Self-Localization and Navigation of Holonomic Mobile Robot using Omni-Directional Wheel Odometry

Jiraphan Inthiam and Chirdpong Deelertpaiboon
Automation Engineering, Department of Instrumentation and Electronics Engineering
Faculty of Engineering, King Mongkut's University of Technology North Bangkok
Bangkok, Thailand
i.jiraphan@gmail.com, chirdpongd@gmail.com

***Abstract*—This paper proposes a simple but effective self-localization and navigation algorithm for the omni-directional mobile robot equipped with three driving and three odometry wheels. Not only can this technique locates the robot's position, but it can also align its heading with respect to the selected target while navigating along the pre-defined trajectory. With a certain configuration of odometry omni-directional wheels, rotational readings from three encoders attached to these odometry wheels provide linear and angular velocities of the mobile robot. Integrating these velocities would result in position and heading of the mobile robot. As a result, the mobile robot becomes a complete holonomic system. Furthermore, navigation algorithm is illustrated for two cases namely point-to-point and multi-point schemes. Three experimental results are conducted to verify the effectiveness of this algorithm.**

Keywords—self-localization; navigation algorithm; omni-directional wheel robot

I. INTRODUCTION

One of the versatile technique for navigating the robot on a flat surface is to utilize omni-directional wheels. With its capability of simultaneous translation and rotation movement, it attracts many researcher to build the mobile robot based on omni-directional wheel platform. Since then, there are various numbers of omni-directional wheels platform under researched such as three wheels [1,2], four wheels [3], or six wheels [4]. However, omni-drive requires a minimum of three wheels to follow any planar trajectory [5]. In other words, three or more omni-directional wheels robot becomes holonomic.

Self-localization is one of the most important features for mobile robot application. Without understanding its location, navigating robot is impossible. There have been many different approaches to locate robot's position such as visual based localization, dead-reckoning system, or probabilistic method utilizing visual and internal sensors. For example, authors in [6] proposed a visual reference scan approach in simultaneous localization and mapping (SLAM) design based on omni-directional camera. The robot required feature detection and comparison with the reference frame. In other words, the robot depended largely on environmental setup. Moreover, the complexity and time usage of visual based method is high. On the other hand, conventional dead-reckoning method is prone to accumulated error over time if there is no correction for slippage effect [7]. Hence, authors in [8] introduced the fusion of odometry system with omni-directional camera and also decoupled the odometry wheels and the driving wheels to decrease the slippage effect. The results of self-localization was improved.

The ultimate goal of controlling a mobile robot is not only capable of locating robot's position, it must also be able to navigate the robot along any defined path. Moreover, avoiding obstacle is an added feature that it must have. In [9], the authors proposed a navigation technique for an holonomic mobile robot which equipped with optical tracking device (OTD).

Aiming at simple self-localization and navigation algorithm, this paper proposed an omni-directional mobile robot with three driving wheels and three odometry wheels which are completely decoupled from each other. This results in minimal slippage. In section II, the overview of the robot architecture is explained in detail and followed by self-localization algorithm in section III. Section IV discusses the navigation algorithm which is described in two cases namely point-to-point navigation and multi-point navigation. Then, the overall system combining the self-localization unit and the navigation unit is explained in section V. Three experimental results and mean error for each case are demonstrated in Section VI.

II. MOBILE ROBOT ARCHITECTURE

This section explains the structure of the omni-directional mobile robot platform. Fig. 1 shows the top view of the platform of three omni-directional wheels robot which is the minimum number of wheels for being a holonomic system. The length of each leg is measured at 340 mm with a height of 1000 mm.

978-1-4799-4075-2/14/$31.00 © 2014

Fig. 1. Three-wheel robot platform

Underneath the platform, there are two sets of wheels as shown in Fig. 2. Each of the outer omni-directional wheels has diameter of 100 mm and is connected directly to 24V dc motor. This is the main driving unit. On the other hand, the inner and smaller omni-directional wheels which have diameter of 50 mm are connected to a rotary incremental encoder. This is an odometry unit. Each of driving wheel and odometry wheel are aligned on the same radial line from the center of mass of the robot. This schematic separates the driving unit and the odometry system so that slippage effect of the driving wheels while trying to increase the speed has less effect on odometry measurement. Furthermore, to further reduce the slippage effect of the odometry wheels, friction between the odometry wheel and the driving surface is increased by attaching a spring as shown in Fig. 3. This spring will force the odometry wheel to maintain its contact with driving surface most of the time.

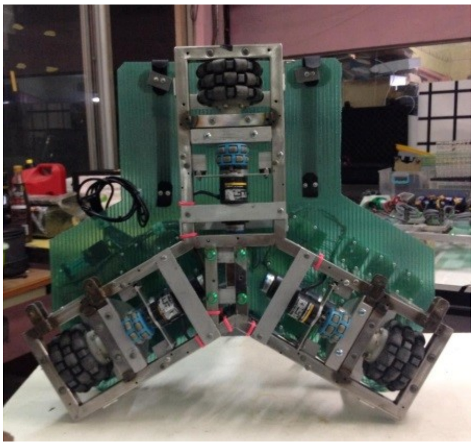

Fig. 2. Wheels configuration

Fig. 3. Slippage effect protection system

III. Self-Localization Algorithm

According to Fig. 4, the derivation of each omni-directional wheel velocities can be determined based on the analysis in [5]. The result is shown in (1).

$$V_w = V_r \cos\left(\theta_{ref} - \alpha\right) \tag{1}$$

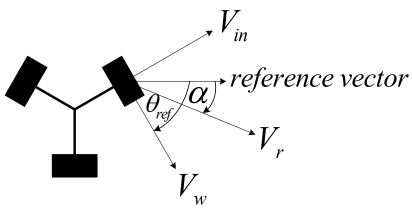

Fig. 4. Omnidirectional wheel velocity

If linear velocity of one wheel is forced to align with a reference axis in robot frame as shown in Fig. 5, odometry wheel linear velocities for all three can be obtained as shown in (2).

$$\begin{aligned} V_{w1} &= V_r \cos\alpha + L\omega \\ V_{w2} &= V_r\left(\frac{-1}{2}\cos\alpha + \frac{\sqrt{3}}{2}\sin\alpha\right) + L\omega \\ V_{w3} &= V_r\left(\frac{-1}{2}\cos\alpha - \frac{\sqrt{3}}{2}\sin\alpha\right) + L\omega \end{aligned} \tag{2}$$

where α is the angle of robot's velocity with respect to the reference axis, L is the distance from the center of mass to the center of the odometry wheel and ω is the angular velocity of the odometry wheel.

It is obvious from Fig. 4 that the robot velocity can be decomposed into two perpendicular velocities with respect to reference robot frame. Hence,

$$\begin{aligned} &\alpha \\ &\chi \end{aligned} \tag{3}$$

Furthermore, the relationship between linear and angular velocity of the general omni-directional wheel is

$$v = r\dot{\varsigma} \tag{4}$$

where r is the radius of the omni-directional wheel and $\dot{\varsigma}$ is the rotational velocity.

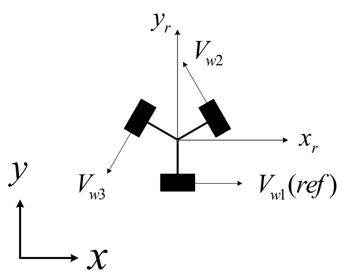

Fig. 5. Mobile robot in world coordinate

Hence, the odometry wheels angular velocity vector can be written as a function of linear and angular velocities of the mobile robot in matrix form as shown in (5).

$$\begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & L \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & L \\ \frac{-1}{2} & \frac{-\sqrt{3}}{2} & L \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \omega \end{bmatrix} \tag{5}$$

Moreover, it is simple to see that linear and angular velocities of the mobile robot can be obtained by solving inverse matrix in (5). The result is shown in (6).

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & \frac{-1}{3} & \frac{-1}{3} \\ 0 & \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{3}} \\ \frac{1}{3L} & \frac{1}{3L} & \frac{1}{3L} \end{bmatrix} \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \end{bmatrix} \tag{6}$$

As a consequence, the current position and heading of the mobile robot can be determined by integration. The result is

$$\begin{aligned} x &= r\left(\frac{2}{3}\phi_1 - \frac{1}{3}\phi_2 - \frac{1}{3}\phi_3\right) \\ y &= r\left(\frac{1}{\sqrt{3}}\phi_2 - \frac{1}{\sqrt{3}}\phi_3\right) \\ \theta_{robot} &= \frac{r}{3L}\left(\phi_1 + \phi_2 + \phi_3\right) \end{aligned} \tag{7}$$

Nonetheless, to obtain the angle ϕ_1, ϕ_2 and ϕ_3 from encoders, conversion must be done. Since the output of encoder has a unit of pulse per revolution, it must be multiplied with conversion factors which are given in (8) and (9) for position (x, y) and angle (θ_{robot}) respectively.

$$C_{position} = \frac{2\pi}{Encoder\ Resolution} \tag{8}$$

$$C_{heading} = \frac{360}{Encoder\ Resolution} \tag{9}$$

To self-localize the mobile robot, the global position must be determined. Using the heading of the robot, the position in robot frame can be transformed and updated in the global frame by (10).

$$\begin{bmatrix} x_{robot}^{new} \\ y_{robot}^{new} \end{bmatrix} = \begin{bmatrix} x_{robot}^{previous} \\ y_{robot}^{previous} \end{bmatrix} + \begin{bmatrix} \cos\theta_{robot} & -\sin\theta_{robot} \\ \sin\theta_{robot} & \cos\theta_{robot} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{10}$$

IV. NAVIGATION ALGORITHM

Navigation algorithm is divided into two categories. The first one is point-to-point navigation. The other one is multi-point navigation. Point-to-point navigation is suitable in case of no obstacle along the path. On the other hand, multi-point scheme is more appropriate for avoiding obstacle.

A. Point-to-point Navigation

The schematic of point-to-point navigation is shown in Fig. 6. The current robot position and goal points are defined by (x_r, y_r) and (x_G, y_G) respectively. Hence, the distance and the angle to reach the destination point can be determined by (11) and (12) accordingly.

$$S = \sqrt{(dx)^2 + (dy)^2} \tag{11}$$

$$\alpha = \tan^{-1}\left(\frac{dy}{dx}\right) \tag{12}$$

The distance is controlled by PD controller. Meanwhile, the heading of the mobile robot can be specified and calculated with respect to the x-axis in a global frame.

B. Multi-point Navigation

In case there is any need to specify via points so that the mobile robot can avoid obstacles. Point-to-point based navigation is not applicable. As a result, multi-point navigation is desired. One constraint for multi-point navigation is that the velocity at via point must be kept constant. The schematic for multi-point navigation is shown in Fig. 7.

Each via point will be bounded by some circle with radius *Err*. If the mobile robot, while approaching via point, touches the circumference of this circle, it is allowed to change its travelling course to the next via point. In this case, the trajectory will be a smooth curve and spend less time at via points. As soon as it reaches the last section of its trajectory, the mobile robot will switches to point-to-point navigation.

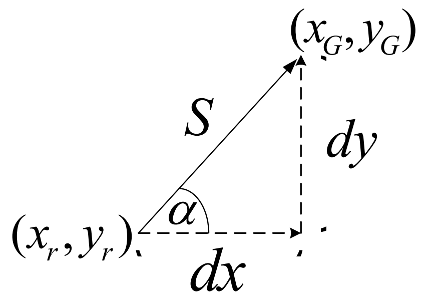

Fig. 6. Point-to-point navigation

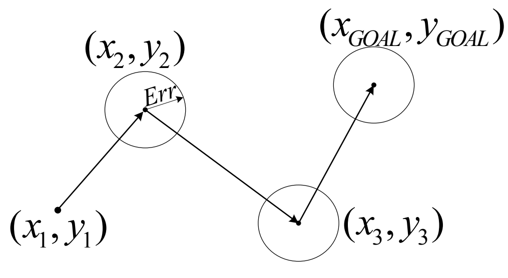

Fig. 7. Multi-point navigation

V. OVERALL SYSTEM DIAGRAM

Fig. 8 shows the overall system diagram. There are two inputs. The first one is the desired position $\begin{bmatrix} x & y \end{bmatrix}_{desired}^{T}$ which is fed to the point-to-point navigation in order to compute the distance, S, and the heading angle, α. These two variables will be used to determine the velocity of the mobile robot, V_r, and the heading direction, φ, On the other hand, another input is the desired angular position of the mobile robot, $\theta_{desired}$. This will determine rotation angle of the robot platform, ω. This navigation system will compute the linear velocity of each omni-directional wheel. As the robot moves to the destination point, the actual wheel angle is measured by the incremental rotary encoder. Then, it is fed back to the localization unit so that the current position, $\begin{bmatrix} x & y \end{bmatrix}_{Robot}^{T}$ and the real-time angular position, θ_{Robot}. are compared to the actual position and angular position.

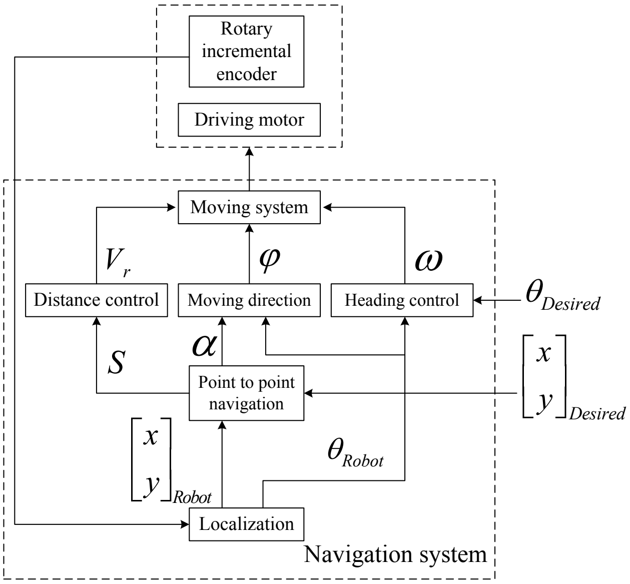

Fig. 8. Overall system diagram

VI. EXPERIMENTAL RESULTS

There are three different trajectories (square, circle, s-shape) to verify the effectiveness of the proposed self-localization and navigation schemes. The square path tracking has a dimension of 200x200 cm. The circle path tracking has the radius of 150 cm. Each half circle of s-shape path tracking has a radius of 150 cm. Not only the mobile robot must follow the specified trajectory, but it must also maintain its heading with respect to some point. In this case, the front of the mobile robot must face the center of curvature for each trajectory at all time. Square and circle path tracking have only one center of curvature. On the other hand, s-shape path presents two centers of curvatures. Hence, the mobile robot must rotate its body to face another center in s-shape tracking. The results of the experiments are shown in Fig 9, 10, and 11 for square, circle and s-shape respectively.

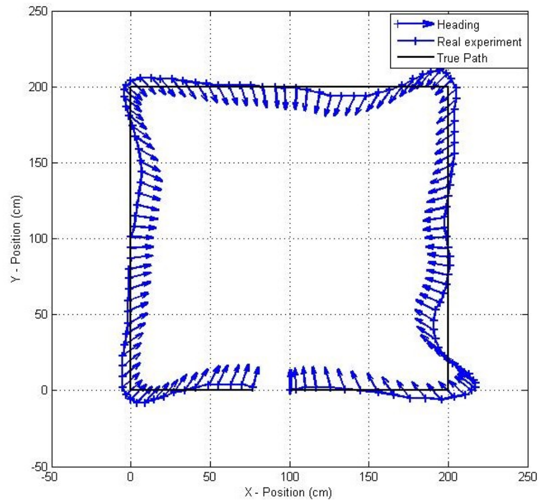

Fig. 9. Square path tracking

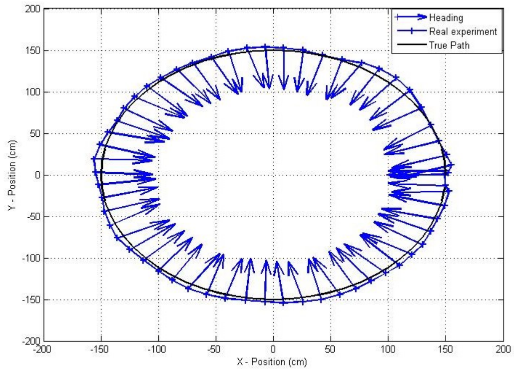

Fig. 10. Circle path tracking

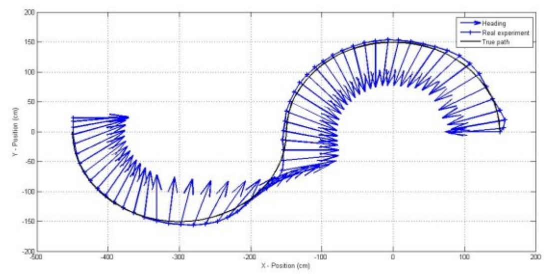

Fig. 11. S-shape path tracking

In each case, the mean error between the actual position which is determined by the self-localization method and the designed position is calculated. The result is shown in Table I. It is obvious that the largest error occur for square shape tracking especially at the corner of the square. This is because the robot tries to maintain its heading with the center of the square at which there is a sudden change in direction. On the other hand, the mobile robot can track circle and s-shape paths very well because it has a smooth change of heading direction along the trajectory.

TABLE I. MEAN ERROR

Path	Mean error (%)
Square	3.96
Circle	2.4
S-shape	2.27

VII. CONCLUSION

Overall, this paper proposes the self-localization and navigation algorithms for omni-directional wheel mobile robot. This is an example of holonomic system. By using separate set of driving wheels and odometry wheels, the robot can self-locate its position accurately. The slippage effect is minimized by attaching a spring to force the odometry wheels to keep contact with the driving surface. The mobile robot can be designed to move as point-to-point or multi-point schemes. Not only it can track the specified trajectory, but it can also maintain its heading with respect to designated point while traveling. This is a useful property especially for both indoor and outdoor robot applications. However, because of constant velocity constraint at via point, this mobile robot will have trouble keeping its heading and change of moving direction simultaneously. However, for any other curvature with smooth change of direction, this mobile robot can perform effectively.

REFERENCES

[1] W. Wang and D.J. Balkcom, "Analytical time-optimal trajectories for an omni-directional vehicle," in *Proc. of IEEE Int. conf. on Robotics and Automation*, pp. 4519 – 4524, May 2012.

[2] H. Kim an B. Kim, "Minimum-energy trajectory planning and control on a straight line with rotation for three-wheeled omni-directional mobile robot," in *Proc. of IEEE Int. conf. on Intelligent Robots and Systems*, pp. 3119 – 3124, October 2012.

[3] Y. Jia, X. Song and S.S.-D. Xu, "Modeling and motion analysis of four-mecanum wheel omni-directional mobile platform," in *Proc. of Int. Automatic Control Conference,"* pp. 328 – 333, December 2013.

[4] K.L. Moore and N.S. Flann, "Hierarchical task decomposition approach to path planning and control for an omni-directional autonomous mobile robot," in *Proc. of IEEE Int. Symposium on Intelligent Control/Intelligent Systems and Semiotics*, pp. 302 – 307, September 1999.

[5] M. Ashmore and N. Barnes, "Omni-drive robot motion on curved paths: the fastest path between two points is not a straight line," in *Proc. of Joint conf. on Advanced in Artificial Intelligence*, pp. 225 – 236, December 2002.

[6] K.T. Song and C.W. Wang, "Self-localization and control of an omni-directional mobile robot based on an omni-directional camera," in *Proc. of the Asian Control Conference*, pp. 899 – 904, August 2009.

[7] J. Borenstein, H.R. Everett, L. Feng, and D. Wehe, "Mobile robot positioning: sensors and techniques," *Journal of Robotic Systems*, Vol. 14, No. 4, 1997, pp. 231 – 249.

[8] H.R. Moballegh, P. Amini, Y. Pakzad, M. Hashemi, and M. Narmiani, "An improvement of self-localization for omnidirectional mobile robots using a new odometry sensor and omnidirectional vision," in *Proc. of Canadian conf. on Electrical and Computer Engineering*, pp. 2337 – 2340, May 2004.

[9] M. Mariappan, C.C. Wee, K. Vellian, and C.K. Weng, "A navigation methodology of an holonomic mobile robot using optical tracking device (OTD)," in *Proc. of IEEE Region 10 Annual Int. conf.*, pp. 1 – 6, January 2009.