A Study on HRI using a Circular Coordinate System in Robot Locomotion Coordinates for Remotely Controlled Robots

***Mun-Seuk Jang*†*, Eung-Hyuk Lee*††**

†Research and Business Foundation, Korea Polytechnic University, Shiheung-City,Kyonggi-Do, Korea
††Dept. of Electronic Engineering, Korea Polytechnic University, Shiheung-City, Kyonggi-Do, Korea

Abstract
In this study, an HRI method using a circular coordinate system, which can implement robot locomotion with variable velocity instead of controlling velocity and direction using buttons in a user program for remotely controlled robots and easily determine direction angle is proposed. Also, the nine characteristics of this proposed circular coordinate system are to be described and the principles of user interfaces using the circular coordinate system are to be analyzed. Thus, a remotely controlled robot system that can be controlled using wireless LAN in a mobile robot, which can perform differential operation, and a PC is configured and tested. In the results of a corridor driving test using the proposed method, driving distance and time decrease about 17% and 33%, respectively, compared with that of other regular methods.

Key words:
Remotely Controlled Robot, HRI, Circular coordinate system, robot locomotion coordinate, Computer Interface Device

1. Introduction

Remotely controlled robots are usually used in an environment that requires remotely controlled environment monitoring and works through wire or wireless networks from a remote location. Technologies for configuring a remotely controlled robot system are a robot fabrication technology for building a robot platform, a network infrastructure technology for implementing remote control, and an internal element technology for providing services to users using a robot [1,2]. In particular, an HRI (Human-Robot Interface) technology should be required to perform interaction between a robot and a user for controlling a robot efficiently from a remote location and guaranteeing its locomotion performance.

Most studies on the HRI in remotely controlled robots are usually focused on five sense information and robot control methods using new devices. In the field of E-Learning robots, some studies on new devices for managing education contents have been conducted [4]. Also, in the robot using five sense information, various studies on face recognition and verification [5] and biometric information for recognizing users [6,7] have been conducted. In addition, studies on user motion recognition [8,9] and robot control using a voice recognition method [10] have been conducted. However, these studies are focused on certain specific functions in robots and difficult to practically implement. Also, as they represent low recognition rates and show difficulties in real-time implementation, studies on the practical applications of these technologies to robots are required.

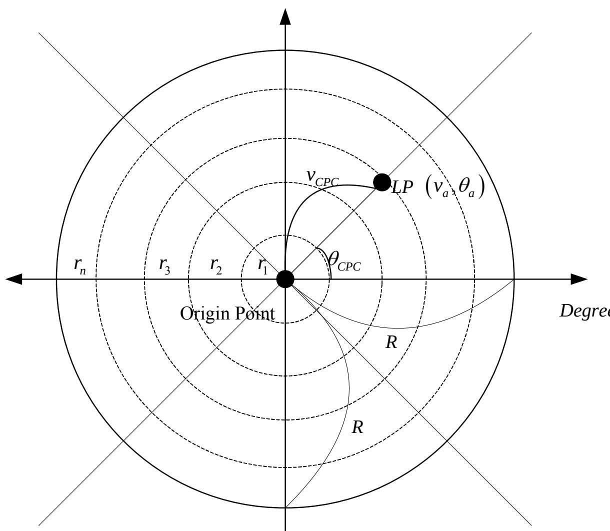
The most largely used devices for the HRI in the conventional remotely controlled robots are user computer interfacing devices such as keyboard, mouse, joystick, and so on. The device using a keyboard controls the direction of a robot as straight or reverse driving and left or right turn using four direction keys. Also, the device using a mouse shows a simple way that controls the direction of a robot using a simple interface after configuring direction buttons in a user program. These devices represent some disadvantages in controlling a robot because the robot should be stopped to turn or change its direction and that can allow uniform velocity driving only and cannot support rotational driving. Although the device using a joystick shows easy velocity control and can support rotational driving compared with that of using a keyboard or mouse, it represents a disadvantage in precision control because it is difficult to exactly control the velocity and direction angle of a robot.

For solving these problems, in this study, an HRI method using a circular coordinate system, which can implement robot locomotion with variable velocity instead of controlling velocity and direction using buttons in a user program for remotely controlled robots and easily determine direction angle is proposed. Experiments for this proposed method include various driving tests, such as direction control with uniform velocity, velocity and direction control with variable direction and velocity, and driving in an 'L' shape corridor based on a circular coordinate system using a keyboard and mouse. In the results of the analysis of these tests, the method using a circular coordinate system represents decreases in driving times by about 33.10% compared with that of applying direction control tests and 32.32% compared with that of applying velocity and direction control tests.

Manuscript received April 5, 2011
Manuscript revised April 20, 2011

2. Circular coordinate system based on robot locomotion coordinates

For the locomotion of a robot using GUI (Graphic User Interface) in a remote control program, a rectangular coordinate system, which presents driving distance based on X, Y, and Z axes in a two-dimensional plan, is to be configured in which the point of (0, 0) that is 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 users determine the velocity and rotation angle of a robot suggestively. Thus, it is necessary to apply a robot locomotion coordinate system that can determines the velocity and rotation angle in a two-dimensional plan in order to control the robot from a remote location. In this study, a circular coordinate system based on robot locomotion coordinates that can determine the robot velocity and direction by using the circular coordinate system in a two-dimensional plan in GUI is proposed to control a remotely controlled robot.



R	: Maximum velocity of a robot
r_n	: Divided velocity
$LP\ (v_a, \theta_a)$	: Locomotor Point
(0, 0)	: Origin Point
v_{CPC}	: Central Velocity
θ_{CPC}	: Direction Angle

Fig. 3 Circular coordinate system based on robot locomotion coordinates

Fig. 3 shows a circular coordinate system based on robot locomotion coordinates applied to the GUI in a remote robot control program. For controlling a robot, the locomotor point, (LP, ●), is to be moved to the desired driving direction. For instance, if the locomotor point is moved to a direction of $LP(V_{CPC}, \theta_{CPC})$, the driving will be implemented to the direction of θ_{CPC} with a velocity of V_{CPC}. Also, the radius (R) of the circle shows the maximum velocity of a robot (V_{MAX}) and r_n represents the divided velocity ($r_1, r_2, \ldots, r_n$) that is determined by dividing the division of the maximum velocity (V_{MAX}) by n. The center point of the circle is determined as a central origin point and the distance between the central origin point and the locomotor point represents the center velocity, V_{CPC}, of the robot. Also, the direction angle, θ_{CPC}, 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 far the robot is close to this point. As the direction angle is limited by $[0, 2\pi)$, the terms of $\pi/2$, $3\pi/2$, 0, and π show forward driving, backward driving, right turn driving, and left turn driving, respectively. This method makes possible to control the robot not only for forward driving but also for rotation driving by checking its turning angle and velocity. That is, the direction and velocity of the robot can be controlled at the same time by moving the locomotor point 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 locomotor point in a 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 velocity of a robot in a circular coordinate system can be determined by moving the locomotor point, (●).
As the locomotor point, (●), is moved to the desired driving direction, the robot driving is implemented by 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 Fig. 3 shows the central origin point that indicates the present position of a robot. If the locomotor point shows the same location as the central origin point, it will show stopping the robot. In this study, it is defined as a CPC (Center of Polar Coordinate) point. If a user moves the locomotor point with an arbitral speed and angle, the robot will be driven in which the present position of the robot during the driving is determined as the central origin point.

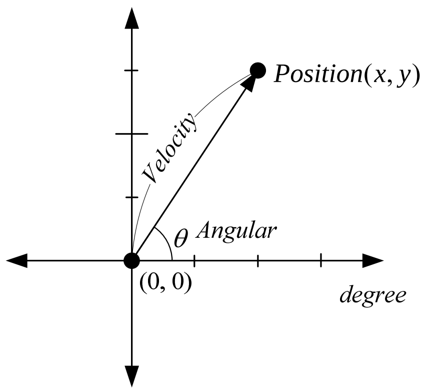


Fig. 4 Transformation of the velocity and directional angle of the GUI in a remote control program

Characteristic 4. In the two points (origin and locomotor points) of the GUI in a remote control program, the velocity is transformed by the Pythagorean theorem for applying it to a circular coordinate system and the direction angle is transformed as trigonometric functions.

Although the GUI in a remote control program uses a rectangular coordinate system for presenting 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 are to be transformed as the values in a circular coordinate system.

Fig. 4 illustrates a transformation method for the velocity and direction angle of the GUI in a remote control program. The origin point of (0, 0) and (*x, y*) indicate the central origin point and locomotor point, respectively, in a rectangular coordinate system. Also, the velocity, V, can be obtained using the Pythagorean theorem as noted in Eq. (1), and the direction angle, θ, can be expressed as Eq. (2) according to the position of x and y as the coordinate of x and y is limited as $[0, 2\pi)$.

$$V = \sqrt{x^2 + y^2} \quad (1)$$

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

Characteristic 5. As the locomotor point is determined based on the central origin point, the longitudinal axis means the forward and backward driving in a robot and the latitudinal axis shows the left and right turn driving.

As shown in Fig. 2.9, if the locomotor point is moved to $\pi/2$ or $3\pi/2$, the robot will be driven as forward or backward driving respectively. In addition, if the locomotor point is moved to $\pi/2 < \theta_{CPC} < \pi$ or $0 < \theta_{CPC} < \pi/2$, the robot will be driven as right or left turn driving respectively. Table 1 shows the driving of the robot according to the direction angles of the locomotor point.

Table 2.1 Locomotion of a remotely controlled robot according to the directional angle in a circular coordinate system

Configuration of the direction angle of the locomotor point ($\theta_{CPC} = radian, V_{CPC} = distance$)	Locomotion direction
$\theta_{CPC} = 0$	Central origin point (stopping)
$3\pi/2 < \theta_{CPC} < \pi/2$	Right turn driving
$\pi/2 < \theta_{CPC} < 3\pi/2$	Left turn driving
$\theta_{CPC} = \pi/2$	Forward driving
$\theta_{CPC} = 3\pi/2$	Backward driving
$0 < \theta_{CPC} < \pi/2, V_{CPC} > 0$	Right turn forward driving
$0 < \theta_{CPC} < \pi/2, V_{CPC} < 0$	Right turn backward driving
$\pi/2 < \theta_{CPC} < 3\pi/2, V_{CPC} > 0$	Left turn forward driving
$\pi/2 < \theta_{CPC} < 3\pi/2, V_{CPC} < 0$	Left turn backward driving

Characteristic 6. The radius of a circular coordinate system, R, is the maximum speed of a robot (V_{MAX}).

The locomotor point shows the velocity and direction of a robot and can be moved within the maximum radius in a circular coordinate system. That is, it can exceed the maximum radius, R, as illustrated in Fig. 3. In addition, as the distance between the central origin point and the locomotor point shows its velocity, the maximum radius indicates the maximum velocity of a robot. The relationship between the locomotion velocity and the radius is presented as Eq. (3).

$$R = V_{MAX} \quad (3)$$

Characteristic 7. The locomotion velocity of a robot is the same as the distance between the central origin point of the robot and the locomotor point, V_{CPC}, θ_{CPC}, in Fig. 3.

In the robot locomotion using a circular coordinate system, the new central coordinate (V_{CPC}, θ_{CPC}) 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 Eq. (4).

$$R : V_{MAX} = V : \tan\theta$$
$$V = \frac{R\tan\theta}{V_{MAX}} \tag{4}$$

Characteristic 8. As the maximum number of small circles presented in a user interface is determined as n, the velocity of a circle can be obtained by dividing the maximum velocity of a robot by n.

Although the small 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 Eq. (5).

$$V_n = \frac{V_{MAX}}{n} \tag{5}$$

The maximum velocity of the *k*th circle presented within n circles can be determined as Eq. (6).

$$V_k = \frac{k \times V_{MAX}}{n} \tag{6}$$

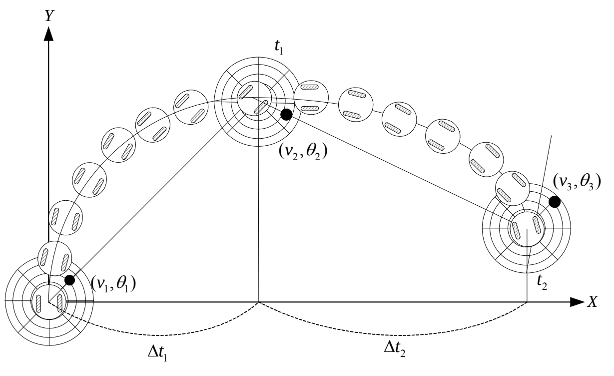


Fig. 5 Location of a robot after Δt

Characteristic 9. In the locomotion of a robot by moving a locomotor point, 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 locomotor point from the central origin point in a circular coordinate system means the movement of a robot. If the locomotor point is not returned to the central origin point, the robot will be moved to the position of the locomotor point continuously. That is, as shown in Fig. 5, as the locomotor point is moved to (V_1, θ_1) for Δt, the robot is moved to the position of t_1. Also, as it is moved to (V_2, θ_2) for Δt, the robot is moved to the point of t_2.

3. Principles of the user interface using a circular coordinate system

It is necessary to increase the usability of users for the interaction between a user and a robot. For achieving it, the principles of the user interface can be largely classified as learnability, flexibility, and robustness [3]. The learnability is related to the effective access to a robot system by a new user and the obtaining of the maximum performance from the implemented work. The flexibility is related to the exchange of information between a user and a robot system through various ways. The robustness is related to the increase in the achievement of the work performed by a user [3]. The detailed principles for these three major user interfaces are as follows:

The learnability is divided into predictability, synthesizability, familiarity, generalizability, and consistency. The predictability is to estimate what will happen in a system based on the mutual response of the past. The synthesizability represents user's activities on the present situation and what will happen in the future. The familiarity shows an increase in the probability of the successive performance in using a new product due to the knowledge and experience of users in such mutual response with the new product. The generalizability shows the extensibility of knowledge for certain mutual response in a different situation. That is, as processing a work of A, it is possible to naturally recognize a work of B, which is similar to that of A. Finally, the consistency represents the maintaining of the similarity in input and output ways under similar situation and purpose.

The flexibility is divided into dialog initiative, multi-threading, task migratability, substitutivity, and customizability. The dialog initiative is to allow the freedom from the artificial limitation in input sentences applied to a system and represents a priority of starting the communication between a user and the system for considering their mutual response. The multi-threading is to support the mutual response of a user that belongs to more than one work at the same time. The task migratability considers the obtaining of the controllability for implementing a work between a system and a user. The substitutivity enables a certain function as various functions having same values. Finally, the customizability is a change in user interface by users and a system. The change represents modifications caused by changing the system by a programmer and automatic modifications by the system.

The robustness is divided into observability, recoverability, responsiveness, and task conformance. The observability is to allow the recognition of the internal situation in a system through some expressions, which can be sensed in interfaces by users. The recoverability makes an attempt of right actions by users immediately as errors are detected. The responsiveness measures the communication speed between a system and a user. Finally, the task conformance shows how much of services are provided to the works, which are to be performed by users, and the degree of understanding the services by users.

In this study, the user interface principles, such as learnability, flexibility, and robustness, with respect to the proposed circular coordinate system can be analyzed as follows:

In the learnability, robots are used in some variously changed environments and not easy subjects in uses because they represent various functions and difficulties in control. Thus, the learnability that enables to easily access robots and to be well acquainted with robots is the most important factor for using robots rather than other elements. The learnability in robots on the basis of using the proposed circular coordinate system provides an easy access to a robot using a mouse, keyboard, and joystick only for the users who are accustomed to PCs. The learnability consists of predictability, synthesizability, familiarity, generalizability, and consistency. The predictability shows that a user estimates the direction and velocity in the locomotion of a robot by configuring the origin, (0, 0), of a circular coordinate system as its present position. The synthesizability means that a user makes possible to move a robot to the central origin point of a circular coordinate system and can estimate the next position. The familiarity shows that a user increases not only the familiarity in controlling a robot using some accustomed devices, such as keyboard, mouse, and joystick, based on their knowledge and experience who are accustomed to PCs, but also the probability of operating the robot successfully even thought the user operates a robot in a circular coordinate system for the first time. The generalizability represents that a user can understand the operation of a robot by verifying the movement of the central origin point using other tools instead of using such keyboard, mouse and joystick. Finally, the consistency shows that a user performs the locomotion of a robot by moving the central origin point and maintains the way of moving such central origin point for the locomotion of the robot.

In the flexibility, as robots are to be used for a long time with its various functions, it is necessary to provide proper functions according to the condition and environment in its applications. Also, it is essential to consider various situations because each user has different experience and personal intellectual capability. The flexibility consists of dialog initiative, multi-threading, task migratability, substitutivity, and customizability. The dialog initiative represents that a user applies a command for the locomotion of a robot by moving the central origin point in a circular coordinate system and that leads to start the communicate by a user first based on the mutual response between a robot and a user. The multi-threading uses a circular coordinate system and supports a user interface that verifies the context information (recognizing images or obstacles) around a robot. The task migratability performs a communication for the locomotion data of a robot through a specific communication protocol between a system and a user. The substitutivity shows a replacement in devices for applying a circular coordinate system in which a keyboard or joystick can be used instead of using a mouse, which is the most commonly used to control a robot. Finally, the customizability is to modify the user interface by a user and a robot. The change can be automatically performed by the modification of a system through modifying the maximum/minimum velocity of a robot by a programmer.

In the robustness, it is necessary to successfully and effectively move a robot based on the mutual response between a robot and users. The robustness consists of observability, recoverability, responsiveness, and task conformance. The observability represents that a user can recognize a circular coordinate system in user interfaces and transmits locomotion commands based on this recognition and that leads to understand environment situations through the information obtained from images and ultrasonic sensors. The recoverability allows the operation of a locomotor point within a circular coordinate system only in order to limit the velocity and direction angle of a robot for achieving user's correct operation. The responsiveness can implement robot operation as a real-time manner by transmitting commands as a user moves a locomotor point in a circular coordinate system. Finally, the task conformance represents that a user can provide robot locomotion services by moving a locomotor point in a circular coordinate system.

4. Establishment of a test environment for remotely controlled robots

Fig. 6 shows the system configured in this study for the test of a remotely controlled robot. The remotely controlled robot consists 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. Also, a user can control a robot using a remote control program. The user receives the image information, distance data to obstacles, and context information compressed by using H.263 from the robot through a wireless LAN and then transmits locomotion commands including the velocity and direction information calculated using a circular coordinate system to the remotely controlled robot.

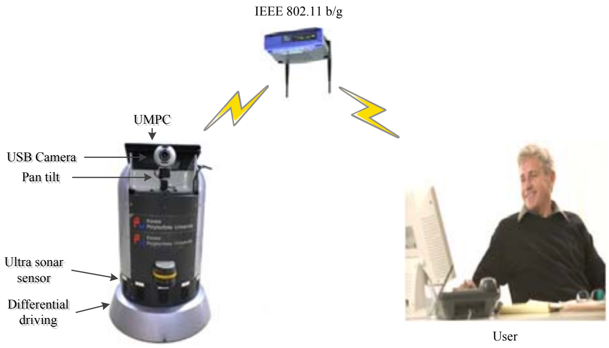


Fig. 6 System configuration for a remotely controlled robot

The robot control process using a remote control program by a user can be summarized as follows:

1. The right to control a remotely controlled robot can be obtained by inputting the IP of the robot. If the right to control the robot is being used by other users, a user who wants to use it waits the next chance.

2. If the right to control the robot is obtained, a user transmits a motion initialization command to the robot. Then, the robot transmits the result to the user.

3. An image information requirement command is to be transmitted to the robot in order to obtain images from the USB camera installed at the robot. Then, the robot transmits the image information to the user.

4. A control command is to be transmitted to the robot for implementing the locomotion of the robot. Then, the robot will move according to the control command. Also, the robot transmits the distance from obstacles obtained from a sensor array during its locomotion.

Table 2 shows the specification of the remotely controlled robot used in this test. Fig. 7 represents the program used to operate the robot. A motion board for three axes was used to control robot motion and designed for operating two motors. Also, a sensor array with 12 ultrasonic sensors was installed at the robot for observing distance and surrounding obstacles. The operating system and development tool used in this system were Windows XP Microsoft Visual Studio 8.0 respectively.

Table 2 Specification of the mobile robot

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

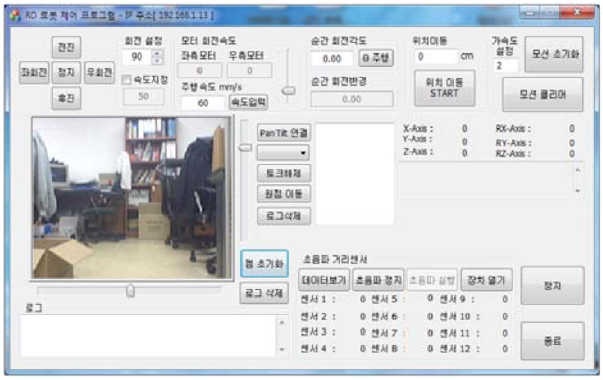


Fig. 7 GUI for the robot control program

Table 3 Specification of the remote control program

	Specification
Device	Keyboard, Mouse, Joystick (5 axes, 12 buttons)
Software	OS: Windows XP Tools: MFC SDK: Windows SDK for Windows Server 2008 .NET Framework 3.5 (Version 6.1) DirectX SDK 9.0 (Version 9.26.1590)

Table 3 shows the PC used to the remote control program. Fig. 8 represents the remote control program implemented in this system. Also, the interface devices for this remote control are a keyboard, mouse, and joystick. The remote control program was configured to control the robot using a circular coordinate system. The program controls the robot by recognizing surrounding environments and obstacles based on the image displayed at the upper left section and the information obtained from the sensor array presented at the lower center section.

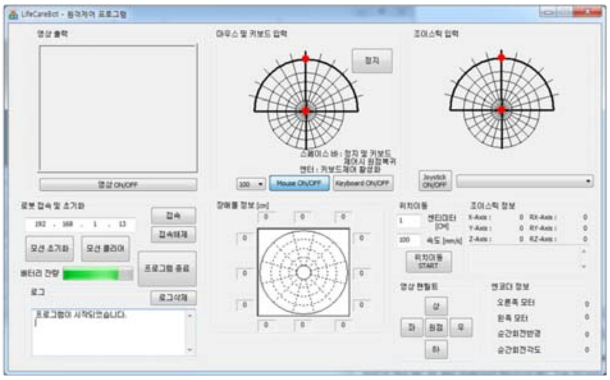


Fig.8 GUI for the remote control program

5. Test and analysis of the results

Table 4 Locomotion method for the performance evaluation of the remotely controlled robot

Reference Coordinate	Driving Method	Description
Rectangular Coordinate System	Direction Control	The robot velocity is fixed by 100cm/sec, and the 90° turn driving is applied using left, right, up, and down keys. The rotation driving is processed as a sequence of driving → stop → turn → driving.
	Velocity and Direction Control	The robot velocity can be controlled as 60cm/sec~360cm/sec. The driving is applied using keys in a keypad in which ‘2’, ‘4’, ‘6’, and ‘8’ keys represent the same direction as the key direction in the keyboard. However, if the same key is repeatedly pushed, the velocity increases an interval unit of 10cm/sec or decreases in the velocity with an interval unit of 10cm/sec. The keys, ‘1’, ‘3’, ‘7’, and ‘9’, are used to 45° angle turn driving.
Robot Locomotion Coordinate System	Circular Coordinate System Control	The velocity and direction can be controlled by moving the central origin point in a circular coordinate system. The velocity of the remotely controlled robot can be controlled as 60cm/sec~360cm/sec. Also, the change in angles can be configured by moving a locomotor point.

The test for evaluating the HRI performance of the robot using a circular coordinate system was performed based on the differences in driving time and distance measured by applying going and return driving through the determined corridor. In the test, the driving considers the obstacle and image information only in the GUI of the remote control program. Also, the direction of the robot can be changed using direction keys in a keyboard based on a rectangular coordinate system. The driving performances of some driving methods that include a direction control method with constant velocity, a velocity variable driving method using a numeric keypad, a velocity and direction control method with 45° turn driving, and a velocity and direction control method based on a circular coordinate system, which enables variable velocity and rotation driving using a robot locomotion coordinate system, were compared. The driving methods for each driving test are presented in Table 4.

Fig. 9 shows the driving distance according to time by extracting the encoder data in the robot with an interval of 100ms as the going and return robot locomotion is applied through a corridor, which shows an 'L' shape with a dimension of length × width, 3.5m × 10.5m. The latitudinal axis shows the passage of time as a unit of 1 second, and the longitudinal axis represents the driving distance as a unit of m. The direction control test for the driving time showed a delay of 418 seconds due to the stop for waiting its constant and turn driving. In the velocity and direction control tests, a user could not recognize the change in the robot velocity where the time showed a delay of about 428 seconds due to the difficulty of controlling its direction angle. However, in the circular coordinate system control test, it was possible to perform the driving with the maximum velocity in a linear section but a curved section represented a delay of about 138 seconds due to the decrease in its velocity for the rotation driving of the robot. The direction control test represented the driving about 42m for the entire driving way of 48m, and the velocity and direction control tests indicated the driving about 43m. Whereas, the circular coordinate system control test showed the driving about 40m and that was the largest reduction in driving distance.

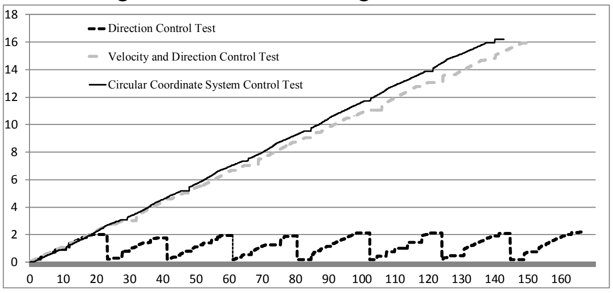


Fig. 9 Results of the locomotion through a corridor

Table 5 shows the average driving time and distance obtained from five different corridor driving tests for the rectangular coordinate system based direction control test, velocity and direction control test, and robot locomotion coordinate based circular coordinate system. In the results of these tests, the circular coordinate system based control tests showed decreases in driving times by about 33.10% and 32.32% compared with that the direction control tests and the velocity and direction control tests, respectively. In addition, in these corridor driving tests, because the robot locomotion in a linear way was performed by the maximum velocity and the rotation driving was performed by maintaining a proper level of velocity, the driving times in this circular coordinate system decreased. Therefore, the driving distances in the rectangular coordinate system based direction control test, velocity and direction test, and circular coordinate system based driving test decreased by about 8%, 12%, and 17%, respectively. In the performance based on driving methods, the circular coordinate system based control method decreased the driving time and distance by about 67% and 9%, respectively, compared with that of the direction control method. Also, the method decreased the driving time and distance by about 68% and 5%, respectively, compared with that of the velocity and direction control method.

Based on the results of these tests, it was verified that the circular coordinate system based driving can reduce the driving time and distance compared with that of the rectangular coordinate system based driving in the remotely controlled robot.

Table 5 Results of the locomotion through a corridor

Reference Coordinate	Rectangular Coordinate System		Robot Locomotion Coordinate System
Driving Method	Direction Control	Velocity and Direction Control	Circular Coordinate System Control
Time (sec)	418.8	428.8	138.6
Distance (m)	43.876131	42.11673	39.83141

6. Conclusion

For improving the locomotion performance of remotely controlled robots, which implement environment monitoring and works from a remote location through wire and wireless networks, an HRI technology is essentially required to present the relationship between a robot and users. In this study, a robot locomotion coordinate based circular coordinate system in the GUI of a remote control program was proposed. In the results of the tests performed by implementing a remotely controlled robot system, it represented decreases in driving distances and times by about 17% and 33%, respectively, compared with other regular locomotion test methods. The method proposed in this study can be applied not only to remotely controlled robots 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 will be used as a useful technology for controlling robots. In future studies, this method will be applied to the remote control system using cell phones and recently interested smart phones through some additional studies.

References

[1]. H.Kim, J.H.Lee, J.Soh, "Intelligent Service Robot with Netwoking-URC," ETRI CEO Information, vol. 15, pp.1-19, 2004, 10.

[2]. K.W.Kang, J.W.Park, H.H.Song, S.R.Oh, "Network Robot of the Industry and Prospects", Journal of electronic engineering of korea, vol. 34, No. 1, pp.55-70. 2007.1

[3] Alan Dix et al., *Human-Computer Interaction*, pp. 260-273, Pearson press, 2004.

[4]. D.S.Kim, M.S.Jang, O.S.Kwon, S.B.Choi, H.K.Min, E.H.Lee, "A Study of Ubiquitous Device for English learning of young children with Ubiquitous Learning Robot", TheInstitute of Electronics Engineers of Korea Summer conference 2009, vol. 32. No. 1, pp.623-624, 2009.7

[6]J.Y.Lee, Y.J.Cho, "User Recognition Technology of Robot Enviroment", The Journal of the Korean Institute of communication sciences, vol. 21, No. 10, pp. 1226-1240, 2004. 10.

[7] D. H. Kim, J. Y. Lee, H. S. Yoon, H. J. Kim, Y. J. Cho, and E. Y. Cha,"A vision based user authentication system in robot environments by using semibiometrics and tracking,"*IEEE/RSJ International Conference of Intelligent Robots and Systems (IROS 2005)*, pp. 246-251. Aug. 2005.

[8] K.K.Kim, H.J.Kim, S.H.Cho, J.Y.Lee "Gesture Recognition for Natural Human-Robot Interaction,"Journal ETRI Brochure & Report,vol. 20, No. 2, pp. 14-20, 2005. 4.

[9] M.S.Ko, K.H.Lee, Ch.W.Kim, J.H.Ahn, J.J.Kim "An Implementation of User Interface using Vision-based Gesture Recognition,"KCC2009, vol. 35, No. 1, pp. 507-511, 2008. 6.

[10] Y.G.Lee, J.Park, S.H.Kim, "The Speech Recogition Technology",Journal ETRIBrochure & Report, vol. 20. No. 5, pp. 1-15, 2005. 10.

Mun-Suck 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, repectively. His main research interests are in the areas of service robot control, mobile healthcare system, Computer Architecture & Network, embedded system, and various industrial applications.

Eung-Hyuk 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.