 12-16]. In the first method, the position of the virtual vehicle to be followed by a real one is determined by the orthogonal projection of the real 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 initial condition constraints of the first method, a desired geometric path is parameterized by the curvilinear abscissa, s ($s \in \Re$). The velocity of the virtual vehicle, $\dot{s}(t)$, can be also seen as an additional degree of freedom, which has been controlled explicitly using different path following control laws.

Please cite this article as: K. Kanjanawanishkul, Path Following and Velocity Optimizing for an Omnidirectional Mobile Robot, International Journal of Engineering (IJE), IJE TRANSACTIONS A: Basics Vol. 28, No. 4, (April 2015) 537-545

Some related work is as follows: Vazquez et al. [17] presented an extension of the computed torque control commonly used in the field of robot manipulators to the case of an OMR. Conceicao et al. [9] proposed a nonlinear model based predictive controller for an OMR. The cost function penalizes the robot position error, the robot orientation angle error and the control effort. Huang and Tsai [18] presented an adaptive robust control method for path following of an OMR with actuators' uncertainties. Kanjanawanishkul and Zell [14] used model predictive control to generate an optimal velocity of the virtual vehicle. Recently, Oftadeh et al. [15] proposed a new solution to the path following problem of independently steered mobile robots that possess omnidirectionality. The speed of the base can be determined analytically to keep the steering and driving velocities of the wheels under prespecified values. In this work, the path following controller proposed by Oftadeh et al. [15] was adopted with slight modification. The interesting point of this controller is that stability of the path following controller is attained without regard to the forward velocity. As a result, the forward velocity becomes an independent variable. This allows us to find its optimal value by using the model predictive control (MPC) framework. It is optimal in the sense that the forward velocity is close to the desired one, whereas the actuator constraints are not violated. To do so, in our MPC framework design, a path following closed-loop system is used as a predicted model. The forward velocity is integrated into the objective function, while the actuator constraints are considered as hard constraints of the MPC framework. The rest of the paper is organized as follows: in Section 2, the OMR kinematics is derived, while the path following controller proposed by Oftadeh et al. [15] with slight modification is explained in Section 3. Our MPC framework with consideration of actuator constraints and optimal forward velocity are developed in Section 4. Then, simulation experiments are conducted in Section 5 to show the effectiveness of our proposed controller.

2. KINEMATIC MODELING

From Figure 1, where the kinematic diagram of an OMR with three independent driving wheels equally spaced at 120 degrees from one to another is depicted, the kinematic model of an OMR can be given by:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix}=\begin{bmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}u\\ v\\ \omega\end{bmatrix} \tag{1}$$

where, $\mathbf{x} = [x, y, \theta]^T$ is the state vector in the world frame (X_w, Y_w) which is fixed at the ground. θ denotes the angle of the robot's orientation. $\mathbf{u} = [u, v, \omega]^T$ is the vector of robot velocities observed in the body frame (X_m, Y_m) which is fixed at the moving robot with the origin at the center of the robot. u and v are the translational velocities and ω is the rotational velocity.

Since translation and rotation of OMRs can be separately controlled, Equation (1) can be rewritten by decoupling translation and rotation:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\phi}\\ \dot{\theta}\end{bmatrix}=\begin{bmatrix}u_R\cos\phi\\ u_R\sin\phi\\ \psi\\ \omega\end{bmatrix} \tag{2}$$

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 with respect to the body frame can be determined by:

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

To enable translation and rotation of the robot body, the velocities of all three wheels can be derived with respect to the body frame as follows:

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

where, $\dot{\mathbf{q}} = [\dot{q}_1 \quad \dot{q}_2 \quad \dot{q}_3]^T$ is the vector of wheel velocities. L_w is the distance from the robot's center of mass (point R in Figure 1) to the wheel center and δ denotes the angle of the wheel orientation in the body frame. In general, the wheel velocities can be divided into two components:

$$\dot{\mathbf{q}}_t=\begin{bmatrix}\cos\delta & \sin\delta\\ -\cos\delta & \sin\delta\\ 0 & -1\end{bmatrix}\begin{bmatrix}u\\ v\end{bmatrix},\qquad \dot{\mathbf{q}}_r=\begin{bmatrix}L_w\\ L_w\\ L_w\end{bmatrix}\omega \tag{5}$$

where, $\dot{\mathbf{q}}_t$ and $\dot{\mathbf{q}}_r$ 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., $\|\dot{\mathbf{q}}_t+\dot{\mathbf{q}}_r\|_\infty \le \dot{q}_{\max}$.

The relationship between the state vector in the world frame and the vector of wheel velocities can be derived by substituting Equation (4) into Equation (1), resulting in:

$$\dot{\mathbf{x}}(t) = R_Z(\theta)P(0)\dot{\mathbf{q}}(t) \tag{6}$$

where:

$$R_Z(\theta)=\begin{bmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix},\quad P(0)=\begin{bmatrix}\sqrt{3}/3 & -\sqrt{3}/3 & 0\\ 1/3 & 1/3 & -2/3\\ 1/(3L_w) & 1/(3L_w) & 1/(3L_w)\end{bmatrix}$$

with $\delta = \pi/6$ rad.

For a given θ, the transformation $R_z(\theta)P(0)$ in Equation (6) maps the cube $\Im(t) = \{\dot{\mathbf{q}}(t) \mid \ |\dot{q}_i(t)| \le \dot{q}_{\max}\}$, where $i = 1, 2, 3$, into the tilted cuboid $R_z(\theta)P(0)\ \Im(t)$.

To depict the problems of actuator constraints and the coupling between translational and rotational limitations, an illustrative example is given as follows. The linear transformation $P(0)$ maps the cube $\Im(t)$ (see Figure 2(a)) into the tilted cuboid $P(0)\Im(t)$ (see Figure 2(b)). The maximum rotational velocity (ω) of 1.5 rad/s along the ω - axis is calculated by using $L_w = 0.2$ m and $|\dot{q}_i(t)| \le 0.3$ m/s and the transformation $R_z(\theta)$ then rotates this cuboid about the ω - axis. From Figure 2(b), the maximum forward velocity is 0.4 m/s. However, when the rotational velocity becomes larger and larger in some situations, e.g., during converging the robot to the path or large rotational velocities required, the velocity of at least one of three wheels may reach the boundary. For example, based on the following state variables and parameter settings: u_R = 0.2 m/s, $\dot{q}_{\max}$ = 0.3 m/s, ϕ = 90°, θ = 0°, and ω = 2 rad/s, then $\dot{q}_1$ = 0.5732 m/s, $\dot{q}_2$ = 0.5732 m/s, and $\dot{q}_3$ = 0.2 m/s are obtained. That is the velocities of wheels 1 and 2 exceed the wheel's maximum velocity. Thus, to achieve the desired orientation angle with large rotational velocity, the forward velocity is required to decrease in order that the allowable rotational velocity can increase. Moreover, boundaries of the tilted cuboid is changed depending on the robot orientation (θ) as seen in Equation (6). This means that the translational and rotational limitations are coupled via θ [19].

3. THE PATH FOLLOWING CONTROLLER

In this work, the concept of the virtual vehicle with its body frame moving along the reference path is employed. In Figure 3, a small black dot represents the location of the virtual vehicle which is followed by the real robot such that the forward velocity, u_R, tracks a desired velocity profile and the velocity of the virtual vehicle ($\dot{s}$) converges to u_R. Then, the error state vector $\mathbf{x}_e$ between the state vector of the robot and that of the virtual vehicle can be expressed in the path frame as follows:

$$\mathbf{x}_e = \begin{bmatrix} x_e \\ y_e \\ \phi_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos\phi_d & \sin\phi_d & 0 & 0 \\ -\sin\phi_d & \cos\phi_d & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - x_d \\ y - y_d \\ \phi - \phi_d \\ \theta - \theta_d \end{bmatrix} \tag{7}$$

where, $[x_d \quad y_d \quad \phi_d \quad \theta_d]^T$ is the state vector of the virtual vehicle, θ_d is the desired orientation angle, and ϕ_d is the tangent angle to the path. The following error state dynamic model chosen in the path frame is derived by using Equations (1), (2) and (7):

$$\begin{aligned} \dot{x}_e &= y_e\kappa(s)\dot{s} - \dot{s} + u_R\cos\phi_e \\ \dot{y}_e &= -x_e\kappa(s)\dot{s} + u_R\sin\phi_e \\ \dot{\phi}_e &= \dot{\phi} - \dot{\phi}_d \\ \dot{\theta}_e &= \omega - \omega_d \end{aligned} \tag{8}$$

where, $\kappa(s)$ is the path's curvature, $\omega_d = \dot{\theta}_d$.

Figure 4 shows the block diagram of the overall system. In the block of the path-following closed-loop system, the feedback control laws [15] with slight modification was adopted, while the block of the MPC framework was used to find the optimal forward velocity. The following path following control laws for $\dot{s}$, ω, and ϕ were developed such that the robot follows the virtual vehicle with $\mathbf{x}_d = [x_d \quad y_d \quad \phi_d \quad \theta_d]^T$:

$$\begin{aligned} \dot{s} &= (k_1 x_e + \cos(\sigma(y_e)))u_R = k_s u_R \\ \omega &= \left(-k_3\theta_e + \frac{\partial\theta_d}{\partial s}k_s\right)u_R \\ \phi &= \phi_d - \sigma(y_e) \end{aligned} \tag{9}$$

where:

$\sigma(y_e) = \arcsin\dfrac{k_2 y_e}{|y_e| + \varepsilon}$, $k_s = k_1 x_e + \cos(\sigma(y_e))$,

$0 < k_1 \le 1$, $k_1, k_3 > 0$, $\varepsilon > 0$.

The proof for semi-global exponential stability can be found in [15] with $u_R > 0$.

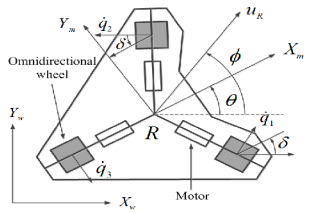
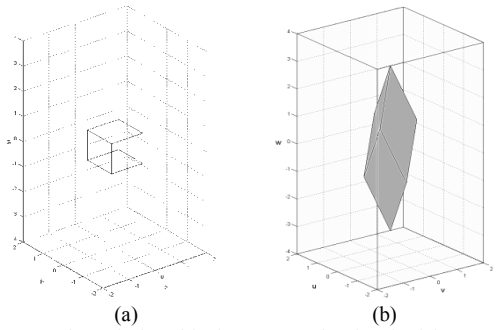


Figure 1. Coordinate frames of an OMR



(a) (b)

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

Then, the control laws, i.e., Equation (9) is substituted into the open-loop system, i.e., Equation (8), resulting in the following closed-loop system:

$$\begin{aligned}\dot{x}_e &= \left(k_s\left(\kappa\left(s\right)y_e - 1\right) + \cos\left(\sigma\left(y_e\right)\right)\right)u_R \\ \dot{y}_e &= \left(-k_s\kappa\left(s\right)x_e + \sin\left(\sigma\left(y_e\right)\right)\right)u_R \\ \dot{\theta}_e &= \left(-k_3\theta_e\right)u_R\end{aligned} \tag{10}$$

As stated in [15] and seen from Equation (10), the forward velocity (u_R) can be considered as an independent variable. Therefore, the aim of this paper is to optimize the forward velocity in the sense that it is as close as possible to the desired one, while actuator constraints must not be violated. Also, final position errors and orientation errors converge to zero, which means, the path following control for an OMR is attained. All of these requirements are met through our proposed MPC framework described in the next section.

4. OPTIMAL FORWARD VELOCITY WITH CONSTRAINTS

Due to the coupling between translational and rotational limitations, the conventional approaches which are shown in Subsections 4.2, 4.3, and 4.4 have been proposed. However, these solutions do not utilize the full capacity of the wheel's maximum velocity as shown in the simulation results. Therefore, an MPC framework with consideration of actuator constraints is proposed to ensure that the proportion between translational and rotational velocities is optimized.

As shown in Figure 4, the block diagram of the proposed control architecture is systematically depicted. The MPC block is used to generate the optimal forward velocity whereas the actuator constraints are taken into account in the online optimization of the MPC framework. This is one of its well-known advantages in using the MPC strategy. Likewise, system performance can be improved since future information (i.e., the reference path) is known beforehand.

4. 1. MPC Framework Design MPC is one of the most successful control techniques used in industry. It relies on a finite-horizon minimization of predicted state errors and control effort with imposing constraints on the control inputs and state variables. An optimal control input sequence is produced at each sampling time after solving an online optimization problem. Then, the first element of the control input sequence is sent to the system. The process is then repeated at the next sampling time with the updated process measurements and a shifted horizon. The conceptual structure of MPC is illustrated in Figure 5. A more comprehensive explanation can be found in [20] and [21]. In the MPC strategy, the control input sequence input is obtained by solving the following finite horizon open-loop optimal control problem at each sampling time:

$$\min_{\bar{\mathbf{u}}(\bullet)} \int_t^{t+T_p} F\left(\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau)\right)d\tau \tag{11}$$

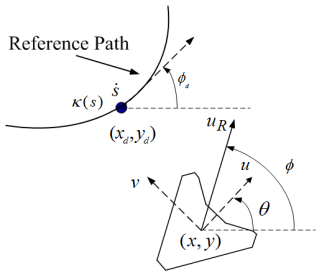


Figure 3. The path following problem where a small black dot represents the location of the virtual vehicle

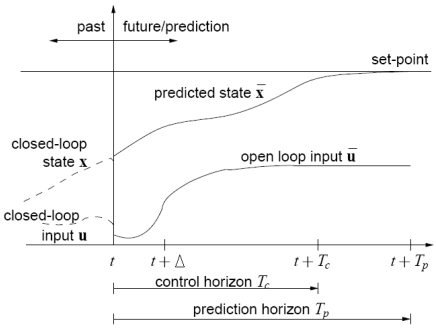


Figure 5. Principle of model predictive control [20]

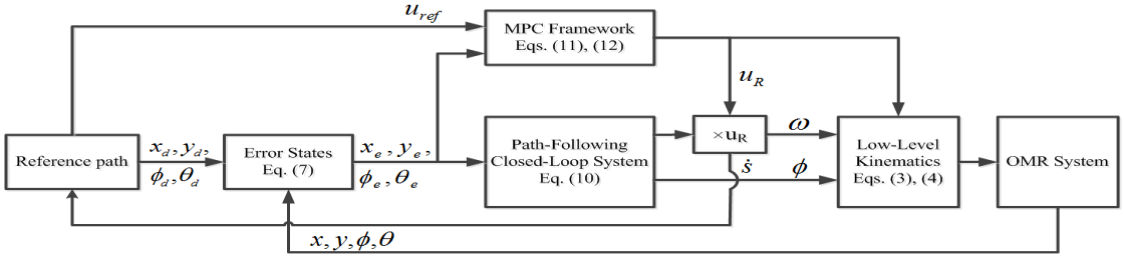


Figure 4. Block diagram of the whole system including the path-following closed-loop system and the MPC framework

subject to

$$\dot{\bar{\mathbf{x}}}(\tau) = \mathbf{f}\left(\bar{\mathbf{x}}(\tau), \bar{\mathbf{u}}(\tau)\right) \tag{12a}$$

$$\bar{\mathbf{u}}(\tau) \in U \quad \forall \tau \in [t, t+T_C] \tag{12b}$$

$$\bar{\mathbf{x}}(\tau) \in X \quad \forall \tau \in [t, t+T_P] \tag{12c}$$

where, $F\left(\bar{\mathbf{x}}, \bar{\mathbf{u}}\right) = \bar{\mathbf{x}}^{\mathbf{T}} Q \bar{\mathbf{x}} + \bar{\mathbf{u}}^{\mathbf{T}} R \bar{\mathbf{u}}$. The bar denotes an internal controller variable. At the beginning of each sampling time (i.e., at $\tau = t$), $\bar{\mathbf{x}} = \left[x_e \quad y_e \quad \theta_e\right]^T$, $\bar{\mathbf{u}} = u_R - u_{ref}$. u_{ref} is the desired forward velocity. T_P represents the length of the prediction horizon, while T_C denotes the length of the control horizon ($T_C \le T_P$). The positive definite matrices Q, and R are used to weight the deviations from the desired values. The constraints of the optimization problem include three wheel velocities which must be maintained within boundaries (i.e., $\left|\dot{q}_i(t)\right| \le \dot{q}_{\max}$, where i = 1, 2, 3).

To show the effectiveness of the proposed MPC framework, a comparison with velocity clamping, velocity scaling, and task prioritizing approaches has been conducted.

4. 2. Velocity Clamping One of the quick and dirty solutions is to clamp the commanded wheel velocities as follows:

$$\dot{q}_i = \begin{cases} \dot{q}_{\min} & \text{if} \quad \dot{q}_i < \dot{q}_{\min} \\ \dot{q}_i & \text{if} \quad \dot{q}_{\min} \le \dot{q}_i \le \dot{q}_{\max} \\ \dot{q}_{\max} & \text{if} \quad \dot{q}_i > \dot{q}_{\max} \end{cases} \tag{13}$$

where, i = 1, 2, 3, and in general, $\dot{q}_{\min} = -\dot{q}_{\max}$. However, this solution destroys the capability of the OMR in which translation and rotation are decoupled.

4. 3. Velocity Scaling As seen in Algorithm 1, each component of the wheel velocities is scaled down such that no components are out of acceptable bounds. Max () function returns the maximum value in the array. However, this strategy has to be proved that asymptotic stability is still achieved.

Algorithm 1: Velocity Scaling

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

4. 4. Task Prioritizing In this solution, a variant of the task prioritizing method [22] was applied to the algorithm used in [23]. Indiveri [22] considered that each task in controller laws should be executed in a prioritized fashion. In our variant, the minimum percentage of translational velocity $\dot{\mathbf{q}}_t$ contributed to the total wheel velocity is first given. The rest of the wheel capability is then devoted to the rotational velocity. This strategy results in the compromise between the position errors and the orientation errors. The minimum percentage of the rotational contribution may be chosen instead of translation if the user wants to specify its minimum contribution.

5. SIMULATION RESULTS

The control strategy proposed in this work was evaluated through simulation experiments. The approaches mentioned in the previous section were implemented and compared with the ideal case, i.e., no limitations on wheel velocities. Mean squared error (MSE), integral squared error (ISE), and traveling distance were used as our performance indices.

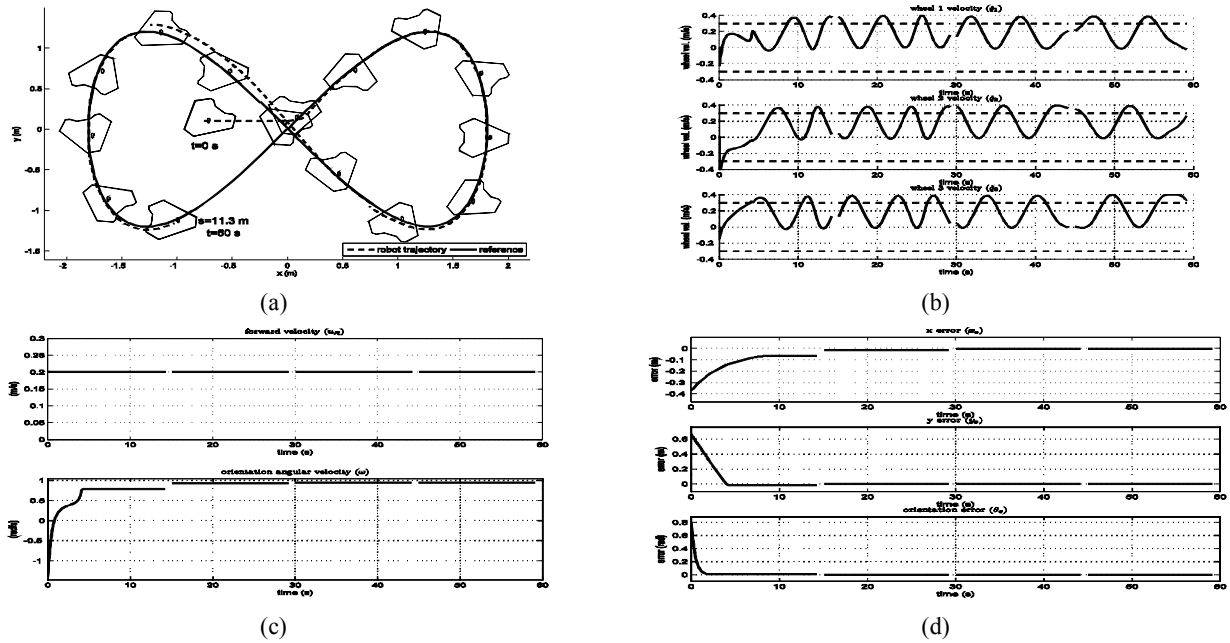
The following eight-shaped path was considered as a desired reference path:

$$\begin{aligned} x_d(t) &= 1.8\sin(t) \\ y_d(t) &= 1.2\sin(2t) \end{aligned} \tag{14}$$

This reference path was numerically parameterized by the curvilinear abscissa (s), while the robot's desired orientation angles were given by $\theta_d(s) = 3\pi s/2$. The desired forward velocity, u_{ref} was 0.2 m/s and the wheel constraint, $\dot{q}_{\max}$, was set to 0.3 m/s.

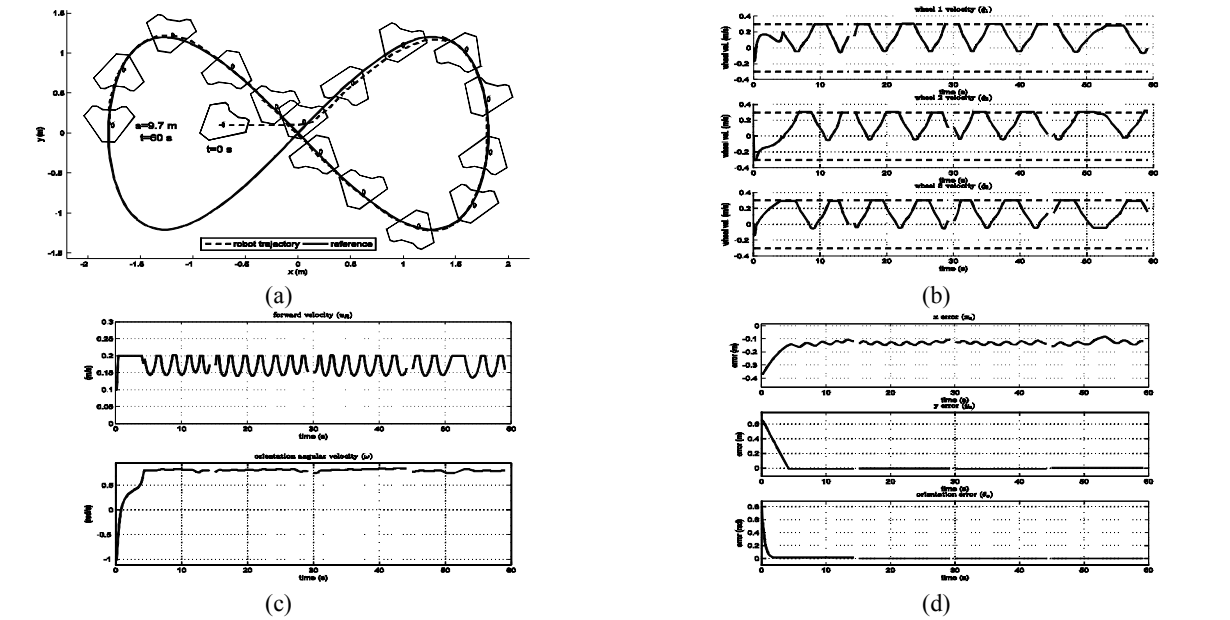
The simulation experiments were carried out by using a set of the following parameters: k_1 = 1.1, k_2 = 0.8, k_3 = 10 for the path following controller (9). For the MPC framework, the following parameters were used: Q = diag (0.1, 1, 0.1), R = 0.01, Δ = 0.05 s, where Δ is the sampling time. The initial conditions for the OMR was given as $\left[x \quad y \quad \phi \quad \theta\right]^T$ = [-0.75 0.1 0 π/4]^T and a total traveling time was 60 s. All snapshots shown in the simulation results were taken at every 4 s.

The simulation results are shown in Table 1, where our proposed MPC framework (MPC) are compared with conventional approaches, including velocity clamping (VC), velocity scaling (VS) and task prioritizing (TP), and also compared with the case of no actuator constraints (NC). In the case of using task prioritizing, two minimum percentage values of the translational velocity were set, i.e., 25 and 50%. In the case of the MPC framework, two lengths of prediction horizon were considered: $T_P = T_C = 3\Delta$ and $T_P = T_C = 15\Delta$.



(a) (b) (c) (d)

Figure 6. The simulation results when actuator constraints were not taken into account: (a) superimposed snapshots, (b) wheel velocities, (c) robot velocities, and (d) robot state errors with respect to the path coordinate



(a) (b) (c) (d)

Figure 7. The simulation results when the task prioritizing with 50% minimum percentage of the translational velocity was implemented: (a) superimposed snapshots, (b) wheel velocities, (c) robot velocities, and (d) robot state errors with respect to the path coordinate

TABLE 1. Comparison results using different strategies for handling actuator constraints

Performance Indices	NC	VC	VS	TP		MPC	
				25%	50%	3Δ	15Δ
MSE (position) (m)	0.0156	0.0303	0.0323	0.0340	0.0338	0.0131	0.0131
ISE (position) (m)	18.78	36.43	38.80	40.81	40.57	15.74	15.74
MSE (orientation) (rad)	0.0027	0.0045	0.0085	0.0027	0.0029	0.0033	0.0033
ISE (orientation) (rad)	3.25	5.43	10.17	3.29	3.51	3.99	3.99
MSE (velocity) (m/s)	0.0000	0.0011	0.0011	0.0014	0.0014	0.0015	0.0014
ISE (velocity) (m/s)	0.00	1.30	1.28	1.66	1.63	1.77	1.73
Travelling distance (m)	11.3149	9.8454	9.7198	9.6587	9.6614	9.7498	9.8415

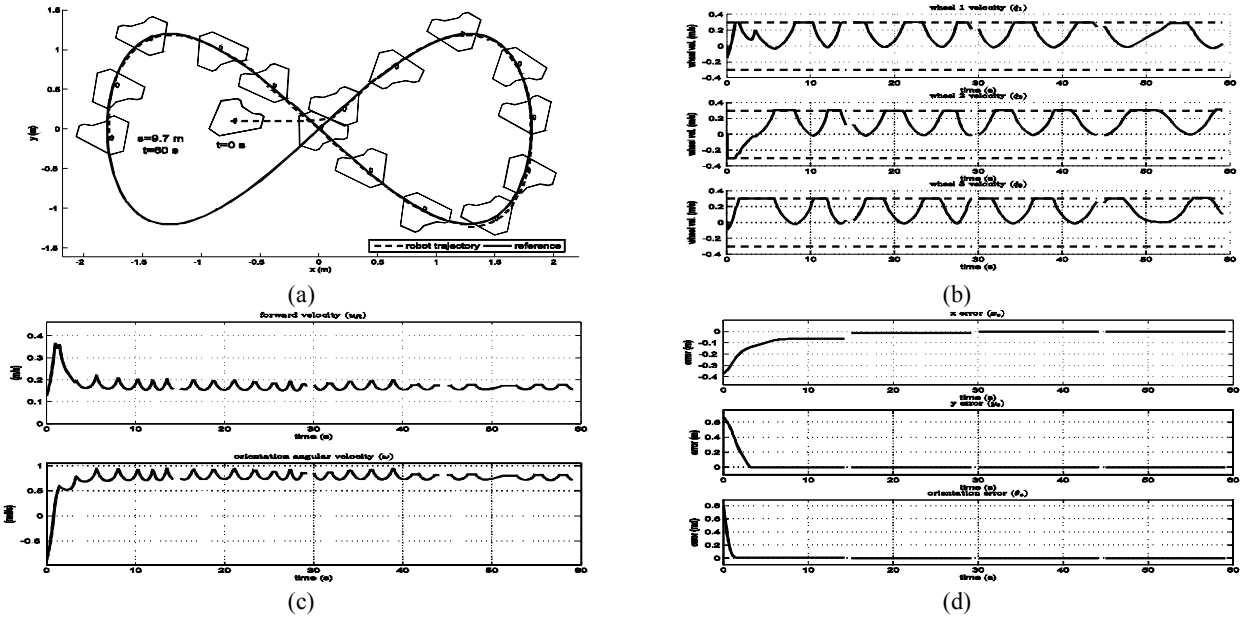


Figure 8. The simulation results when the MPC framework with TP = TC = 3Δ was employed: (a) superimposed snapshots, (b) wheel velocities, (c) robot velocities, and (d) robot state errors with respect to the path coordinate

In Table 1, MSE and ISE of position errors, orientation errors (defined in Equation (7)) and the velocity errors are given when different strategies for handling actuator constraints were implemented. Position errors and orientation errors represent the performance of the path following control and the distinct characteristic of the OMR, i.e., translations and rotations of the OMR can be controlled separately. Velocity errors and traveling distance are used to show the capability of the overall system. Obviously, the NC case gives the maximum traveling distance as shown in Table 1 and Figure 6. However, it is impossible to use this setting in practice since the summation of both translational and rotational velocities make each wheel's velocity exceeding the boundary.

When the velocity clamping (VC) and the velocity scaling (VS) approaches were used, the position errors and the orientation errors significantly increased, which means poor performance for path following control. For the case of task prioritizing (TP), the position errors increased when the minimum percentage of the translational velocity decreased. However, the orientation errors increased when the minimum percentage of the translational velocity increased. Therefore, this strategy leads to the trade-off between the position errors and the orientation errors. Moreover, as seen in Figure 7(a), there were some deviations from the desired reference path at sharp corners. In Figure 7(b), the forward velocity decreased in order that the wheels' velocity constraints were not violated, as required. The simulation results of our proposed framework are shown in Figure 8. It obviously outperforms the conventional approaches mentioned above. The position errors and the orientation errors were close to zero (see Figure 8(d)), while the actuator velocity constraints were within boundaries all the time (see Figure 8(b)). This is because our control strategy can adjust the forward velocity in order to be suitable for the path-following closed-loop system and takes the constraints into account.

6. CONCLUSIONS

An OMR without nonholonomic constraints own remarkable advantages over more common design platforms like car-like robots and differential drive robots. Regardless of its current pose, it can move instantly in any direction and at the same time, it can attain any desired orientation as seen in the simulation results. This kind of maneuverability is particularly preferred for congested applications.

In this paper, we presented an MPC scheme integrated with the path-following closed-loop system proposed by Oftadeh et al. [15]. Our control strategy can handle actuator constraints straightforwardly, while the final position errors and orientation errors were closed to zero as the requirement of the path following control. Likewise, the boundaries of translations and rotations are coupled via the angle of the robot orientation as shown in Section 2. Thanks to the control strategy proposed in the previous section, the optimal proportion between translational and rotational velocities was attained in such a way that the

translational velocity is close to the desired one and the desired orientation angle is obtained at a given point on the path. In the future, our robot should work in a real environment. Thus, we will integrate obstacle avoidance into our system and implement an on-line path planner by using local measurements from exteroceptive sensing, e.g., ultrasonic, laser scanner, and vision.

7. ACKNOWLEDGEMENTS

This work was financially supported by National Science and Technology Development Agency (NSTDA), Thailand.

8. REFERENCES

1. Campion, G., Bastin, G. and D'Andrea-Novel, B., "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots", ***IEEE Transactions on Robotics and Automation***, Vol. 12, No. 1, (1996), 47-62.
2. Korayem, M.H., Peydaie, P. and Azimirad, V., "Investigation on the effect of different parameters in wheeled mobile robot error", ***International Journal of Engineering-Transactions A: Basics***, Vol. 20, No. 2, (2007), 195-210.
3. Korayem, M.H. and Bani Rostam, T., "Experimental analysis for measuring errors in wheeled mobile robots", ***International Journal of Engineering-Transactions A: Basics***, Vol. 18, No. 2, (2005), 115-133.
4. Soetanto, D., Lapierre, L. and Pascoal, A., "Adaptive, non-singular path-following control of dynamic wheeled robots", in Decision and Control, 2003. Proceedings. 42nd IEEE Conference on, IEEE. Vol. 2, (2003), 1765-1770.
5. Aguiar, A.P., Dacic, D.B., Hespanha, J.P. and Kokotovic P., "Path-following or reference-tracking? An answer relaxing the limits to performance", in IFAC/EURON Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal, (2004).
6. Micaelli, A. and Samson, C., "Trajectory tracking for unicycle-type and two-steering-wheels mobile robots", (1993).
7. Płaskonka, J., "Different kinematic path following controllers for a wheeled mobile robot of (2, 0) type", ***Journal of Intelligent & Robotic Systems***, (2013), 1-18.
8. Altafini, C., "Following a path of varying curvature as an output regulation problem", (2002).
9. Conceicao, A.S., Oliveira, H.P., Sousa e Silva, A., Oliveira, D. and Moreira, A.P., "A nonlinear model predictive control of an omni-directional mobile robot", in IEEE International Symposium on Industrial Electronics, (2007), 2161-2166.
10. Petrov, P. and Dimitrov, L., "Nonlinear path control for a differential-drive mobile robot", ***Recent Journal***, Vol. 11, No. 1, (2010), 41-45.
11. Pishkenari, H.N., Mahboobi, S.H. and Alasty, A., "Trajectory tracking of a mobile robot using fuzzy logic tuned by genetic algorithm", ***International Journal of Engineering-Transactions A: Basics***, Vol. 19, No. 1, (2006), 95-104.
12. Aicardi, M., Casalino, G., Bicchi, A. and Balestrino, A., "Closed loop steering of unicycle like vehicles via lyapunov techniques", ***Robotics & Automation Magazine, IEEE***, Vol. 2, No. 1, (1995), 27-35.
13. Faulwasser, T., Kern, B. and Findeisen, R., "Model predictive path-following for constrained nonlinear systems", in Decision and Control, held jointly with 28th Chinese Control Conference. CDC/CCC. Proceedings of the 48th IEEE Conference on, IEEE, (2009), 8642-8647.
14. Kanjanawanishkul, K. and Zell, A., "Path following for an omnidirectional mobile robot based on model predictive control", in Robotics and Automation,. ICRA'09. IEEE International Conference on, IEEE, (2009), 3341-3346.
15. Oftadeh, R., Ghabcheloo, R. and Mattila, J., "A novel time optimal path following controller with bounded velocities for mobile robots with independently steerable wheels", in Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on, IEEE, (2013), 4845-4851.
16. Yu, S., Li, X., Chen, H. and Allgöwer, F., "Nonlinear model predictive control for path following problems", ***International Journal of Robust and Nonlinear Control***, (2013).
17. Vazquez, J. and Velasco-Villa, M., "Path-tracking dynamic model based control of an omnidirectional mobile robot", in 17th IFAC World Congress, (2008).
18. Huang, H.-C. and Tsai, C.-C., "Adaptive robust control of an omnidirectional mobile platform for autonomous service robots in polar coordinates", ***Journal of Intelligent and Robotic Systems***, Vol. 51, No. 4, (2008), 439-460.
19. Kalmar-Nagy, T., D'Andrea, R. and Ganguly, P., "Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle", ***Robotics and Autonomous Systems***, Vol. 46, No. 1, (2004), 47-64.
20. Allgower, F., Findeisen, R. and Nagy, Z.K., "Nonlinear model predictive control: From theory to application", ***Journal of the Chinese Institute of Chemical Engineers***, Vol. 35, No. 3, (2004), 299-315.
21. Mayne, D.Q., Rawlings, J.B., Rao, C.V. and Scokaert, P.O., "Constrained model predictive control: Stability and optimality", ***Automatica***, Vol. 36, No. 6, (2000), 789-814.
22. Indiveri, G., "Swedish wheeled omnidirectional mobile robots: Kinematics analysis and control", ***IEEE transactions on robotics***, Vol. 25, No. 1, (2009), 164-171.
23. Heinemann, P., "Cooperative multi-robot soccer in a highly dynamic environment, Citeseer, (2007).

Path Following and Velocity Optimizing for an Omnidirectional Mobile Robot

K. Kanjanawanishkul

Mechatronics Research Unit, Faculty of Engineering, Mahasarakham University, Khamriang, Kantarawichai, Mahasarakham, Thailand

P A P E R I N F O

Paper history:
Received 06 October 2014
Accepted in revised form 29 January 2015

Keywords:
Path Following Control
Robot Motion
Model Predictive Control
Omnidirectional Mobile Robots
Actuator Constraints

چکیده

در این مقاله، مسیر کنترل کننده یک ربات همراه چندوجهی (OMR) به گونه ای توسعه داده شده است که سرعت رو به جلو بهینه و محدودیت های سرعت محرک در نظر گرفته شده است. هر دو از طریق چارچوب کنترل پیش بینی مدل پیشنهادی (MPC) به دست آمده است. سرعت رو به جلو درتابع هدف گنجانده شده است، در حالی که اشباع محرک به عنوان محدودیت های سخت در نظر گرفته می شود. همانطور که در نتایج شبیه سازی نشان داده شده است، OMR می تواند به یک مسیر مرجع به طور موفقیت آمیز و امن همگرایی و پیروی کند. سرعت رو به جلو ربات به یک شخص مورد نظر نزدیک و زاویه گرایش مورد نظر در یک نقطه داده شده در مسیر به دست آمد، در حالی که محدودیت های محرک نقض نمی شود. علاوه بر این، برای نشان دادن اثربخشی چارچوب پیشنهادی ما، یک مقایسه با روشهای مرسوم که برای اتصال به محدودیت های محرک استفاده می شود، هدایت شده است. میانگین مربعات خطا (MSE)، خطای مربع جدایی ناپذیر (ISE)، و مسافت سفر به عنوان شاخص عملکرد استفاده شد. همانطور که در نتایج دیده می شود، استراتژی کنترل پیشنهاد بهتر از روشهای مرسوم عمل می کند. نسبت بین ترجمه و سرعت چرخشی بهینه سازی شد اگر چه محدودیت سرعت چرخشی و ترجمه ای از طریق زاویه گرایش OMR کوپل شده است.

***doi**: 10.5829/idosi.ije.2015.28.04a.07*