

Motion Control of Omnidirectional Mobile Platform for Path Following Using Backstepping Technique

Viet-Tuan Dinh*, Thinh Doan-Phuc*, Hoang Giang*, Hak Kyeong Kim*, Sea June Oh** and Sang Bong Kim*

*Department of Mechanical and Automotive Engineering, Pukyong National University, Busan, Korea

**Division of Mechatronics Engineering, Korea Maritime University, Busan, Korea

KEY WORDS: Backstepping control, Kinematic controller, Lyapunov function candidate, Omnidirectional mobile platform (OMP).

ABSTRACT: This paper proposes a controller design for an omnidirectional mobile platform (OMP) with three wheels using backstepping control. A kinematic model and dynamic model of the system are presented. Based on the dynamic modeling, a backstepping controller is designed to stabilize the OMP when following a desired path. The controller is designed based on a backstepping control theory. It includes two steps: first, a virtual state and a stability function are introduced. Second, Lyapunov functions for the system are chosen and an equation for the virtual control that makes the system stable is obtained. The system stability is guaranteed by the Lyapunov stability theory. The simulation and experimental results are presented to demonstrate the effectiveness of the proposed controller.

1. INTRODUCTION

Path following has been studied and applied in several applications such as in the fields of factories, ship navigations and hospitals. It must have high operational performance, be able to move freely in narrow spaces, and reach desired points precisely without complicated switchovers.

Many research results of omnidirectional mobile robot have been implemented for a path-following problem using various techniques. Li et al. (2007) issued the PD controller of OMP for sinusoidal path-following problem, but considered only the kinematic model which ignored the system dynamic. Vazquez and Villa (2007) also proposed computed-torque controller using PD feedback law of OMP based on its dynamic model for solving the circular path-tracking problem. However, they presented only simulation results. Nagy et al. (2004) proposed the real time control strategy to move OMP from initial point to target point with minimum time. Hung et al. (2009) proposed the OMP controller for trajectory tracking problem using proportional-differential sliding mode controller. Watanabe et al. (1998) considered omnidirectional mobile manipulator to use the computed torque control and the resolved acceleration control methods. However, they did not focus on real time control.

It should be noted that the conventional studies on the omnidirectional vehicles have been focused on the development of mechanisms or only on the analysis of kinematics, and hence there are little studies on the development of dynamic models and highly accurate control systems. Hung

et al. (2009) proposed the adaptive backstepping control algorithm for the kinematic model of the welding mobile manipulator. Hyeong et al. (2007) issued the adaptive backstepping controller for improving position accuracy of linear motor, but not for the OMP. With the main advantages of backstepping control technique such as fast response and strong robustness, several researches using a backstepping controller have done for wheeled mobile robots (Yagiz and Hacıoglu 2008; Sulaiman et al. 2010; Fierro and Lewis 1997; Zhang et al. 2003). They used a control scheme of integrating a kinematic modeling into a dynamic modeling. However, there are little researches about an omnidirectional mobile platform using a backstepping control technique.

This paper proposes a robust controller design for an omnidirectional mobile platform (OMP) using backstepping control. A kinematic modeling and a dynamic modeling of the system are presented. Based on the dynamic modeling, a backstepping controller is designed to stabilize the OMP to follow a desired path. The backstepping technique has two steps (Chen et al. 2009; Slotine and Li 1991): firstly, a virtual state and a stability function are introduced. Next, Lyapunov functions are chosen. A control law is obtained based on Lyapunov's direct method with backstepping technique that guarantee the system's stability. The simulation and experimental results are presented to demonstrate the effectiveness of the proposed controller.

2. SYSTEM MODELING

Figure 1 shows the configuration of the OMP. It consists of

Corresponding author Sang Bong Kim: 365, Sinseon-ro, Nam-Gu, Busan, Korea, +82-51-629-6158, kimsb@pknu.ac.kr.

a frame, three driving omnidirectional wheels, a control system, a camera sensor, etc.

Figure 2 shows the configuration for geometric model of the OMP. The three omnidirectional wheels are equally spaced at 120° from one another. They have the same radius denoted by r and are driven by DC motors. L is the distance from wheel's center to the geometric center C . $\mathbf{p}_{W_i} \in \mathcal{R}^{2 \times 1}$ ($i=1, 2, 3$) is a position vector of each wheel with respect to the moving coordinate frame CX_0Y_0 attached at point C on the OMP. $\mathbf{d}_{W_i} \in \mathcal{R}^{2 \times 1}$ is the drive direction vector of each wheel with respect to the global coordinate frame OXY . The posture vector of point C in the global coordinate frame is specified by $\mathbf{q}_C = [x_C \ y_C \ \theta_C]^T$. $\mathbf{P}_C = [x_C \ y_C]^T$ is defined as the position vector of point C with respect to the global coordinate frame. v_C and w_C are the linear velocity and the angular velocity of the OMP. v_i and w_i are the linear velocity and the angular velocity of the wheel of the OMP.

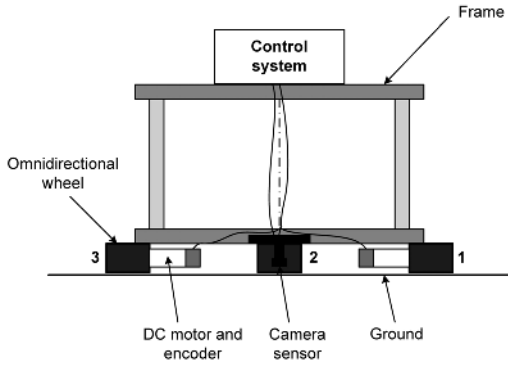


Fig. 1 Configuration of the OMP

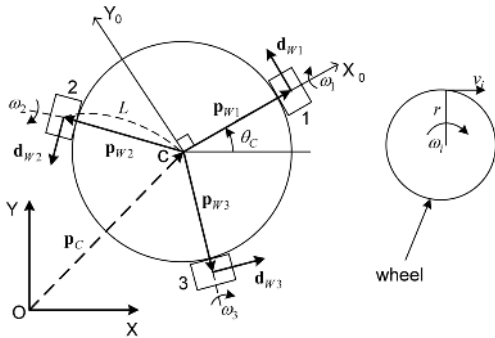


Fig. 2 Configuration for geometric model of the OMP

The OMP is modeled under the following assumptions:

- (1) The mass center of the OMP is coincident with the geometric center C of the OMP.
- (2) Kinematic's parameters such as wheel's radius r , distance L are known exactly.
- (3) The disturbance vector exerted on the OMP consists of motion surface friction and slip phenomena between the

wheel and the ground is ignored.

Kinematic modeling and dynamic modeling of the OMP are presented.

2.1 Kinematic modeling

The rotation matrix $\mathbf{R}(\theta_C)$ from the moving coordinate frame to the global coordinate frame is given by:

$$\mathbf{R}(\theta_C) = \begin{bmatrix} \cos(\theta_C) & -\sin(\theta_C) \\ \sin(\theta_C) & \cos(\theta_C) \end{bmatrix} \in \mathfrak{R}^{2 \times 2} \quad (1)$$

The position vectors $\mathbf{p}_{W1}, \mathbf{p}_{W2}, \mathbf{p}_{W3} \in \mathcal{R}^{2 \times 1}$ of the three wheels with respect to the moving coordinate frame attached at the mass center of the OMP can be obtained as follows:

$$\begin{aligned} \mathbf{p}_{W1} &= L \begin{bmatrix} 1 \\ 0 \end{bmatrix}, & \mathbf{p}_{W2} &= \mathbf{R}\left(\frac{2\pi}{3}\right)\mathbf{p}_{W1} = \frac{L}{2} \begin{bmatrix} -1 \\ \sqrt{3} \end{bmatrix}, \\ \mathbf{p}_{W3} &= \mathbf{R}\left(\frac{4\pi}{3}\right)\mathbf{p}_{W1} = -\frac{L}{2} \begin{bmatrix} 1 \\ \sqrt{3} \end{bmatrix} \end{aligned} \quad (2)$$

The drive direction vectors of the i^{th} wheel $\mathbf{d}_{W_i} \in \mathcal{R}^{2 \times 1}$ ($i=1, 2, 3$) are calculated as follows:

$$\begin{aligned} \mathbf{d}_{W1} &= \frac{1}{L} \mathbf{R}(\theta_C) \mathbf{p}_{W1}, & \mathbf{d}_{W1} &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \\ \mathbf{d}_{W2} &= -\frac{1}{2} \begin{bmatrix} \sqrt{3} \\ 1 \end{bmatrix}, & \mathbf{d}_{W3} &= \frac{1}{2} \begin{bmatrix} \sqrt{3} \\ -1 \end{bmatrix} \end{aligned} \quad (3)$$

A kinematic equation of a three-wheeled omnidirectional mobile platform (Sulaiman, 2010) can be expressed as follows:

$$\mathbf{z} = \frac{1}{r} \mathbf{H}^{-1} \dot{\mathbf{q}}_C \quad (4)$$

where $\mathbf{z} = [w_1 \ w_2 \ w_3]^T$ is the wheel angular velocity vector, and $\mathbf{H}^{-1} \in \mathcal{R}^{3 \times 3}$ matrix is given as:

$$\mathbf{H}^{-1} = \begin{bmatrix} -\sin \theta_C & \cos \theta_C & L \\ -\sin\left(\frac{\pi}{3} - \theta_C\right) & -\cos\left(\frac{\pi}{3} - \theta_C\right) & L \\ \sin\left(\frac{\pi}{3} + \theta_C\right) & -\cos\left(\frac{\pi}{3} + \theta_C\right) & L \end{bmatrix} \quad (5)$$

From Eq. (4), the velocity vector $\dot{\mathbf{q}}_C = [x_C \ y_C \ \theta_C]^T$ of the OMP is reduced into:

$$\dot{\mathbf{q}}_C = r \mathbf{H} \mathbf{z} \quad (6)$$

2.2 Dynamic modeling

The dynamic equation of OMP (Nagy et al. 2004) can be written as:

$$\begin{bmatrix} m\ddot{x}_C \\ m\ddot{y}_C \\ I\ddot{\theta}_C \end{bmatrix} + \frac{3\beta}{2} \begin{bmatrix} \dot{x}_C \\ \dot{y}_C \\ 2L^2\dot{\theta}_C \end{bmatrix} = \delta \mathbf{H}^{-T} \mathbf{u} \quad (7)$$

Eq. (7) is rewritten as follows:

$$\begin{cases} \ddot{x}_C = \frac{-3\beta}{2m} \dot{x}_C - \frac{\delta}{m} \sin\theta_C u_1 - \frac{\delta}{m} \sin\left(\frac{\pi}{3} - \theta_C\right) u_2 \\ \quad + \frac{\delta}{m} \sin\left(\frac{\pi}{3} + \theta_C\right) u_3 \\ \ddot{y}_C = \frac{-3\beta}{2m} \dot{y}_C + \frac{\delta}{m} \cos\theta_C u_1 - \frac{\delta}{m} \cos\left(\frac{\pi}{3} - \theta_C\right) u_2 \\ \quad - \frac{\delta}{m} \cos\left(\frac{\pi}{3} + \theta_C\right) u_3 \\ \ddot{\theta}_C = \frac{-3\beta}{I} L^2 \dot{\theta}_C + \delta L u_1 + \delta L u_2 + \delta L u_3 \end{cases} \quad (8)$$

I represents the moment of inertia and m is the mass of the mobile robot. The parameters, δ and β , are the motor characteristic coefficients, they are obtained from motor experiment. The control input vector $\mathbf{u}=[u_1 \ u_2 \ u_3]^T$ is a voltage vector applied to each motor. $\mathbf{x}_1=\mathbf{q}_C=[x_C \ y_C \ \theta_C]^T$ is defined as a posture vector and the velocity vector is defined as $\mathbf{x}_2=\dot{\mathbf{q}}_C=[\dot{x}_C \ \dot{y}_C \ \dot{\theta}_C]^T$.

Eqs. (6) and (8) can be written into:

$$\begin{cases} \dot{\mathbf{x}}_1 = \mathbf{x}_2 = \dot{\mathbf{q}}_C = \mathbf{r} \mathbf{H} \mathbf{z} \\ \dot{\mathbf{x}}_2 = \mathbf{A} \mathbf{x}_2 + \mathbf{B} \mathbf{u} \end{cases} \quad (9)$$

where:

$$\mathbf{A} = \text{diag} \left[\frac{-3\beta}{2m} \quad \frac{-3\beta}{2m} \quad \frac{-3\beta}{I} L^2 \right] \quad (9a)$$

$$\mathbf{B} = \begin{bmatrix} -\frac{\delta}{m} \sin\theta_C & -\frac{\delta}{m} \sin\left(\frac{\pi}{3} - \theta_C\right) & \frac{\delta}{m} \sin\left(\frac{\pi}{3} + \theta_C\right) \\ \frac{\delta}{m} \cos\theta_C & -\frac{\delta}{m} \cos\left(\frac{\pi}{3} - \theta_C\right) & -\frac{\delta}{m} \cos\left(\frac{\pi}{3} + \theta_C\right) \\ \delta L & \delta L & \delta L \end{bmatrix} \quad (9b)$$

3. DESIGN OF CONTROLLER

In this section, the backstepping controller design is proposed. The backstepping methodology is a sort of recursive design algorithm. Since the strict-feedback system is formulated, the Lyapunov direct method is employed to stabilize the feed-back loop.

***Step 1:** A posture tracking error vector $\mathbf{e}_p=[e_x \ e_y \ e_\theta]^T$ is defined as follows:

$$\mathbf{e}_p = \mathbf{x}_1 - \mathbf{x}_r \quad (10)$$

where $\mathbf{x}_r=[x_r \ y_r \ \theta_r]^T$ is defined as a reference posture vector

on the reference path.

Figure 3 shows the configuration of the posture tracking errors.

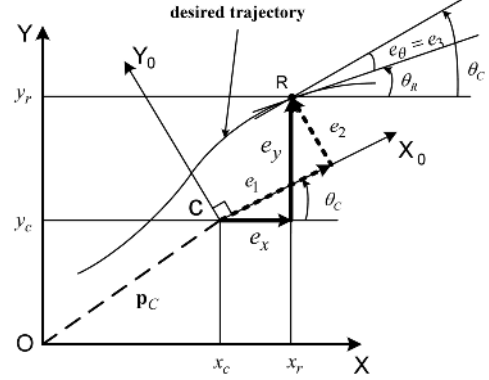


Fig. 3 Configuration of the posture tracking errors $e_x \ e_y \ e_\theta$

In applying the backstepping technique, a backstepping error vector $\mathbf{e}_b=[e_{b1} \ e_{b2} \ e_{b3}]^T$ is defined as:

$$\mathbf{e}_b = \mathbf{x}_2 - \boldsymbol{\alpha} \quad (11)$$

where \mathbf{x}_2 is chosen as a virtual control input and a stability function vector $\boldsymbol{\alpha}$ for \mathbf{x}_2 is chosen as follows:

$$\boldsymbol{\alpha} = -\mathbf{K}_1 \mathbf{e}_p + \dot{\mathbf{x}}_r \quad (12)$$

where \mathbf{K}_1 is a positive definite matrix.

From Eqs. (9)~(12), the time derivative of \mathbf{e}_p is given into:

$$\begin{aligned} \dot{\mathbf{e}}_p &= \dot{\mathbf{x}}_1 - \dot{\mathbf{x}}_r = \mathbf{x}_2 - \dot{\mathbf{x}}_r = \mathbf{e}_b + \boldsymbol{\alpha} - \dot{\mathbf{x}}_r \\ &= \mathbf{e}_b - \mathbf{K}_1 \mathbf{e}_p \end{aligned} \quad (13)$$

The first Lyapunov function candidate associated with the tracking error is chosen as:

$$V_1 = \frac{1}{2} \mathbf{e}_p^T \mathbf{e}_p \quad (14)$$

The derivative of the Lyapunov function candidate is evaluated as:

$$\dot{V}_1 = \mathbf{e}_p^T \dot{\mathbf{e}}_p = \mathbf{e}_p^T (\mathbf{e}_b - \mathbf{K}_1 \mathbf{e}_p) = \mathbf{e}_p^T \mathbf{e}_b - \mathbf{e}_p^T \mathbf{K}_1 \mathbf{e}_p \quad (15)$$

Eq. (15) cannot guarantee $\dot{V}_1 \leq 0$ when $\mathbf{e}_p \neq 0$ and $\mathbf{e}_b \neq 0$. Thus, the second Lyapunov function candidate must be considered.

***Step 2:** According to step 1, the second Lyapunov function candidate is chosen as:

$$V_2 = V_1 + \frac{1}{2} \mathbf{e}_b^T \mathbf{e}_b \quad (16)$$

From Eq. (11), the time derivative of \mathbf{e}_b is written as:

$$\dot{\mathbf{e}}_b = \dot{\mathbf{x}}_2 - \dot{\boldsymbol{\alpha}} \quad (17)$$

Substituting Eqs. (9), (12) and (13) into Eq. (17) yields:

$$\begin{aligned}\dot{\mathbf{e}}_b &= \mathbf{A}\mathbf{x}_2 + \mathbf{B}\mathbf{u} + \mathbf{K}_1\dot{\mathbf{e}}_p - \ddot{\mathbf{x}}_r \\ &= \mathbf{A}\mathbf{x}_2 + \mathbf{B}\mathbf{u} + \mathbf{K}_1(\mathbf{e}_b + \boldsymbol{\alpha} - \dot{\mathbf{x}}_r) - \ddot{\mathbf{x}}_r \\ &= \mathbf{A}\mathbf{x}_2 + \mathbf{B}\mathbf{u} + \mathbf{K}_1\mathbf{e}_b + \mathbf{K}_1\boldsymbol{\alpha} - \mathbf{K}_1\dot{\mathbf{x}}_r - \ddot{\mathbf{x}}_r.\end{aligned}\quad (18)$$

The derivative of the second Lyapunov function candidate is evaluated as:

$$\begin{aligned}\dot{V}_2 &= \dot{V}_1 + \mathbf{e}_b^T \dot{\mathbf{e}}_b \\ &= \mathbf{e}_p^T \dot{\mathbf{e}}_p - \mathbf{e}_p^T \mathbf{K}_1 \dot{\mathbf{e}}_p + \mathbf{e}_b^T (\mathbf{A}\mathbf{x}_2 + \mathbf{B}\mathbf{u} + \\ &\quad \mathbf{K}_1\mathbf{e}_b + \mathbf{K}_1\boldsymbol{\alpha} - \mathbf{K}_1\dot{\mathbf{x}}_r - \ddot{\mathbf{x}}_r) \\ &= -\mathbf{e}_p^T \mathbf{K}_1 \dot{\mathbf{e}}_p + \mathbf{e}_b^T (\mathbf{e}_p + \mathbf{A}\mathbf{x}_2 + \mathbf{B}\mathbf{u} + \\ &\quad \mathbf{K}_1\mathbf{e}_b + \mathbf{K}_1\boldsymbol{\alpha} - \mathbf{K}_1\dot{\mathbf{x}}_r - \ddot{\mathbf{x}}_r)\end{aligned}\quad (19)$$

So the feedback control law is chosen as follows:

$$\mathbf{u} = \mathbf{B}^{-1}(\mathbf{K}_1\dot{\mathbf{x}}_r + \ddot{\mathbf{x}}_r - \mathbf{e}_p - \mathbf{A}\mathbf{x}_2 - \mathbf{K}_1\mathbf{e}_b - \mathbf{K}_1\boldsymbol{\alpha} - \mathbf{K}_2\mathbf{e}_b)\quad (20)$$

where \mathbf{K}_2 is a positive definite matrix.

Substituting Eq. (20) into Eq. (19) yields:

$$\dot{V}_2 = -\mathbf{e}_p^T \mathbf{K}_1 \dot{\mathbf{e}}_p - \mathbf{e}_b^T \mathbf{K}_2 \mathbf{e}_b\quad (21)$$

$\dot{V}_2 \leq 0$ is guaranteed negative.

Now the time derivative of \mathbf{e}_b is rewritten as:

$$\begin{aligned}\dot{\mathbf{e}}_b &= \mathbf{A}\mathbf{x}_2 + \mathbf{B}\mathbf{u} + \mathbf{K}_1\mathbf{e}_b + \mathbf{K}_1\boldsymbol{\alpha} - \mathbf{K}_1\dot{\mathbf{x}}_r - \ddot{\mathbf{x}}_r \\ &= -\mathbf{e}_p - \mathbf{K}_2\mathbf{e}_b\end{aligned}\quad (22)$$

From Eqs. (13) and (22), the tracking error dynamic in the closed loop system is:

$$\begin{cases} \dot{\mathbf{e}}_p = \mathbf{e}_b - \mathbf{K}_1\dot{\mathbf{e}}_p \\ \dot{\mathbf{e}}_b = -\mathbf{e}_p - \mathbf{K}_2\mathbf{e}_b \end{cases} \Leftrightarrow \begin{pmatrix} \dot{\mathbf{e}}_p \\ \dot{\mathbf{e}}_b \end{pmatrix} = \begin{bmatrix} -\mathbf{K}_1 & \mathbf{I} \\ -\mathbf{I} & -\mathbf{K}_2 \end{bmatrix} \begin{pmatrix} \mathbf{e}_p \\ \mathbf{e}_b \end{pmatrix}\quad (23)$$

The block diagram for the proposed control algorithm of the OMP is shown in Figure 4.

4. ERROR DETECTING SCHEME

To achieve the controller, the errors have to be detected. The track is a black smooth curve line as a desired path marked on a floor. A camera sensor combined with an angular sensor is used to detect a posture tracking error vector. A camera sensor of module OV7620 is used to detect auxiliary posture errors e_1, e_2, e_3 . It is mounted in such a way as to capture directly an image of the tracking line under the platform of the OMP. The scheme for the auxiliary posture errors is shown in Fig. 5 (Hung et al. 2010).

The auxiliary posture errors can be expressed by:

$$\begin{cases} e_1 = (117 - m_x) \sin e_3 \\ e_2 = (117 - m_x) \cos e_3 \\ e_3 = \arctan[(x_A - x_B) / (y_A - y_B)]. \end{cases}\quad (24)$$

A CMPS03-Compass Module is an angular sensor for measuring a heading angle. The compass module uses the Philips KMZ51 magnetic field sensor, which is sensitive enough to detect the Earth's magnetic field. The heading angle θ_c is received through a generated PWM signal or I2C interface.

From the auxiliary posture errors measured by camera sensor and the heading angle θ_c measured by compass module, the following posture tracking error vector $\mathbf{e}_p = [e_x \ e_y \ e_\theta]^T \in \mathcal{R}^{3 \times 1}$ in Fig. 3 can be calculated by:

$$\begin{cases} e_x = e_1 \cos \theta_c - e_2 \sin \theta_c \\ e_y = e_1 \sin \theta_c + e_2 \cos \theta_c \\ e_\theta = e_3. \end{cases}\quad (25)$$

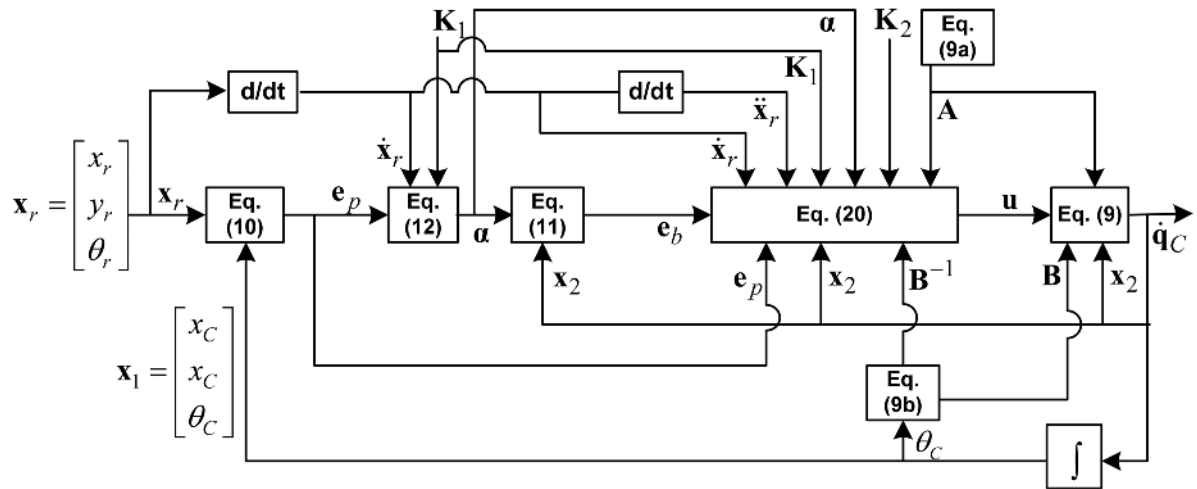


Fig. 4 Block diagram of controller

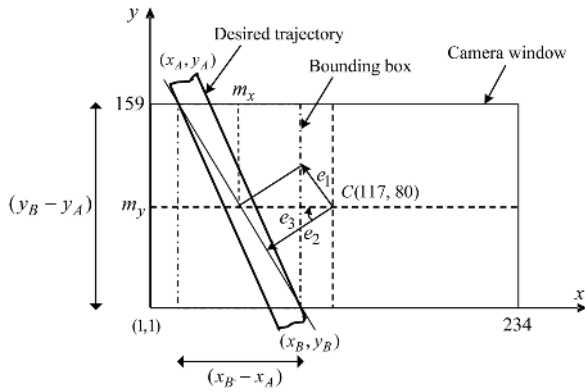


Fig. 5 Auxiliary error detecting scheme

5. CONTROL SYSTEM ARCHITECTURE

A control system is developed based on PIC18F452 type of microcontrollers which are operated with the clock frequency 40MHz. A hardware configuration of the proposed control system using four PIC18F452's is shown in Figure 6. One PIC18F452 is used as a mater unit, and other PIC18F452's are used as slave units. The control algorithm designed in section 3 is embedded into PIC18F452 for controlling the DC motors of three wheels of the OMP. The slave unit integrates three PIC18F452's with LMD18200 motor drivers for control of the DC motors. The servo controller of the slave unit can perform a complete servo operation with a closed loop feedback control using an encoder for velocity control of three wheels. The master unit functions as the main controller, that is, to receive the auxiliary posture error vector and the heading angle of the OMP using the camera sensor and the angular sensor and, in turn, to send the commands to the slave controllers via I2C communication, respectively.

The master unit can be used to interface other devices such as display and keypad devices for manual control. The sampling time of the proposed control system in experiment is about 75ms. The prototype of the experimental OMP with three degrees-of-freedom (DOFs) is shown in Figure 7.

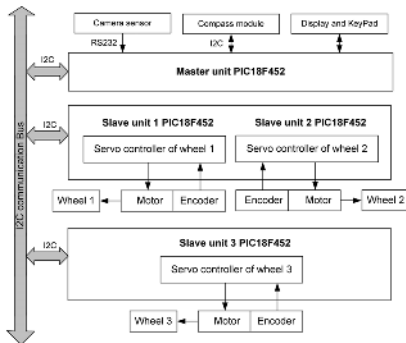


Fig. 6 Control architecture of hardware system of the OMP

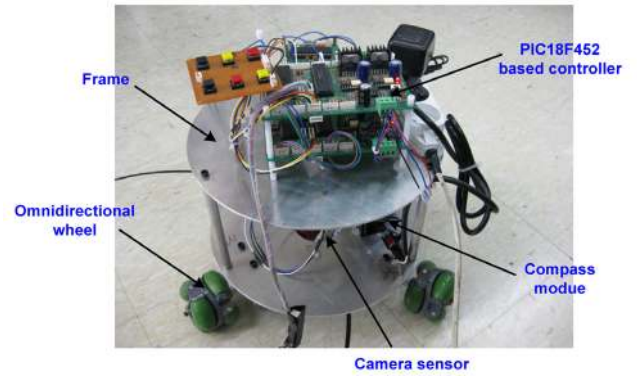


Fig. 7 Prototype of the experimental OMP

6. SIMULATION AND EXPERIMENTAL RESULTS

To verify the effectiveness of the proposed controller, simulations have been done for the OMP be stabilized when it tracks the desired path. The desired path is the circular line with radius $R = 0.3$ m and center $(X, Y) = (0 \text{ m}, 0 \text{ m})$ as shown in Figure 8. The constant desired velocity of the OMP is $v_R = 0.0127$ m/s The sampling time is 1 ms. Table 1 and Table 2 show the numerical parameter values and the initial values of the state variables for simulation.

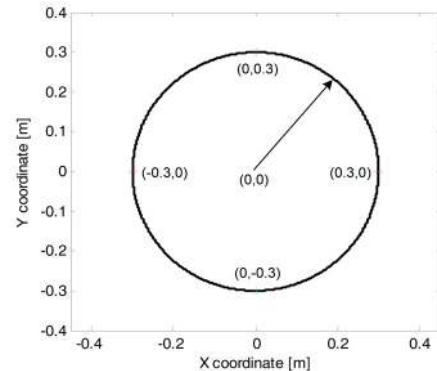


Fig. 8 Desired path

Table 1 Numerical parameter values

Parameters	Values	Units
L	0.18	[m]
m	4.5	[kg]
I	0.12	[kgm ²]
K_1	Diag ([1.2 1.2 1.9])	
K_2	Diag ([4.3346 4.3346 4.9617])	
δ	0.8	
β	0.01	
T	10	[ms]

The simulation results during about 5 seconds at the beginning time and the full time for the system are shown in Figs. 9-12 with the sample time 1 ms. Figure 9 shows the

performance of tracking a desired circle path. The backstepping controller makes the OMP track the reference path well. Figure 10 shows the posture tracking error vector. It shows that the errors go to zero from about 4 seconds. Figure 11 shows the backstepping error vector. The backstepping error vector also goes to zero from 4 seconds and it keeps zero values for full time. The angular velocities of three wheels are shown in Figure 12. It shows that the angular velocities have quick changes at the first time and converge to the constant values $w_1 = 0.2056$ rad/s, $w_2 = -0.0929$ rad/s, $w_3 = 0.05120$ rad/s, from about 4 seconds for tracking the circular line in counterclockwise. The simulation and experimental results show that the OMP can track a given desired path with the desired constant velocity.

Table 2 Initial values for simulation

Parameters	Values	Units
x_r	0.3	[m]
y_r	0	[m]
θ	90	[deg]
x_c	0.289	[m]
y_c	-0.01	[m]
θ_c	101	[deg] ≈ 1.761 (rad)

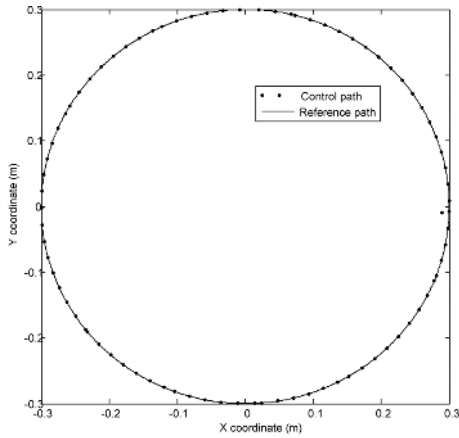


Fig. 9 Simulation results of tracking a circle

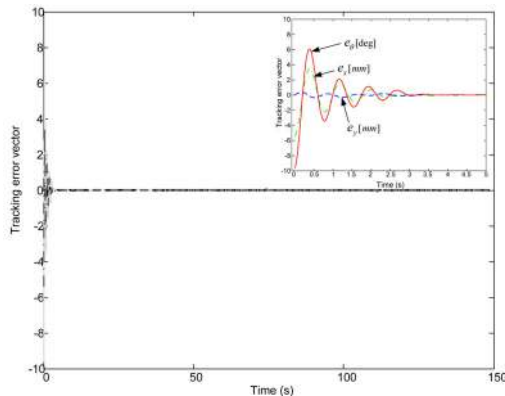


Fig. 10 Posture tracking error vector e_p for the full time

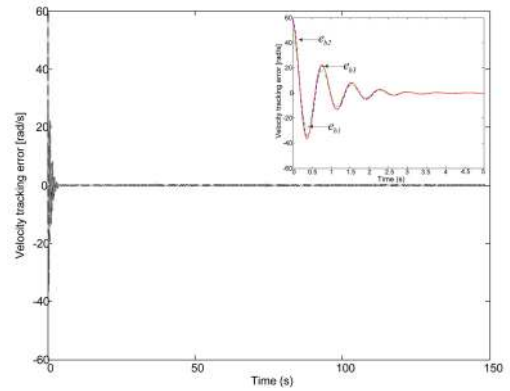


Fig. 11 Backstepping error vector e_b for the full time

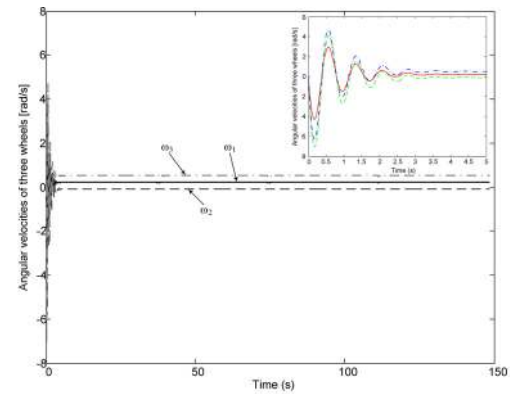


Fig. 12 Angular velocities of three wheels

The simulation and experimental results for the tracking trajectory are shown in Figs. 13-17. The experimental results are data filtered by low pass filter. Figure 13-15 show the simulation and experimental results for the posture tracking error vector for the full time. They show that the experimental results of the posture tracking errors in X, Y directions and rotation e_x , e_y , e_θ are bounded along the simulation results within ± 2.96 mm, ± 2.17 mm, ± 2.89 deg, respectively. Figure 16 shows the simulation and experimental results for the velocity of the OMP. The experimental velocity of the OMP is bounded from about 0.0115 m/s to 0.0159 m/s along the simulation linear velocity 0.0127 m/s. The experimental angular velocity of the OMP is bounded from about 0.0407 rad/s to 0.0482 rad/s along the simulation angular velocity 0.0423 rad/s as shown in Figure 17. The simulation and experimental results show the effectiveness of the proposed tracking controller.

7. CONCLUSIONS

In this paper, the backstepping tracking controller based on Lyapunov method is proposed for OMP to track a desired path. The Lyapunov functions are used to guarantee the stability of the control system. The proposed control laws make the

posture tracking error vector and the angular velocity tracking error vector converge to zero asymptotically. To implement the designed tracking controller, a control system is developed based on PIC18F452. A scheme for measuring the posture tracking error vector using camera sensor combined with an angular sensor is proposed.

The simulation and experimental results show that the backstepping controller could track the desired trajectories well. These results demonstrate the effectiveness of the proposed backstepping controller.

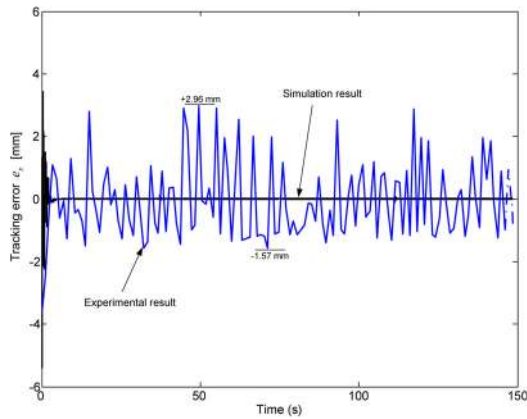


Fig. 13 Simulation and experimental results of tracking error e_x

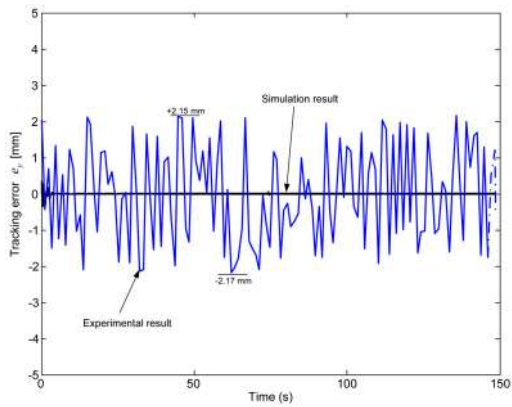


Fig. 14 Simulation and experimental results of tracking error e_y

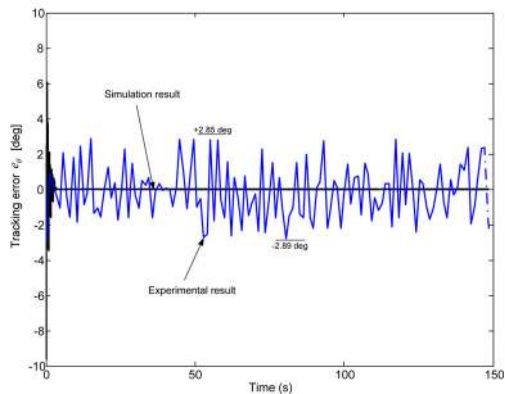


Fig. 15 Simulation and experimental results of tracking error e_θ

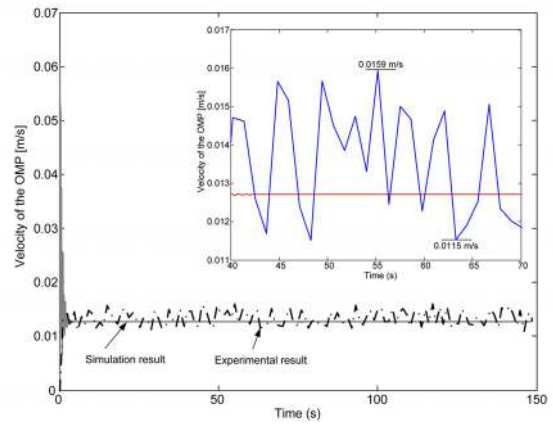


Fig. 16 Simulation and experimental results of velocity of the OMP

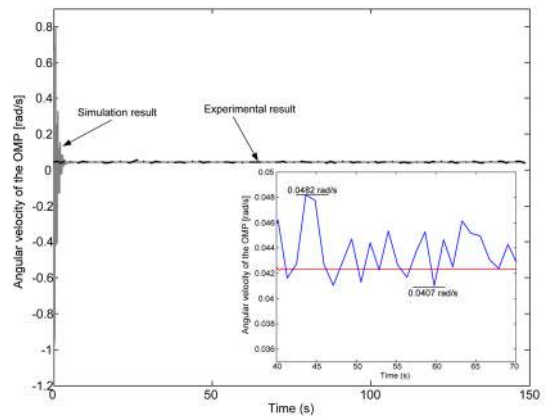


Fig. 17 Simulation and experimental results of angular velocity of the OMP

REFERENCES

Chen, S.H., Juang, J.C. and Su, S.H. (2009). "Backstepping Control with Sum of Squares Design for Omni-directional Mobile Robots," ICROS-SICE International Joint Conference 2009, Fukuoka International Congress Center, Japan, pp 545-550.

Fierro, R. and Lewis, F.L. (1997). "Control of a Nonholonomic Mobile Robot: Backstepping Kinematics into Dynamics," Journal of Robotic Systems, Vol 14, No 3, pp 149-163.

Hung, N., Kim D.H., Kim, H.K. and Kim S.B. (2009). "Tracking Control of Omnidirectional Mobile Robot Using Sliding Mode Control Technique," Proceeding of the 2009 International Symposium on Advanced Engineering, Busan, Korea, pp 42-45.

Hung, N., Im, J.S., Jeong, S.K., Kim, H.K. and Kim, S.B. (2010). "Design of a Sliding Mode Controller for an Automatic Guided Vehicle and Its Implementation," International Journal of Control, Automation, and Systems, Vol 8, No 1, pp 81-90.

- Hyeong, S.J., Jo, H.K., Sun, I. and U, I.J. (2007). "A Design of Adaptive Backstepping Controller for Improving Position Accuracy of Linear Motor-based Container Transportation System with Dynamic," *Journal of Ocean Engineering and Technology*, Vol 29, No 5, pp 548-555.
- Nagy, T.K., D'Andrea, R. and Ganuly, P. (2004). "Near-optimal Dynamic Trajectory Generation and Control of an Omnidirectional Vehicle," *Robotics and Autonomous Systems*, Vol 47, pp 47-64.
- Li, X., Wang, M. and Zell, A. (2007). "Dribbling Control of Omnidirectional Soccer Robots," 2007 IEEE International conf. on robotics and Automation, Italy, pp 2623-2628.
- Slotine, J.J.E. and Li, W. (1991). *Applied Nonlinear Control*, Prentice-Hall International Inc.
- Sulaiman, N.A.B., Yusop, A.B.M. and Samsudin, S.I.B. (2010). "Solving Tracking Problem of a Nonholonomic Wheel Mobile Robot Using Backstepping Technique," *Journal of Mechanical Engineering and Technology*, Vol 2, No 1, pp 85-92.
- Vazquez, J.A. and Villa, M.V. (2007). "Computed-Torque Control of an Omnidirectional Mobile Robot," *ICEEE 2007 4th International conf. on Electrical and Electronics Engineering, Mexico*, pp 274-277.
- Watanabe, K., Shiraiishi, Y., Tzafestas, S.G., Tang J. and Fukuda, T. (1998). "Feedback Control of an Omnidirectional Autonomous Platform for Mobile Service Robots," *Journal of Intelligent and Robotic Systems*, Vol 22, pp 315-330.
- Yagiz, N. and Hacioglu, Y. (2008). "Backstepping Control of a Vehicle with Active Suspensions," *Control Engineering Practice*, Vol 16, pp 1457-1467.
- Zhang, Y., Fidan and Ioannou, P.A. (2003). "Backstepping Control of Linear Time-Varying Systems With Known and Unknown Parameters," *IEEE Transactions on Automatic Control*, Vol 48, No 11, pp 1908-1925.

2011년 6월 13일 원고 접수

2011년 8월 8일 심사 완료

2011년 10월 17일 게재 확정