# A Study on Human Robot Interaction Technology Using a Circular Coordinate System for the Remote Control of the Mobile Robot

Munsuck Jang[1], Eunghyuk Lee[1] and Sangbang Choi[2]

[1]*Dept. of Electronic Engineering, Korea Polytechnic Univ., Gyeonggi-Do Korea*
[2]*Dept. of Electronic Engineering, Inha Univ. Incheon, Korea*
*msjang@kpu.ac.kr, ehlee@kpu.ac.kr, sangbang@inha.ac.kr*

***Abstract***

*To improve the robot's locomotion performance, it definitely needs the HRI (Human Robot Interaction) technology, the location awareness technology, and the real time location correction technology. In this paper, an HRI method that uses a circular coordinate system is proposed. The proposed method can implement robot locomotion with variable speed and directions instead of conventional method that controls speed and direction separately. Also, the nine characteristics of proposed circular coordinate system are to be described and the principles of user interfaces using the circular coordinate system are to be analyzed. For the performance evaluation, we have used a mobile robot that can be controlled through the wireless LAN and can perform differential drive. In addition, we have configured and tested a smart phone environment that can control the robot. The evaluation results show that the circular coordinate system reduces the locomotion time with accuracy.*

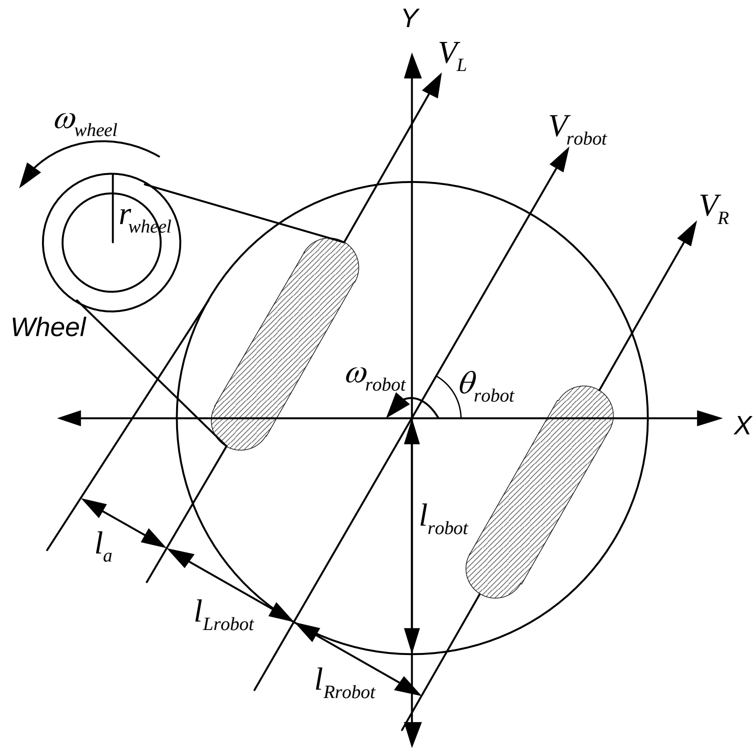
***Keywords:*** *Mobile Robot, Circular Coordinate System, HRI, Graphic User Interface, Mobile Robot, Robot Locomotion Coordinates*

## 1. Introduction

Mobile robot is a robot that enables a user to monitor the environments and control the robot in a safe place which helps a user to carry out the tasks more efficiently. Robot technology, network technology, and embedded device technology are essential ones to build a robot [1, 2]. Among them, the embedded device technology includes the HRI (Human Robot Interaction) technology which is to interact between the human and the robot.

The conventional way to control the mobile robot is through the GUI (Graphic User Interface) remote control system which uses a rectangular coordinate system and the computer interface devices, such as a keyboard, a mouse, and a joystick. And a controller has to implicitly assume the speed and the direction of the mobile robot when controlling the robot with the conventional method. To solve the problem, we propose the novel HRI algorithm based on a circular coordinate system that can reduce the locomotion distance and time by enabling the control of the speed and the direction of the mobile robot as the controller intended. For performance evaluation, the proposed circular coordinate system control method, the conventional direction control method with a constant speed and the conventional velocity-direction control method are compared in terms of the locomotion distance and the locomotion time while travelling twice along the square shaped track, which has 2m width and 2m height. The results show that the proposed method achieved 14.43% and 6.44% better performance than direction control method and velocity-direction control method in terms of the locomotion time, respectively. And in terms of the locomotion distance error, the proposed method shows 1.26%error which is meaningless, the direction control method 1.19%, and the velocity-direction control method -0.42% while travelling 16m.

## 2. Locomotion Model of the Robot




$\omega_{wheel}$ : Angular velocity of a wheel

$r_{wheel}$ : Radius of a wheel

$V_L$ : Velocity of a left wheel

$V_R$ : Velocity of a right wheel

$l_{Lrobot}$ : Distance between a left wheel and a center

$l_{Rrobot}$ : Distance between a right wheel and a center

$l_{robot}$ : Distance between chassis and a center

$l_a$ : Distance between chassis and a wheel

$V_{robot}$ : Velocity of a center

$\theta_{robot}$ : Angle of a center

$\omega_{robot}$ : Angular velocity of a center

**Figure 1. The Kinematics Model of a Robot with the Differential Drive Model**

It is essential to analyze the locomotion path and set the track for the locomotion of a robot. We adopted the differential drive model to analyze the path and set the track, guided from kinematics analysis which is to analyze the relationship between the control variables, location, and the velocity.

A robot with the differential drive model has its wheels on the same axle and each wheel is controlled independently by its own motor. An wheel has a radius of $r_{wheel}$ rotating with an angular velocity $\omega_{wheel}$. The distances between the center of the robot and a left or a right wheel are notated as $l_{Lrobot}$ and $l_{Rrobot}$, and the velocities of a left or a right wheel are notated as $V_L$ and $V_R$, as shown in Figure 1. The robot moves at a velocity of $V_{robot}$ and an angle of $\theta_{robot}$, counterclockwise from the x-axis.

The velocity of each wheel can be obtained through the equation (1) with $r_{wheel}$ and $\omega_{wheel}$, where $\omega_L$ stands for angular velocity of the left wheel and $\omega_R$ stands for angular velocity of the right wheel.

$$\begin{aligned} V_L &= r_{wheel} \cdot \omega_L \\ V_R &= r_{wheel} \cdot \omega_R \end{aligned} \quad (1)$$

The angular velocity of a center of the robot $\omega_{robot}$ and the velocity of a center of the robot $V_{robot}$ can be represented by the equations (2) and (3), in regard to the angular velocity of the left and the right wheels. Then, the angular velocity of the left and the right wheels, $\omega_L$ and $\omega_R$, can be represented by the equations (4) and (5). Here, $l_{Lrobot} = l_{Rrobot}$ as the distance between the left wheel and a center of the robot and the distance between the right wheel and a center of the robot are equal.

$$V_{robot} = \frac{V_L + V_R}{2} = r_{robot} \frac{\omega_L + \omega_R}{2} \quad (2)$$

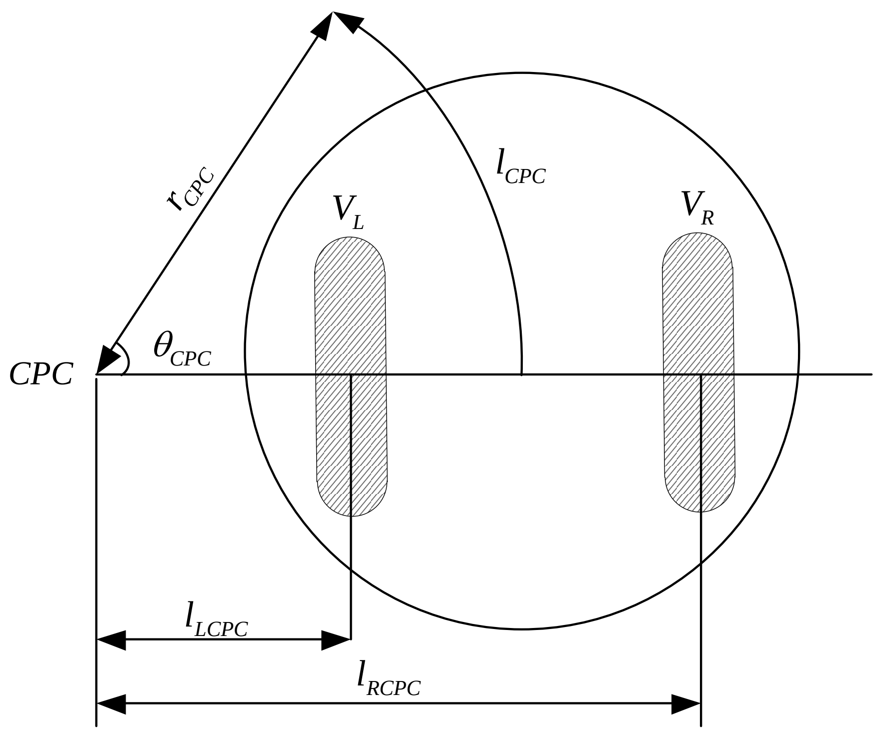
$$\omega_{robot} = \frac{1}{2l_{Lrobot}}(\omega_R - \omega_L) = \frac{1}{2l_{Rrobot}}(\omega_R - \omega_L) \quad (3)$$

$$\omega_L = V_{robot} - l_{Lrobot} \cdot \omega_{robot} \quad (4)$$

$$\omega_R = V_{robot} + l_{Rrobot} \cdot \omega_{robot} \quad (5)$$

In addition, the location of the robot on a rectangular coordinate system is denoted by the $x_{robot}(t)$ and the $y_{robot}(t)$, and the direction by the $\theta_{robot}(t)$. The $\dot{x}_{robot}(t)$ and the $\dot{y}_{robot}(t)$ are the linear speed components, the values from dividing $x_{robot}(t)$ and the $y_{robot}(t)$ with the unit time. And the $\dot{\theta}_{robot}(t)$ is the value from dividing $\theta_{robot}(t)$ with the unit time. Therefore, $\dot{x}_{robot}(t) = \cos\theta_{robot} \cdot V_{robot}$, $\dot{y}_{robot}(t) = \sin\theta_{robot} \cdot V_{robot}$ and $\dot{\theta}_{robot}(t) = \omega_{robot}$. It can be expressed by kinematics model on equation (6).

$$\begin{bmatrix} \dot{x}_{robot} \\ \dot{y}_{robot} \\ \dot{\theta}_{robot} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_{robot} \\ \omega_{robot} \end{bmatrix} \quad (6)$$




$l_{CPC}$ : Locomotion distance of the robot

$r_{CPC}$ : Turning radius of the robot

$\theta_{CPC}$ : Turning angle of the robot

$l_{LCPC}$ : Distance between a center of polar and a left wheel

$l_{RCPC}$ : Distance between a center of polar and a right wheel

**Figure 2. Center of Polar Proportional to the Velocity of Wheels**

A robot with the differential drive model changes its locomotion state according to the velocity of both wheels. For example, if the velocity of both wheels are 1m/s, the robot moves forward with a velocity of 1m/s according to the equations (2) and (3). And if the right wheel has the velocity of 1m/s and the left wheel has the velocity of -1m/s, then the robot rotates around at the same place. That is, a turning angle of the robot is determined by the velocity of both wheels. In case of $V_R = V_L$, which is $R = \infty$, the robot moves straight forward or backward. The robot turns its direction when $V_R \neq V_L$, and the robot rotates around at the same place when $V_R = -V_L$, which is $R = 0$. Especially, there exists a center of rotation when the robot moves around the corners or obstacles. In this paper, we define the center of rotation as the Center of Polar Coordinate (CPC). Figure 2 shows the CPC proportional to the velocity of both wheels. The velocity of both wheels $V_L$ and $V_R$ are proportional to $l_{LCPC}$ and $l_{RCPC}$, the distance between a CPC and wheels. And the instantaneous center of rotations is the intersection of extension of the wheels'axles. Therefore, the turning radius of the robot $r_{CPC}$ can be represented by equation (7), using the proportional expression of $l_{LCPC}$ and $l_{RCPC}$.

$$r_{CPC} = \left(\frac{V_R + V_L}{V_R - V_L}\right) l_{LCPC} \tag{7}$$

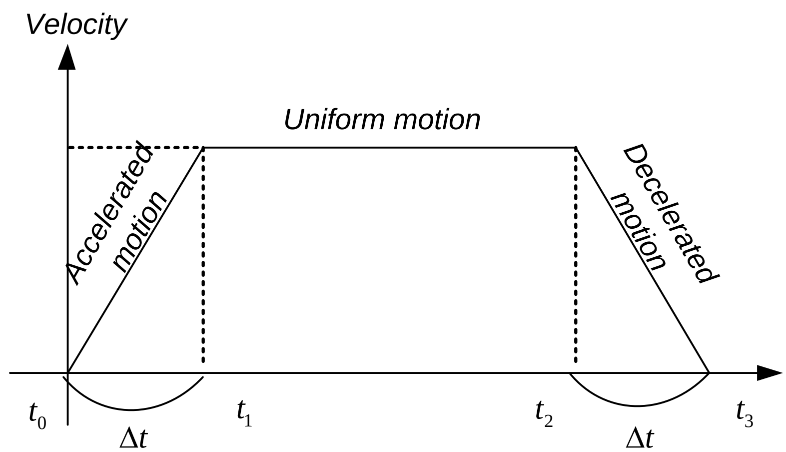



**Figure 3. Driving Method of a Robot**

With the conventional locomotion method, a robot has to go through the acceleration motion, the uniform motion, and the deceleration motion to get to the destination or turn its direction. Figure 3 shows the driving method of the robot; acceleration motion ($t_0 \rightarrow t_1$) which is to gain speed, uniform motion ($t_1 \rightarrow t_2$) which is to move with constant speed, and deceleration motion ($t_2 \rightarrow t_3$) which is to reduce speed, where acceleration motion and deceleration motion has the equal period $\Delta t$. With above, the locomotion distance of the robot $l_{CPC}$, the turning radius of the robot $r_{CPC}$, and the turning angle of the robot $\theta_{CPC}$ are represented by equations (8), (9), and (10).

$$l_{cpc} = \int \frac{V_L + V_R}{2} dt = \frac{1}{2} \int (V_L + V_R) dt = \frac{1}{2} r_{robot} \frac{\omega_L + \omega_R}{2} (t_3 - t_0 + t_2 - t_1) \tag{8}$$

$$r_{cpc} = \frac{L}{2} \frac{V_R + V_L}{V_R - V_L} = \frac{L}{2} \frac{\omega_R + \omega_L}{\omega_R - \omega_L} \tag{9}$$

$$\theta_{cpc} = \frac{l_{cpc}}{r_{cpc}} = \frac{\int (V_R + V_L) dt}{V_R + V_L} (V_R - V_L) \tag{10}$$

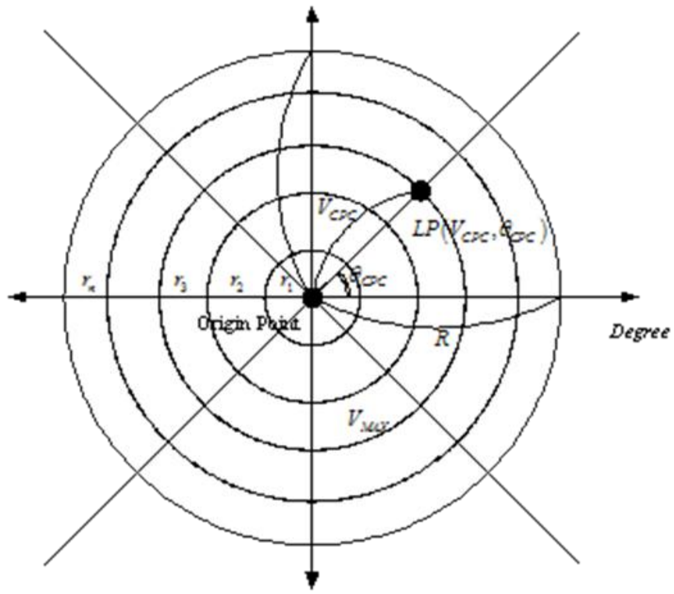



**Figure 4. Circular Coordinate System based Robot Locomotion Coordinate**

## 3. Circular Coordinate System based on Robot Locomotion Coordinates

A rectangular coordinate system is configured at the point of (0, 0) for the locomotion of a robot using GUI in a conventional remote control system which presents driving distance based on X, Y, and Z axes in a two-dimensional plane. The point at which these three axes cross represents the present location of the robot. Although the rectangular coordinate system can easily be implemented and generalized to arbitral dimensions, it has a disadvantage that a controller has to implicitly assume the velocity and the rotation angle of a robot. Thus, it is necessary to apply a robot locomotion coordinate system that can determine the velocity and the direction on a two-dimensional plane in order to control the robot. In this paper, a circular coordinate system based robot locomotion coordinate that can determine the velocity and the direction on a two-dimensional plane through GUI is proposed to control a mobile robot.

Figure 4 shows the circular coordinate system based robot locomotion coordinate which is applied to the GUI in a remote robot control system. On controlling a robot, the Locomotor Point(*LP*, ●) is to be moved to the desired driving direction. For instance, if the *LP* is moved to a point of $LP(V, \Theta)$, the driving will be carried out to the direction of $\theta$ with a velocity of $V$. Also, the radius $R$ of the circle shows the maximum velocity of the robot $V$, and $r_n$ represents the steps of the velocity.$r_1$, $r_2$, …, $r_n$ are determined by dividing the maximum velocity $V$ by $n$. The center point of the circle is defined as a central origin point and the distance between the central origin point and the *LP* represents the velocity of a center, $V$, of the robot. Also, the direction angle, $\theta$, shows a rotation angle. The present location of the robot is the central origin point. The velocity increases as far the robot is from this point and decreases as close the robot is to this point. As the direction angle is limited by $[0, 2\pi)$, the terms of $\pi/2$, $3\pi/2$, 0, and $\pi$ show forward driving, backward driving, right turn driving, and left turn driving, respectively. This method makes it possible to control the robot not only for forward driving but also for rotation driving while monitoring its turning angle and velocity. That is, the direction and the velocity of the robot can be controlled at the same time by moving the *LP* of the circular coordinate system. It represents following nine characteristics.

**Characteristic 1. A circular coordinate system represents a robot locomotion coordinate system.**

The distance between the central origin of a robot and the *LP* in the circular coordinate system shows the robot velocity, and the rotation angle shows its direction angle as a robot locomotion coordinate system.

**Characteristic 2. The driving direction and the velocity of a robot in a circular coordinate system can be determined by moving the *LP*, (●).**

As the *LP*, (●), is moved to the desired driving direction, the robot driving is carried out according to the determined velocity and rotation angle.

**Characteristic 3. The origin point, (0, 0), in a rectangular coordinate system shows the central origin point that indicates the present position of a robot.**

The origin point, (0, 0), in a rectangular coordinate system as presented in figure4 shows the central origin point that indicates the present position of a robot. If the *LP* stays at the central origin point, it will stop the robot. If a controller moves the *LP* to a specific position, the robot will be driven from the present position towards the determined direction with the determined speed.

**Characteristic 4. On the two points (origin and *LP*) of the GUI in a remote control system, the velocity is transformed with the Pythagorean theorem for the application to a circular coordinate system and the direction angle is transformed to trigonometric functions.**

Although the GUI in a remote control system uses a rectangular coordinate system to present a circular coordinate system, the locomotion of a robot is determined by its velocity and direction angle. Thus, the values in a rectangular coordinate system have to be transformed to the values in a circular coordinate system.

Equation (11) and (12) illustrate transformation methods for the velocity and direction angle of the GUI in a remote control system. The origin point of (0, 0) and (x, y) indicate the central origin point and *LP*, in a rectangular coordinate system. The velocity, *V*, can be obtained using the Pythagorean Theorem as noted in equation (11), and the direction angle, $\theta$, can be expressed as equation (12) according to the position of x and y. The coordinate of x and y is limited to [0, 2$\pi$).

$$v = \sqrt{x^2 + y^2} \tag{11}$$

$$\theta = \begin{cases} \cot\frac{y}{x} & \text{if } x > 0 \quad \text{and} \quad y \geq 0 \\ \cot\frac{y}{x} & \text{if } x > 0 \quad \text{and} \quad y < 0 \\ \cot\frac{y}{x} + 2\pi & \text{if } x < 0 \\ \frac{\pi}{2} & \text{if } x = 0 \quad \text{and} \quad y > 0 \\ \frac{3\pi}{2} & \text{if } x = 0 \quad \text{and} \quad y < 0 \end{cases} \tag{12}$$

**Table 1. Locomotion of a Mobile Robot According to the Directional Angle in the Circular Coordinate System**

| Configuration of the direction angle of the *LP* | Locomotion direction |
|---|---|
| $\theta = 0$ | Central origin point (stopping) |
| $3\pi / 2 < \theta < \pi / 2$ | Right turn driving |
| $\pi / 2 < \theta < 3\pi / 2$ | Left turn driving |
| $\theta = \pi / 2$ | Forward driving |
| $\theta = 3\pi / 2$ | Backward driving |
| $0 < \theta < \pi / 2, v > 0$ | Right turn forward driving |
| $0 < \theta < \pi / 2, v < 0$ | Right turn backward driving |
| $\pi / 2 < \theta < 3\pi / 2, v > 0$ | Left turn forward driving |
| $\pi / 2 < \theta < 3\pi / 2, v < 0$ | Left turn backward driving |

**Characteristic 5. Since the *LP* is determined by the central origin point, the vertical axis means the forward and the backward driving and the horizontal axis means the left and the right turn.**

As shown in Figure 4, if the *LP* is moved to $\pi/2$ or $3\pi/2$, the robot will be driven as forward or backward. In addition, if the *LP* is moved to $\pi/2 < \theta < \pi$ or $0 < \theta < \pi/2$, the robot will be driven as right or left turn. Table 1 shows the driving of the robot according to the direction angles of the *LP*.

**Characteristic 6. The radius of a circular coordinate system, *R*, is the maximum speed of a robot, *V*.**

The *LP* stands for the velocity and the direction of a robot and can be moved within the maximum radius in a circular coordinate system. That is, it cannot exceed the maximum radius, R. In addition, the maximum radius indicates the maximum velocity of a robot as the distance between the central origin point and the *LP* shows its velocity. The relationship between the velocity and the radius is as follow.

$$R=V \tag{13}$$

**Characteristic 7. The velocity of a robot is the same as the distance between the central origin point of the robot and the LP, *v*, *θ*.**

In the robot locomotion using a circular coordinate system, the new central coordinate (v, θ) moved from the initial central origin point within the circular coordinate system is the same as the velocity ratio between the maximum velocity and the new central origin point in the circular coordinate system. It can be noted as equation (14).

$$v = \frac{R \tan \theta}{V} \tag{14}$$

**Characteristic 8. As the number of concentric circles presented in a user interface is *n*, the velocity that a circle refers to can be obtained by dividing the maximum velocity of a robot by *n*.**

Although the concentric circles in a circular coordinate system can be expressed using a user definition, these are to be presented as a uniform ratio based on the maximum size circle. Regarding *n* small circles, which have a uniform ratio, the velocity between circles $v_n$ is presented as equation (15).

$$v_n = \frac{V}{n} \tag{15}$$

The maximum velocity of the *k*th circle presented within *n* circles can be determined as equation (16).

$$v_k = \frac{k \times V}{n} \tag{16}$$

**Characteristic 9. In the locomotion of a robot by moving the *LP*, the location of the robot after *Δt* can be presented using the direction and velocity in a circular coordinate system at the present position.**

A movement of the *LP* from the central origin point in a circular coordinate system means the movement of a robot. If the *LP* is not returned to the central origin point, the robot will follow the *LP* continuously. That is, as shown in Figure5, as the *LP* is moved to (*v1*, *θ1*) during$\Delta t_1$, the robot moves to the position of $t_1$. Also, as the *LP* is moved to (*v2*, *θ2*) during $\Delta t_2$, the robot moves to the point of $t_2$.

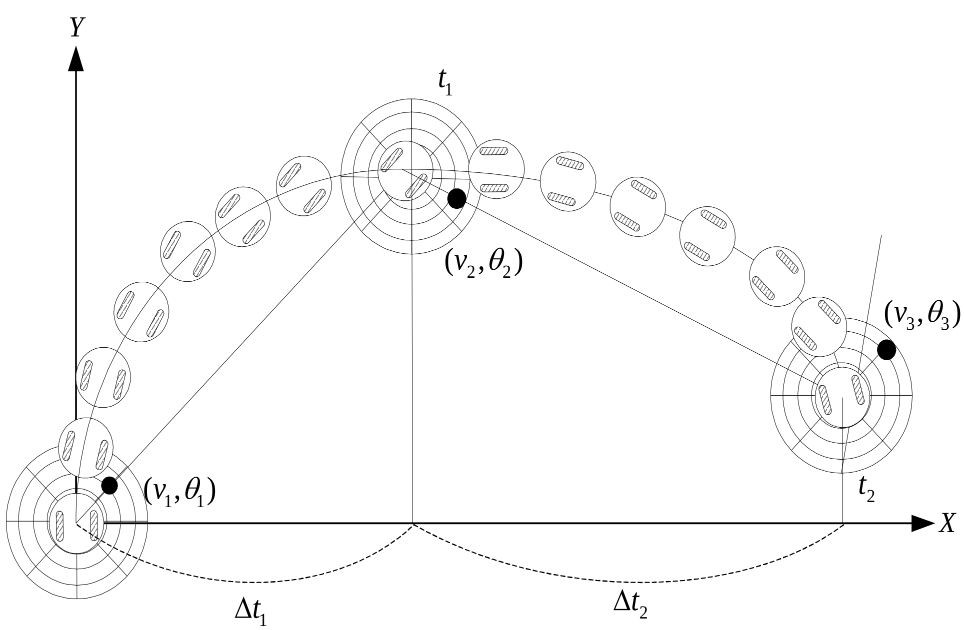



**Figure 5. Position of a Robot after Δt**

## 4. Establishment of a Test Environment

**Figure 6. Mobilerobot System Configuration**

Figure 6 shows the system for the test of mobile robot that is configured in this paper. The mobile robot consist of a motion controller for the control of differential operation, ultrasonic sensor array for measuring the distance between the robot and obstacles, USB camera for obtaining the images of surrounding environments, wireless LAN for transmitting data, and main controller for processing data. A controller can control the robot through the remote control system. The controller receives the video, distance to obstacles, and context information from the robot through a wireless LAN which are compressed with the H.263 and then transmits locomotion commands to the mobile robot which include the velocity and the direction information calculated using a circular coordinate system. Table 2 shows the specification of the mobile robot.

**Table 2. Specification of the Mobile Robot**

| | Specification |
|---|---|
| Mobile Robot | 1. Platform: Hanool Robotics Ltd., Hanuri-RD<br>2. Motion control board for three axes<br>3. Locomotion with two driving motors<br>4. Detecting obstacles using a sensor array with 12 ultrasonic sensors<br>5. USB camera with 1.30 million pixels<br>6. Pan/Tilt module (Left-Right: 330˚, Up-Down: 180˚) |
| Main Controller | Main controller: UMPC<br>1. CPU: Pentium IV (1GHz)<br>2. Wireless LAN available |
| Software | OS : Windows XP<br>Development tool: MFC |

The remote control application has been implemented on a smartphone, Samsung Galaxy A with Android 2.2. To build the remote control application, the Java SE Development Kit 6, the Android SDK, and the Eclipse Classic 3.6.1 with the ADT plugin are used. With this environment, the smartphone and the mobile robotcommunicates each other to communicatethe current status information and the locomotion commands.

## 5. Performance Evaluation

**Table 3. The Evaluation Method of the Mobile Robot**

| | |
|---|---|
| Evaluation method | Travelling twice along the square shaped track with 2m width and 2m height |
| Direction Control (DC) | Velocity: 100cm/sec<br>With forward, backward, turn left, turn right key.<br>Turn should be done after stopping. |
| Velocity-Direction Control (VDC) | Velocity: 60cm/sec ~ 360cm/sec<br>Angle: Changes according to the left or right key.<br>10cm/sec increases or decreases with a tap of forward or backward key |
| Circular Coordinate System Control (CCSC) | Velocity: 60cm/sec ~ 360cm/sec<br>Angle: Calculated from LP<br>Uses the Circular Coordinate System and changes velocity and angle according to the movement of the central origin point. |

Table 3 shows the evaluation method to evaluate the performance of the mobile robot. The evaluation is carried out while travelling twice along the square shaped track with 2m width and 2m height. A controller controls the mobile robot with the GUI based remote control system and arrow keys on the keyboard, referring real-time video and information about obstacles. On evaluation, the performance of the DC method, VDC method, and the CCSC method are compared.

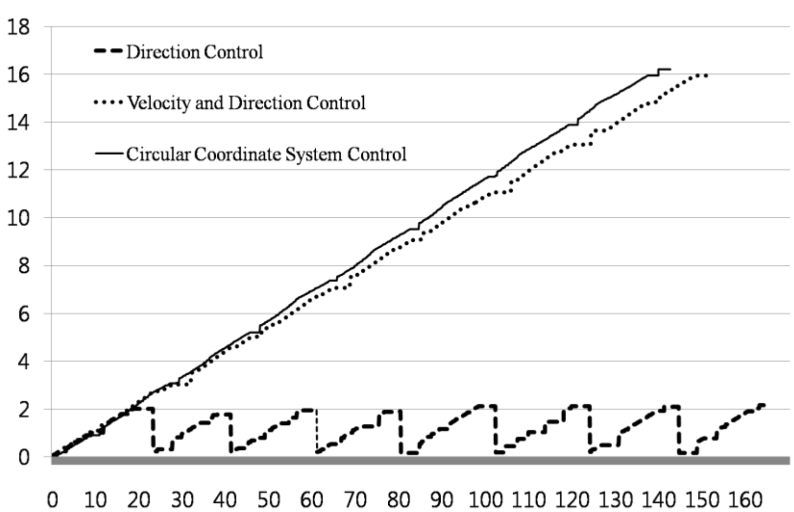



**Figure 7. The Result of Experiments on Square Track with 2m Width and 2m Height**

Figure 7 shows the locomotion distance of the mobile robot with respect to the time when the experiments have been carried out according to the table 3. The horizontal axis represents the time in seconds and the vertical axis represents the distance in meters. The result of the DC method which is based on a rectangular coordinate system has saw-toothed graph as the mobile robot has to change its direction after stopping and the encoder is reset when the mobile robot stops. The result of the VDC method which is based on a rectangular coordinate system and the CCSC method which is based on a circular coordinate system have linear graphs as the mobile robot does not stop while moving, though they are uneven. Table 4 shows the average locomotion time and the distance which are the results of the experiments on table 3. In terms of the locomotion time, the CCSC method achieved 14.43% and 6.44% better performance than the DC and the VDC methods. And in terms of the locomotion distance, there are no meaningful errors between the methods; the CCSC method achieved 1.19% and -0.42% better performance than the DC and the VDC method while travelling 16m.

**Table 4. The Results of Experiments on Square Track with 2m Width and 2m Height**

| Locomotion method | Direction Control | Velocity-Direction Control | Circular Coordinate System Control |
|---|---|---|---|
| Time (sec) | 163.4 | 152 | 142.8 |
| Distance (m) | 16.19016 | 15.93295 | 16.20172 |

The evaluation results show that controlling the mobile robot with the CCSC method reduces the locomotion time while providing accuracy of locomotion.

## 6. Conclusion

To improve the locomotion performance of a mobile robot, an HRI technology is essentially required. In this paper, a robot locomotion coordinate based circular coordinate system for the GUI of a remote control system was proposed. In the results of the tests performed by implementing a mobile robot system, it represented decreases in locomotion times by about 33.10% and 32.32%, compared with other regular locomotion methods. The proposed method can be applied not only to a mobile robot but also to intelligent robots for the locomotion of home cleaning robots, guarding robots, and so on. In addition, the remote control method using a circular coordinate system can be used as a useful technology in controlling the robots.

## Acknowledgements

This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the CITRC(Convergence Information Technology Research Center) support program (NIPA-2012-H0401-12-1007) supervised by the NIPA (National IT Industry Promotion Agency).

## References


[1] J. H. Kim, D. W. Kim, B. J. You and G. T. Park, "A Design of Framework for Smart Service of robots in Intelligent Environment", International Journal of Control and Automation, vol. 2, no. 4, **(2009)**, pp. 1-12.

[2] R. C. Luo, T. M. Chen and C. C. Yih, "Intelligent autonomous mobile robot control through the Internet", Proceedings of The 2000 IEEE International Symposium on Industrial Electronics, **(2000)** December 4-8, Peubla, Mexico.

[3] P. T. M. Saito, R. J. Sabatine, D. F. Wolf and K. R. L. J. C. Branco, "An Analysis of Parallel Approaches for a Mobile Robotic Self-Localization Algorithm", International Journal of Feature Generation Communication and Networking, vol. 2, no. 4, **(2009)**, pp. 49-64.

[4] T. Takiguchi, T. Yamagata, A. Sako, N. Miyake, J. Revaud and Y. Ariki, "Human-Robot Interface Using System Request Utterance Detection Based on Acoustic Features", International Journal of Multimedia and Ubiquitous Engineering, vol. 1, no. 2, **(2006)**, pp. 25-29.

[5] A. Dix, J. E. Finlay, G. D. Abowd and R. Beale, Editor, "Human-Computer Interaction", Pearson press, **(2003)**.


## Authors

**MunSuck Jang** received the B.S. degree in Computer Engineering from KonYang Univ., NonSan, Korea, in 1997, and the M.S. and Ph.D degrees in Electronic Engineering from Inha Univ., Incheon, Korea, in 2000 and 2010, respectively. He is currently a Industrial Collaboration professor at Korea Polytechnic University. His main research interests are in the areas of service robot control, mobile healthcare system, computer architecture & network, embedded system, and various industrial applications.

**EungHyuk Lee** received the B.S. degree in Electronics Engineering from Inha University, Incheon, Korea, in 1985, and the M.S. degree and the Ph.D. degree in Electronic Engineering from Inha University, Incheon, Korea, in 1985 and 1987, respectively. From 1987 to 1992, he was a researcher at Industrial Robot Lab. of Daewoo Heavy Industry Co. Ltd. From 1995 to 2000, he was a assistive professor at Dept. of Computer Engineering in KonYang University. Since 2000, he has been with the Department of Electronics Engineering at Korea Polytechnic University. His main research interests are in the areas of service robot control, mobile healthcare system, image processing and various industrial applications.

**SangBang Choi** earned the M.S and Ph.D. degrees in Electrical Engineering from the University of Washington, Seattle, in 1988 and 1990, respectively. He is currently a professor of Electronic Engineering at Inha University, Incheon, Korea. His research interests include computer architecture, computer networks, wireless communication, and parallel and distributed systems. He is a member of the IEEE and IEEE Computer Society.