

Article

Intelligent Lead: A Novel HRI Sensor for Guide Robots

Keum-Bae Cho * and Beom-Hee Lee

Department of Electrical Engineering and Computer Science, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 151-742, Korea; E-Mail: bhlee@snu.ac.kr

* Author to whom correspondence should be addressed; E-Mail: kbcho98@snu.ac.kr;
Tel: +82-2-880-6487; Fax: +82-2-888-4182.

Received: 23 May 2012; in revised form: 6 June 2012 / Accepted: 6 June 2012 /

Published: 14 June 2012

Abstract: This paper addresses the introduction of a new Human Robot Interaction (HRI) sensor for guide robots. Guide robots for geriatric patients or the visually impaired should follow user's control command, keeping a certain desired distance allowing the user to work freely. Therefore, it is necessary to acquire control commands and a user's position on a real-time basis. We suggest a new sensor fusion system to achieve this objective and we will call this sensor the "intelligent lead". The objective of the intelligent lead is to acquire a stable distance from the user to the robot, speed-control volume and turn-control volume, even when the robot platform with the intelligent lead is shaken on uneven ground. In this paper we explain a precise Extended Kalman Filter (EKF) procedure for this. The intelligent lead physically consists of a Kinect sensor, the serial linkage attached with eight rotary encoders, and an IMU (Inertial Measurement Unit) and their measurements are fused by the EKF. A mobile robot was designed to test the performance of the proposed sensor system. After installing the intelligent lead in the mobile robot, several tests are conducted to verify that the mobile robot with the intelligent lead is capable of achieving its goal points while maintaining the appropriate distance between the robot and the user. The results show that we can use the intelligent lead proposed in this paper as a new HRI sensor joined a joystick and a distance measure in the mobile environments such as the robot and the user are moving at the same time.

Keywords: guide robot; visually impaired; serial linkage; IMU; Kinect sensor; EKF

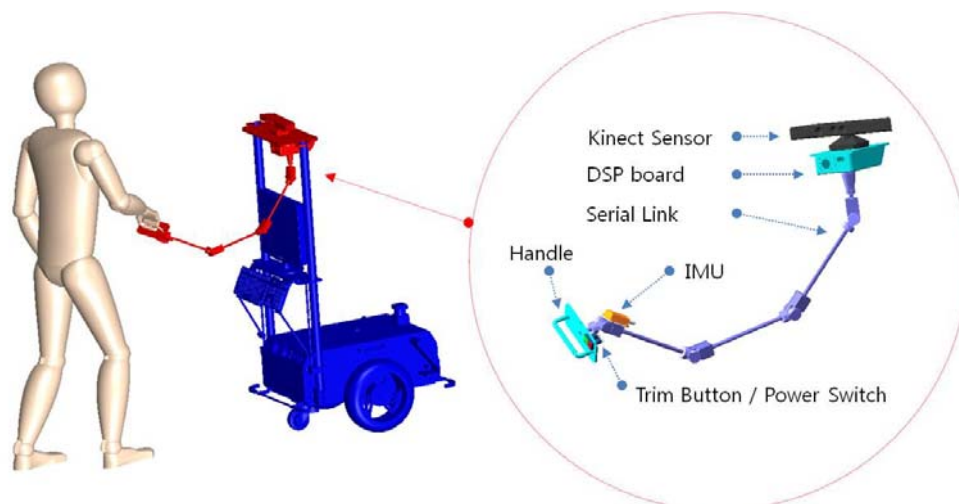
1. Introduction

There are approximately 285 million visually impaired people around the globe. Among them, 39 million people are totally blind. In addition, 90% of the visually handicapped live in developing countries [1]. In Korea, there are approximately 45,000 visually impaired people. However, they are provided with only 60~70 guide dogs. Due to the aging of the population, the numbers of geriatric patients who suffer from dementia or poor physical health are on the rise. It is therefore important to develop guide robots for the elderly as well as for the visually handicapped.

A number of studies have been performed to identify a means of providing information on self location and obstacle location for the visually impaired. In an RFID-based study [2,3], a robot cart converts the location information which has been received through the RFID into voice data. As a navigation tool, various sensors are attached to clothes and then the acquired environment information with these sensors is delivered to the visually impaired through other sensory organs [4,5]. Many robots such as GuideCane [6,7], RoJi [8] and ROVI [9] have been studied in the area of autonomous robots for the visually impaired. In the above studies ultrasonic sensors are attached to the two-wheeled mobile robot as a means of detecting obstacles. The user uses the feedback force through a stick for avoiding obstacles when the steering wheel is acting. This becomes the cause that any ground bumps are directly transmitted to the user. In another study, eyeDog [10] is designed for the user to follow the robot with a dog leash. The shock to the robot is not transmitted to the user because of the flexibility of the leash, but the robot cannot acquire information about the relative location of the user and the robot.

Ultimately, our goal is to fabricate a human-friendly guide robot. Among the visually-impaired population, totally blind are only 15%. For this reason human-friendly guide robot which is in a cooperative relationship with the user, not just a common mobile robot which only follows control commands is more needed. This study addresses the fabrication of a new HRI sensor (Figure 1) with which a user's location and control volumes can be acquired at the same time. We will call this an HRI sensor the "intelligent lead". The intelligent lead proposed in this study and a 3D conceptual view about its application is shown in Figure 1.

Figure 1. Intelligent lead and its application.



The objective of the intelligent lead is to acquire a stable distance from the user to the robot, speed-control volume and turn-control volume, even when the robot platform with the intelligent lead is shaken on uneven ground. In this paper we explain the precise Extended Kalman Filter (EKF) procedure for this. The intelligent lead physically consists of a Kinect sensor, the serial linkage attached with eight rotary encoders, and an Inertial Measurement Unit (IMU), and their measurements are fused by the EKF. Since a user swings his/her arms back and forth while walking, it is very difficult to locate the user’s body using only his/her hand position information. A new sensor is needed to locate the user’s body, or one of the sensors used to detect obstacles can be used as well.

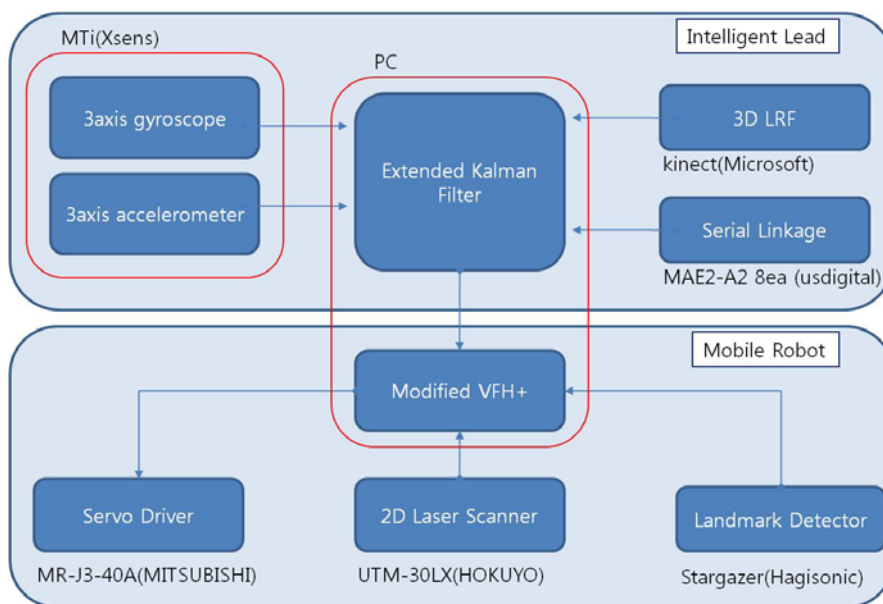
In many previous studies, an ultrasonic sensor was often used as a means for detecting obstacles. Recently, many studies on how to detect and avoid obstacles with 2D laser scanners, Kinect sensors and vision sensors [11,12] have been conducted. A vision system is light and cheap as a passive sensor. Among others, the Kinect sensor has been used to recognize a 3D object or environment in the robot industry due to its significant price competitiveness. Compared to the SR-3000 which is a commercial 3D laser scanner, it is less accurate. Since it is much cheaper, however, the Kinect sensor has been used to locate a user’s body in this study.

The intelligent lead is structured to fuse the data which has been acquired through the serial linkage, Kinect and IMU sensor with the EKF. This paper is organized as follows. Section 2 describes a configuration of the major hardware of the intelligent lead. Section 3 describes how to configure the serial linkage and an IMU integrator with the EKF, and identifies a process to obtain sensor output values from the user’s joint positions. Section 4 contains an analysis of the results of two tests which were been performed after attaching the proposed intelligent lead to a mobile robot. Finally, Section 5 contains conclusions and plans for the future.

2. Sensor System Architecture

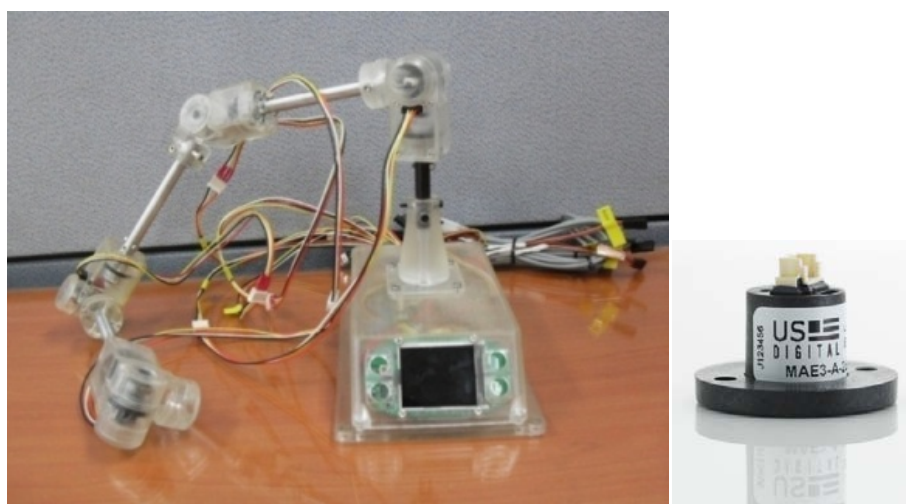
Figure 2 shows the block diagram of the intelligent lead and the mobile robot. The intelligent lead consists of an IMU, a serial linkage and the Kinect sensor.

Figure 2. Block diagram of a proposed intelligent lead and a mobile robot for testing.



The mobile robot consists of a 2D laser scanner, a landmark detector and a servo driver module. The intelligent lead output is used for generation of steering angle in modified VFH+ algorithm. The serial linkage is connected with rotary encoders or absolute encoders consecutively. The encoders which are selected should be small and precise. In general, however, larger encoders are more precise. MAE2-A2 made by USDigital Company is used in this study (Figure 3). The encoder is magnetic and an absolute encoder which provides PWM pulse with a 4,096 resolution at a 360° turn. Thus, 0.088° of accuracy per joint is found. In addition, the sampling rate of the PWM pulse is 250 Hz.

Figure 3. Fabricated serial linkage and absolute encoder.



The Kinect sensor was attached to the end of the serial linkage to measure a user's shoulder and torso position. The Kinect sensor acquires depth information by applying IR patterns to an object and measuring them with an IR sensor. Compared to a general camera, there is no interference of ambient light, and it also works at night. Usability at night is considered an important characteristic for human-friendly guide robots.

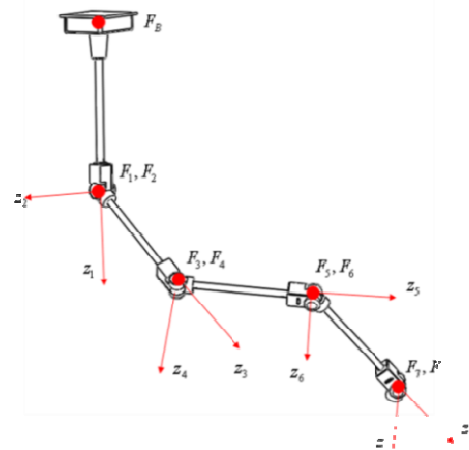
We acquired a user's shoulder and torso position from depth image with OpenNI library. If the base of the serial linkage is shaken because a robot passes over a rough surface or makes a dynamic motion, the vibration is directly transmitted in the coordinates on a user's hand. To compensate for the vibration of the robot's base frame, an IMU is installed in the area which is held by the user's hand. Accelerometer and gyroscope data are used in the time prediction procedures. For this, an MTi sensor made by Xsens Company with 100 Hz of sampling rate has been used.

The three sensors mentioned above have measurement errors. First, in the case of the low-price IMU, the drift of the gyroscope causes cumulative errors. We used the encoder connected in the serial linkage with small products to reduce weight. As the size of an encoder decreases, accuracy declines as well. Also, as the serial linkage gets longer, the amount of position and pose errors increases proportionally. The position and pose errors differ depending on assembly accuracy as well as the length of the link and the encoder accuracy itself.

The Kinect sensor can suffer from occlusions by smoke/fog or obstacles. In addition, the performance of the Kinect sensor decreases in a dynamic environment. Therefore, it is necessary to reduce error in the integration system through fusion instead of separately using these sensors. This kind of integration can improve accuracy and prevent output errors generated from sensor failures

from suddenly increasing. We made the serial linkage which consists of eight axes of rotation. We attached an absolute encoder to each axis. The link coordinate parameters of the serial linkage fabricated for the test is given in the table below (Table 1).

Table 1. Link coordinates parameters of the serial linkage.

Frame and Transformation Matrix	θ (deg.)	α (deg.)	a (mm)	d (mm)
	T_1^B	θ_1	0	50
	T_2^1	θ_2	90	0
	T_3^2	θ_3	-90	200
	T_4^3	θ_4	90	0
	T_5^4	θ_5	-90	200
	T_6^5	θ_6	90	0
	T_7^6	θ_7	-90	200
	T_8^7	θ_8	90	0
	T_R^8	0	-90	10

If T_s^8 is a transformation matrix from the IMU sensor coordinate frame to the last joint coordinate frame, the transformation matrix T from the IMU sensor coordinate frame to the base coordinate frame is calculated as follows:

$$T = T_1^B T_2^1 T_3^2 \dots T_8^7 T_s^8 \quad (1)$$

If we denote the entry in the i -th row and the j -th column of the matrix T as $T[i,j]$, the relative position and pose of sensor coordinates frame with respect to the navigation coordinate frame are calculated as follows:

$$P_x = T[1,4], P_y = T[2,4], P_z = T[3,4] \quad (2)$$

$$\theta = \tan^{-1} \left(\frac{-T[3,1]}{\sqrt{T[3,2]^2 + T[3,3]^2}} \right), \psi = \tan^{-1} \left(\frac{T[2,1]}{T[1,1]} \right), \phi = \tan^{-1} \left(\frac{T[3,2]}{T[3,3]} \right) \quad (3)$$

In Equation (3), the notation θ, ψ, ϕ mean roll, pitch, yaw angle respectively.

3. Intelligent Lead Sensor

3.1. Serial Linkage/IMU Integration with EKF

The objective of using the serial linkage, the IMU and the Kinect sensor is to produce a stable distance from a user to a robot, speed-control volume and turn-control volume, even when the robot platform in which the base frame of the serial linkage is installed is shaken by a rough surface. To achieve this objective we acquire the hand position from the EKF with the serial linkage and the IMU.

In Figure 4, the symbol P_{HS}, θ_{HS} represent position data and orientation data of the IMU coordinate frame which is attached to the user's hand respectively. The symbols ω and α represent angular

velocity and linear acceleration, respectively. The symbol P_H means a position data calculated from the EKF. In this section, a system dynamic nonlinear equation, system matrix, and an output matrix which are necessary to design the EKF to fuse both the serial linkage and the IMU are introduced.

Figure 4. Block diagram of EKF for acquiring hand position.

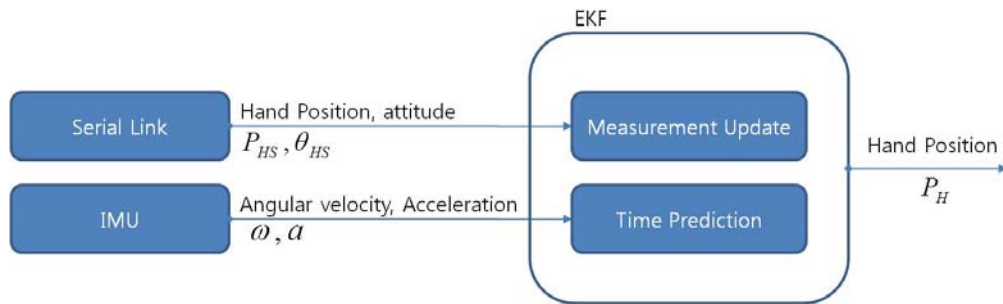


Table 2. Symbols used for EKF.

Symbol	Meaning	Symbol	Meaning
P_n	Hand position	Q	Process noise matrix
C_b^n	Direct Cosine Matrix	R	Measurement noise matrix
v_b	Velocity	x	State Vector
m	Mass	y	Output Vector
f_b	Force	θ	Euler angle
g_n	Gravity	A	System Matrix
q	Quaternion	H	Output Matrix
P	Covariance matrix	K	Kalman gain

Table 2 represents the related symbols used to explain the EKF. The linear speed equation of the hand is as follows:

$$\dot{p}_n = (C_n^b)^T v_b \tag{4}$$

where “n” refers to the definition in the navigation coordinate frame, and “b” refers to the definition in the body coordinate frame. The navigation frame is the same as the base frame of the serial linkage attached to the robot. The navigation coordinate frame is placed at the pinhole of the depth camera of the Kinect sensor. The body coordinate frame is located at the center of the IMU sensor. For the convenience of calculation, we assume that the hand coordinate frame coincides with the body coordinate frame. In the Equation (4), v_b is defined in the body coordinate frame. However, a position term, P_n is a data in the navigation coordinate frame at the composition of the system state which is defined below. Therefore, DCM (Direct Cosine Matrix) is multiplied.

The linear acceleration equation is as follows:

$$\dot{v}_b = \frac{1}{m} f_b + C_n^b g_n - \omega \otimes v_b \tag{5}$$

The quaternion equations of rotation velocity, which represent coordinate transformation from body coordinate frame to navigation coordinate frame, are expressed in a differential equation as shown below:

$$\dot{q} = \frac{1}{2} q \otimes \omega \quad (6)$$

The C_n^b used in Equations (4) and (5) can be stated with the normalized quaternion as shown below:

$$C_n^b = \begin{bmatrix} 1 - 2(q_2^2 + q_3^2) & 2(q_0q_3 + q_1q_2) & 2(-q_0q_2 + q_1q_3) \\ 2(-q_0q_3 + q_1q_2) & 1 - 2(q_1^2 + q_3^2) & 2(q_0q_1 + q_2q_3) \\ 2(q_0q_2 + q_1q_3) & 2(-q_0q_1 + q_2q_3) & 1 - 2(q_1^2 + q_2^2) \end{bmatrix} \quad (7)$$

To induce state space representation from the nonlinear Equations (4~6), state vector and output vector are defined as follows:

$$x = \begin{bmatrix} P_n \\ v_b \\ q \end{bmatrix}, \quad y = \begin{bmatrix} P_n \\ v_b \\ \theta \end{bmatrix} \quad (8)$$

Then, the system matrix and the output matrix are obtained as follows:

$$A = \begin{bmatrix} 0 & (C_n^b)^T & \frac{d(C_n^b v_b)}{dq} \\ 0 & -\omega \otimes & \frac{d(C_n^b g_n)}{dq} \\ 0 & 0 & -\frac{1}{2} \omega \otimes \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{d\theta}{dq} \end{bmatrix} \quad (9)$$

We need to first initialize x_0 , P_0 , Q , R and then to calculate time prediction procedure and measurement update procedure consequently. First we acquire roll, pitch and yaw angles from MTi sensor and then we convert Euler angle to quaternion q_0 with these data. The covariance matrix P_{n0} is set to an initially calculated value from the serial linkage. And the initial value of v_b is set to 0. From these values, the initial state and the initial covariance matrix are assigned as follows:

$$x_0 = \begin{bmatrix} P_{n0} \\ 0_{3 \times 3} \\ q_0 \end{bmatrix}, \quad P_0 = I_{10 \times 10} \quad (10)$$

The process noise matrix and measurement noise matrix were determined to avoid the divergence of the filter output by trial and error as follows:

$$Q = \begin{bmatrix} 0I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 4} \\ 0_{3 \times 3} & 0.1I_{3 \times 3} & 0_{3 \times 4} \\ 0_{4 \times 3} & 0_{4 \times 3} & 0.00001I_{4 \times 4} \end{bmatrix} \quad (11)$$

$$R = \begin{bmatrix} 0.8I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 4} \\ 0_{3 \times 3} & 0.2I_{3 \times 3} & 0_{3 \times 4} \\ 0_{4 \times 3} & 0_{4 \times 3} & 1I_{4 \times 4} \end{bmatrix} \quad (12)$$

In the above Equation (12), $I_{n \times n}$ represents n by n dimensional identity matrix.

The time prediction procedure shown in the block diagram above can be stated as follows:

- (1) With the current state vector and IMU sensor data, update the system matrix from the Equations (9).
- (2) Calculate the derivative of state vector by using the equation, $\dot{x}_{n-1} = Ax_{n-1}$
- (3) Update the current state vector by using the equation, $x_n = x_{n-1} + \dot{x}_{n-1}dt$
- (4) Normalize the quaternion among the updated states.
- (5) Update the P matrix by using the equation, $P_n = AP_{n-1}A^T + Q$

The measurement update procedure can also be stated as follows:

- (1) Determine the output matrix with the current quaternion from the Equation (9).
- (2) Measure the output, z_m with the hand position, P_{HS} and pose, θ_{HS} measured from the serial linkage.
- (3) Calculate the output error by using the equation, $e = z_m - Hx_n$
- (4) Update Kalman gain by using the equation, $K = P_n H^T [HP_n H^T + R]^{-1}$
- (5) Update the state vector by using the equation, $x_{n+1} = x_n + Ke$
- (6) Update the P matrix by using the equation, $P_{n+1} = P_n - KHP_n$

The first 3-terms of the state vector, which are updated through the above procedures, are the coordinates of the hand position viewed from the base coordinate frame of the serial linkage. As mentioned previously, we assume that the base coordinate frame of the serial linkage is placed at the pinhole of the depth camera of the Kinect sensor.

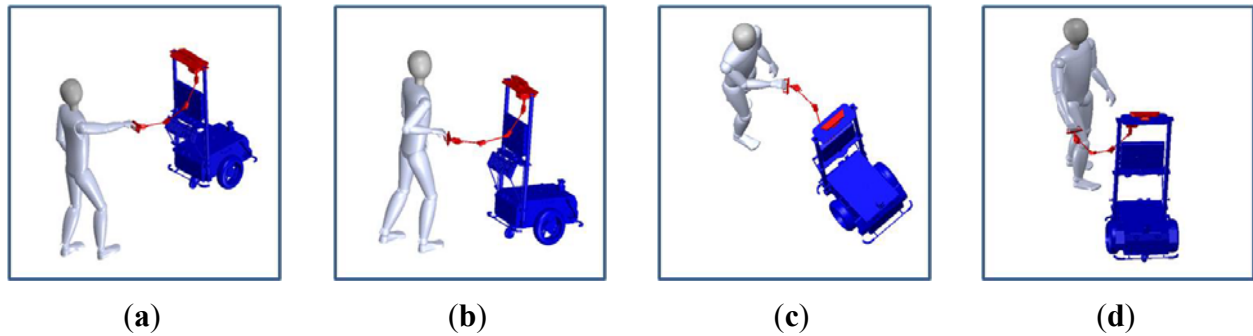
3.2. Intelligent Lead Output Generation

The intelligent lead proposed in this study acquires a user's left shoulder position, right shoulder position and torso position from the Kinect sensor and hand position with Serial linkage and IMU on a real-time basis.

The intelligent lead has two buttons. One is the start button and another is the trim button. If the user pushes the trim button current output volumes are saved in trim buffers. The intelligent lead has a hand position fault checker block. If the distance from the shoulder to the hand is larger than the sum of the upper arm length and the lower arm length, than the output volumes become zero. This prevents unwanted generation of output volumes when the user loses hold of the handle. If the user's hand position is obtained from the output of the EKF, the user's control intention should be investigated by using this data and the user's right shoulder, left shoulder and torso position data which have been obtained from the Kinect sensor. Because the size of walking steps, the amount of back and forth arm swings and the amount of movements to the left and right differ from individual to individual, it is necessary to get information on changes of the user's arm-joints while walking.

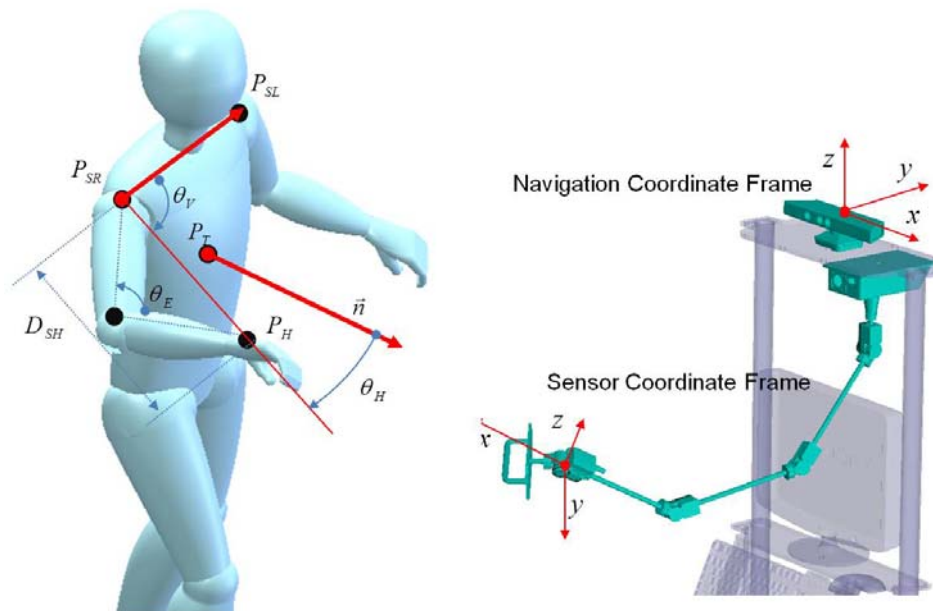
When people walk with a dog, it is more common to move the hand to the left and right to point out directions instead of changing the angle of the wrist. In addition, many people push their hand forward and pull down backward to increase or decrease the velocity. Figure 5 describes the actions for generating commands of speed increase, speed decrease, left turn and right turn respectively.

Figure 5. Control motions. (a) Speed Increase; (b) Speed Decrease; (c) Left turn; and (d) Right turn.



We assume the user is a right-hander. Figure 6 shows frames, angles and distance defined from the joint positions.

Figure 6. Frames, angles and distance definition.



The normal vector represents the direction of the plane in which a triangle is formed by the left shoulder position, the right shoulder position and the torso position. The symbol D_{SH} means the size of the line created by the right shoulder position and the hand position. The symbol θ_V means the angle between the normal vector and the line created by the right shoulder position and the hand position. The symbol θ_H means the angle between the line created by the right shoulder position and the hand position and the line formed by the left shoulder position and the right shoulder position. The distance from the shoulder to the hand is calculated as follows:

$$D_{SH} = \|P_H - P_{SR}\| \quad (13)$$

If the left shoulder position, P_{SL} is $(x_1 \ y_1 \ z_1)^T$, right shoulder position, P_{SR} is $(x_2 \ y_2 \ z_2)^T$ and torso position, P_T is $(x_3 \ y_3 \ z_3)^T$, then the normal vector \vec{n} of the plane that includes these three points can be calculated as follows:

$$\vec{n} = \left(\begin{array}{cc|cc} y_2 - y_1 & z_2 - z_1 & x_2 - x_1 & z_2 - z_1 \\ y_3 - y_1 & z_3 - z_1 & x_3 - x_1 & z_3 - z_1 \end{array} \right)^T \quad (14)$$

The angles defined above can be obtained using the law of cosine:

$$\theta_E = \pi \pm \arccos\left(\frac{L_{UA}^2 + L_{LA}^2 - D_{SH}^2}{2L_{UA} \cdot L_{LA}}\right) \quad (15)$$

$$\theta_H = \arccos\left(\frac{(P_{SL} - P_{SR}) \cdot (P_H - P_{SR})}{\|P_{SL} - P_{SR}\| \times \|P_H - P_{SR}\|}\right) \quad (16)$$

$$\theta_V = \arccos\left(\frac{\vec{n} \cdot (P_H - P_{SR})}{\|\vec{n}\| \times \|P_H - P_{SR}\|}\right) \quad (17)$$

If the height of the hand position is above the height of the torso position, we assume that the user has the intention to control the robot's direction or speed. In addition, if θ_V is above a certain limit value, then we conclude that the user bent his/her waist. We fixed the limited value to 45 degrees in the following test. In these cases the intelligent lead output forced to set 0. This assumption prevents unwanted output generation. The intelligent lead has the trim button to set trim values for speed control and turn control. The default trim value is as follows:

$$TRIM_{SPEED} = \sqrt{L_{UA}^2 + L_{LA}^2}, TRIM_{TURN} = \frac{\pi}{2} \quad (18)$$

Finally we get intelligent lead output values for control command as follows:

$$O_S = \begin{cases} D_{SH} - TRIM_{SPEED} & , \text{if } \theta_V < \frac{\pi}{4} \text{ and } P_H(z) > P_T(z) \\ 0 & , \text{otherwise} \end{cases} \quad (19)$$

$$O_T = \begin{cases} \theta_H - TRIM_{TURN} & , \text{if } \theta_V < \frac{\pi}{4} \text{ and } P_H(z) > P_T(z) \\ 0 & , \text{otherwise} \end{cases} \quad (20)$$

In Equations (19) and (20), O_S and O_T mean speed control value and turn control value respectively, $P_H(z)$ and $P_T(z)$ represent z-axis value of hand and torso position respectively. The distance from the user's body to the robot can be calculated simply with the mean of two shoulder positions and a torso position:

$$O_D = \frac{\|P_{SL} + P_{SR} + P_T\|}{3} \quad (21)$$

The intelligent lead generates O_D , O_S , O_T represented by the above Equations (19~21) simultaneously on a real-time basis. Figure 7 shows the summary block diagram of the proposed intelligent lead.

Figure 7. Block diagram of the intelligent lead.

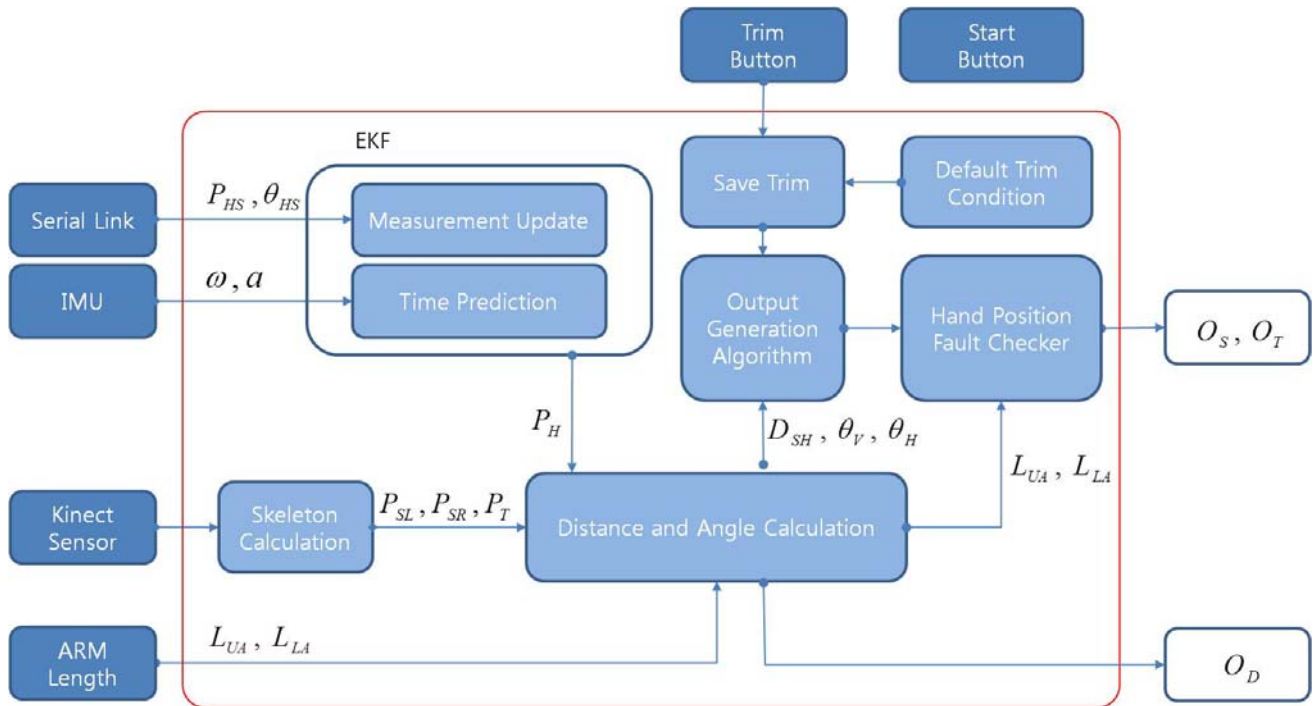


Table 3 lists the related symbols for generating intelligent lead output.

Table 3. Symbols for intelligent lead output generation.

Symbol	Meaning	Symbol	Meaning
p_H	Hand Position calculated from EKF	O_S	Speed Control Volume
p_{SL}	Left Shoulder Position	O_T	Turn Control Volume
P_{SR}	Right Shoulder Position	O_D	Distance Output from Body to Robot
P_T	Torso Position	D_{SH}	Distance from Shoulder to Hand
θ_E	Angle of Elbow	\vec{n}	Normal vector generated Body plane
θ_V	Angle of Vertical	L_{UA}	Length of Upper Arm
θ_H	Angle of Horizontal	L_{LA}	Length of Lower Arm


4. Experimental Results

4.1. Mobile Robot for Testing

Many studies have proposed strategies for robots to complete their tasks without any collision with obstacles and/or other robots. Especially the methods using the potential field [13] and the Vector Field Histogram Plus (VFH+) algorithm [14,15] have given diverse and useful results. In this study, the robot direction and speed were controlled based on the VFH+ algorithm by using the obstacle information acquired from a 2D laser scanner and current location information acquired from an IR landmark detector. The size of the robot and local map were fixed to be 0.8 m and 3×3 meters in the VFH+ algorithm.

The HOKUYO UTM30-LX was used as a 2D laser scanner to detect obstacles. Even though UTM-30LX has measure range from -135° to 135° , the data measured from -90° to 90° was used in this test. Its specifications are listed in Table 4. Hagisonic's Stargazer was used as an IR landmark detector. We firmly affixed a total of 16 landmarks as IR reflective materials to the ceiling, whose height is 2.5 meters. Landmarks were aligned with a 1 meter interval horizontally.

Table 4. Hokuyo UTM-30LX specifications.

	Parameter	Typical value
	Light Source	Laser diode ($\lambda = 785 \text{ nm}$), Class 1 (FDA)
	Detection Range	0.1 to 30 m, Max. 60 m
	Accuracy	0.1 to 10 m: $\pm 30 \text{ mm}$, 10 to 30 m: $\pm 50 \text{ mm}$
	Scan Angle	270°
	Angular Resolution	0.25° ($360^\circ/1,440$ steps)
	Scan Time	25 msec/scan
	Interface	USB2.0 (Full Speed)

To determine the steering angle with the VFH+ algorithm, it is first necessary to create a local map with 2D laser scanner data. Then from the local map, we obtain an obstacle vector, a Primary Polar Histogram, a Binary Polar Histogram, and a Masked Polar Histogram, in that order. Steering angle h_{vfh} is calculated based on the final Masked Polar Histogram. The desired steering angle is simply determined by the sum of h_{vfh} and O_T , and steering angle error value h_e is calculated by the difference desired angle and current angle as follows:

$$h_e = (h_{vfh} + O_T) - h_c \quad (22)$$

In the above equation, h_c is the current direction which is measured from the IR landmark detector and O_T is the desired differential direction which is generated from the intelligent lead. The position error, p_e is calculated as follows:

$$p_e = (p_d + O_S) - O_D \quad (23)$$

In the above equation, p_d represents the desired distance from a user to a robot which is fixed at 0.8 meters in this test. The symbol O_S is the desired differential speed which is generated from the intelligent lead output, and O_D is the measured distance from the user to the robot with the intelligent lead. For the test, a two wheel-type mobile robot was designed. Mitsubishi's 400W AC servo motor, 10:1 bevel gearbox, 13-inch wheels and PC (Intel i7-2600 CPU @3.4 GHz) were used. Figure 8 pictures the mobile robot and the intelligent lead sensor fabricated for the test.

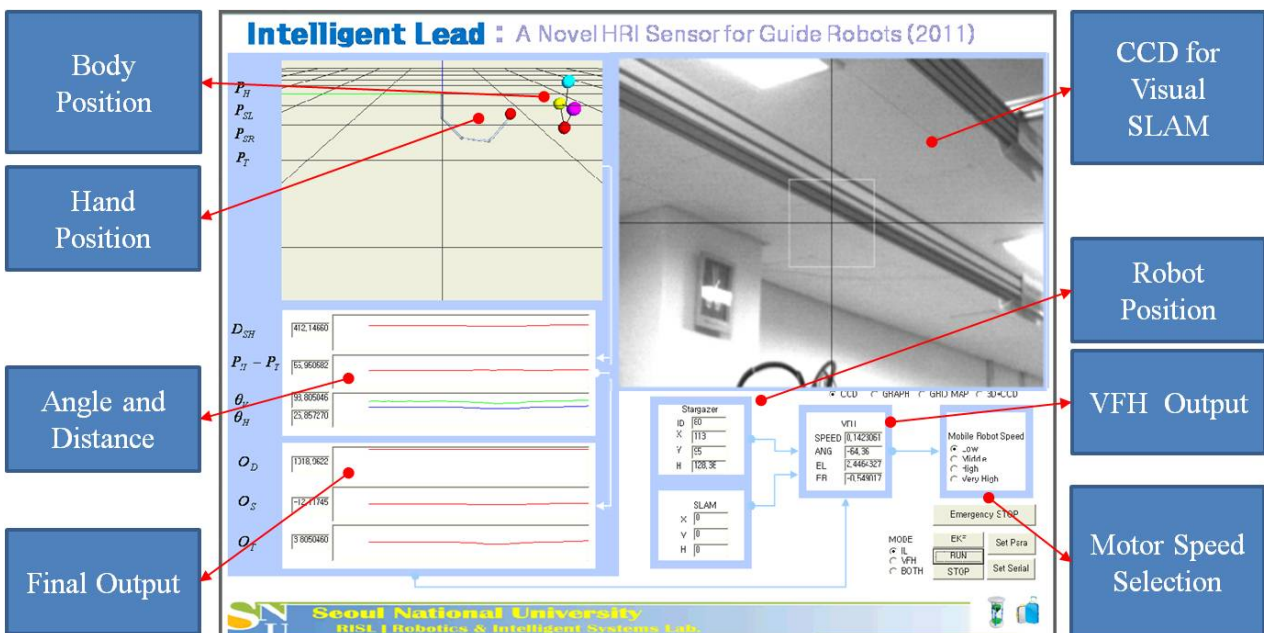
Figure 9 below shows the captured screen from the monitoring program made for our testing. We expressed calculated positions on the upper left side in 3D using the OpenGL. In Figure 9, the four dots on the top represent the user's head position, left shoulder position, right shoulder position, and torso position which have been detected by the Kinect sensor, respectively. In addition, the dot in the middle refers to the hand location which has been detected through the EKF. The values which were calculated from the four positions were marked on the bottom left. On the right side of the display a CCD camera image for visual tracking is shown, and on the bottom side we show current robot positions

calculated from SLAM and detected from the Stargazer landmark detector. The values calculated from the VFH+ algorithm are displayed and selection buttons for servo motor speed setting are arrayed.

Figure 8. Experimental system.



Figure 9. Monitoring program.



4.2. Test Results

In the first test, the control volumes were measured by moving the handle of the intelligent lead horizontally while the robot is stopped. Figure 10 shows acquired positions on the 3D viewer when we generated speed increase, speed decrease, left turn and right turn commands respectively.

Figure 10. Measured positions. (a) Speed Increase; (b) Speed Decrease; (c) Left turn; and (d) Right turn.

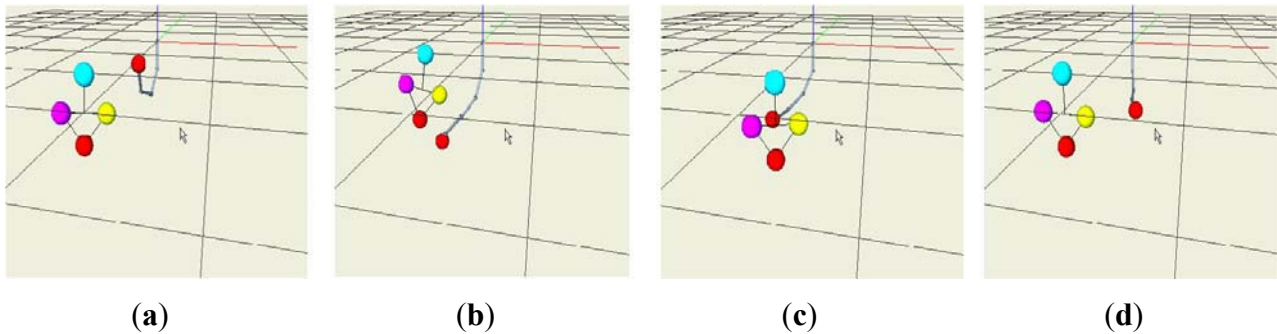
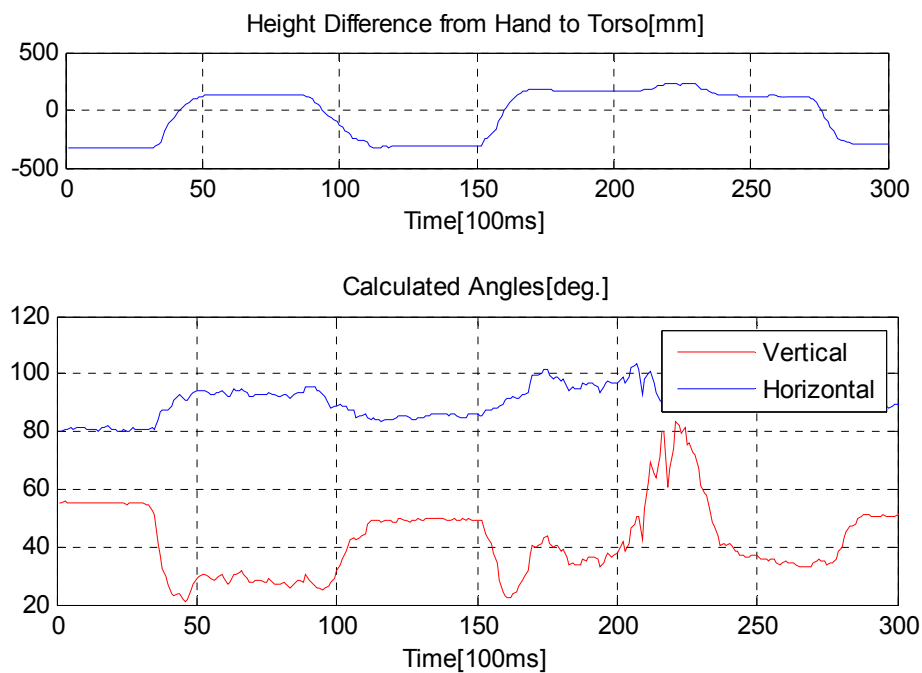


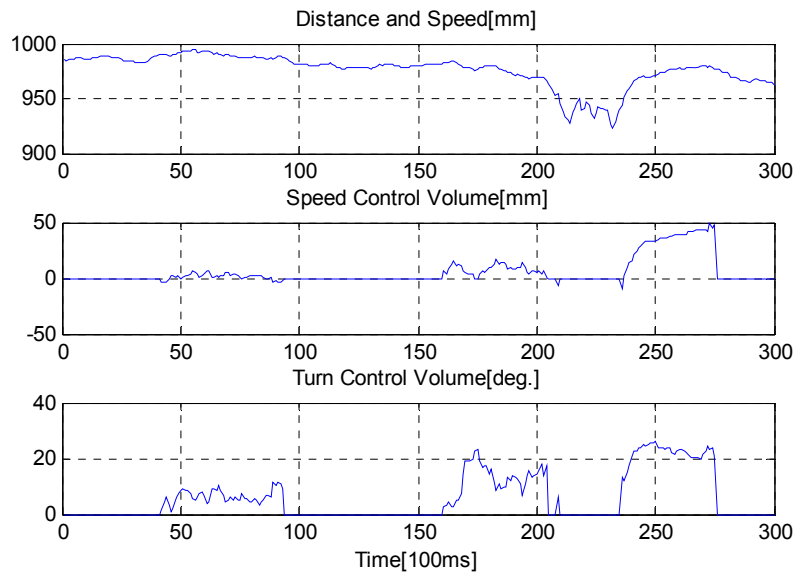
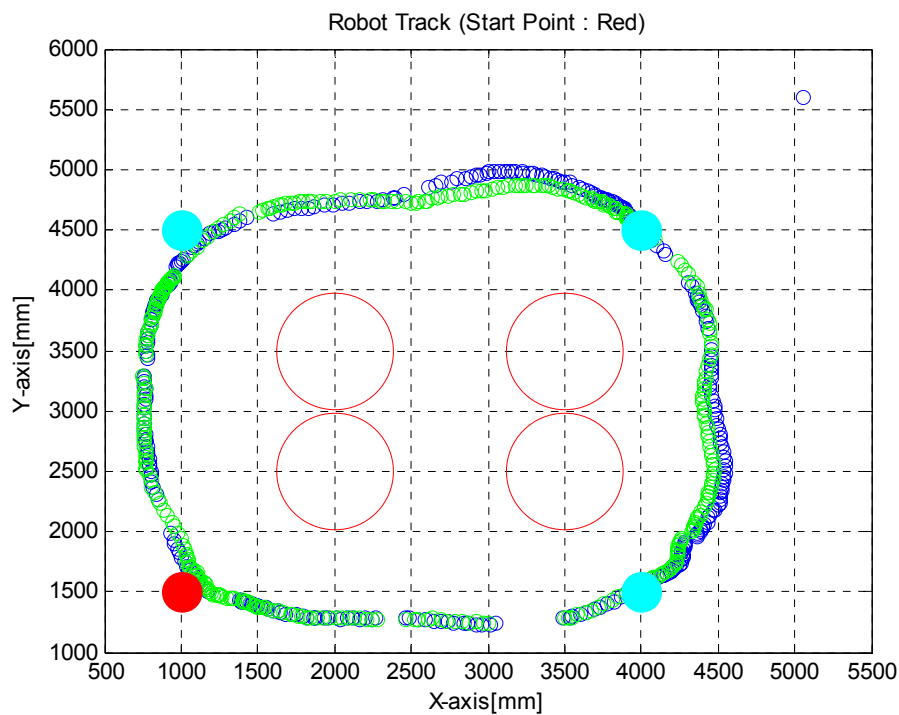
Figure 11 shows measured values when the hand position was shaking. We intended that θ_V changes via the defined limit value, 45 degrees and the height of the hand position goes from lower to upper than that of the torso position, and then from upper to lower.

Figure 11. Calculated value.



As shown in the Figure 12, in the region of 200~250 time tick, the user bent down at the waist. Since the height of the hand was above than that of the torso position, as θ_V was larger than 45 degrees the intelligent lead output was not generated. The distance from the user to the robot was always generated, regardless of θ_V value and the height of the hand position.

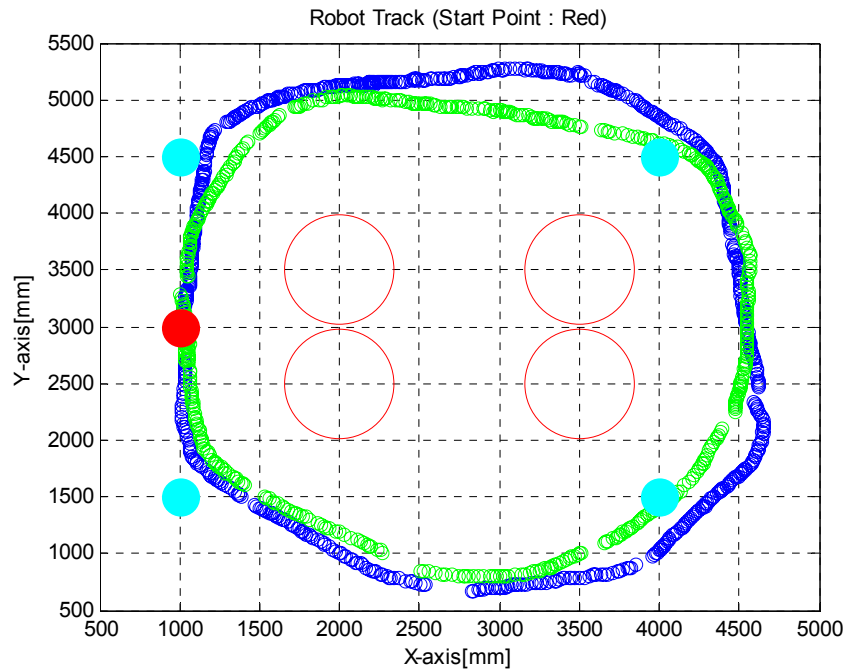
In the second test, the robot was programmed to move to (1, 1.5), (1, 4.5), (4, 4.5) and (4, 1.5) points with avoiding obstacles twice by using VFH+ algorithm. We located obstacles at (2, 2.5), (2, 3.5), (3.5, 2.5) and (3.5, 3.5) points. The obstacles were cylinders with 0.5 meter diameter. Figure 13 shows robot track measured with the Stargazer. Because of the alignment and flatness errors of landmarks, it is shown that the robot path was cut off whenever detected landmarks are changing. Nevertheless the robot reached the desired destinations by avoiding obstacles.

Figure 12. Sensor output.**Figure 13.** Robot track during VFH+ algorithm test.

When a robot is moving around following a user, it is difficult to use a joystick without a fixed joystick pad. The proposed intelligent lead can solve this problem. We show how to use intelligent lead as a joystick and a distance measure. In the third test, the ability of the robot equipped with the intelligent lead to reach a destination by avoiding obstacles was tested. The intention was for the robot to lead the users keeping 0.8 meter distance from the user. Our objective was to make a track twice by passing points at (1, 1.5), (1, 4.5), (4, 4.5) and (4, 1.5). We controlled the speed of the robot with the intelligent lead to make a first track within 120 seconds and then a second track within 80 seconds. And we generated turn control commands through the intelligent lead to arrive at the destinations and avoid obstacles. Figure 14 shows the robot track acquired from the third test. A red point means a start

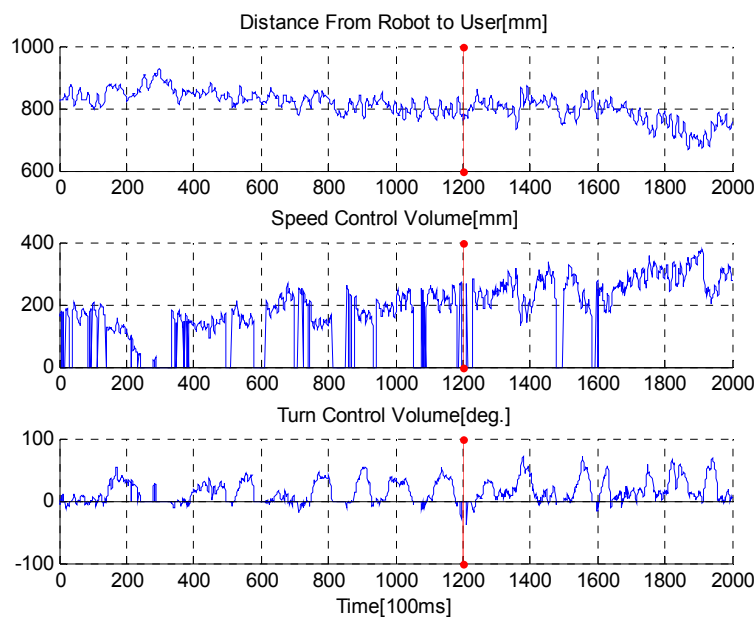
point and green points mean en route points or destination points, respectively. The result shows that we could reach the destination points keeping pre-defined distance from the robot.

Figure 14. Robot track acquired from the third test.



Output data from the intelligent lead during the test is shown in Figure 15. Mean distance between user and robot was 0.8101 meters. We confirmed that the proposed intelligent lead could be used as a new HRI joystick. Moreover, our results suggested that motion of humans could interact with robots using the proposed intelligent lead. During many tests, the OpenNI algorithm for searching skeleton fails occasionally at turning intervals. This is because the user gets out of the FOV due to the limited FOV. In future work, we need to convert the software to be able to solve this problem.

Figure 15. Sensor output acquired from the third test.



5. Conclusions

In this paper, we introduced the intelligent lead as a new HRI sensor. To detect the user's hand position, we made a serial linkage consisting of eight rotary encoders and then we constructed the intelligent lead with the Kinect sensor, the serial linkage and MTi sensor as an IMU. Their measurements were fused by an EKF. A mobile robot was designed to test the performance of the proposed intelligent lead and the related positions are monitored by calculating in 3D with the OpenGL. Several tests were conducted to verify that the mobile robot with the intelligent lead is capable of achieving its goal points while maintaining the appropriate distance between the robot and the user. It is shown that the intelligent lead acquires a stable distance from the user to the robot, speed-control volume and turn-control volume on a real-time basis. Moreover, our results suggested that the motion of humans could interact with the robot by the proposed intelligent lead in mobile environments such as those where the robot and the user are moving at the same time. During many tests, the OpenNI algorithm for searching skeleton failed occasionally at turning intervals for reasons of the limited FOV. In the future work, we need to convert the software to be able to solve this problem.

Acknowledgments

This work was supported in part by a Korea Science and Engineering Foundation (KOSEF) NRL Program grant funded by the Korean government (No.R0A-2008-000-20004-0), and in part by the ASRI, and in part by the Brain Korea 21 (BK21) Project, and in part by the Industrial Foundation Technology Development Program of MKE/KEIT [Development of CIRT (Collective Intelligence Robot Technologies)].

References

1. World Health Organization. Media Centre, Visual impairment and blindness. Available online: <http://www.who.int/mediacentre/factsheets/fs282/en/> (accessed on 20 March 2012).
2. Kulyukin, V.; Gharpure, C.; Nicholson, J. RoboCart: Toward Robot-Assisted Navigation of Grocery Stores by the Visually Impaired. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Canada, 2–6 August 2005; pp. 2845–2850.
3. Kulyukin, V.; Gharpure, C.; Nicholson, J.; Pavithran, S. RFID in Robot-Assisted Indoor Navigation for the Visually Impaired. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, 28 September–2 October 2004; pp. 1979–1984.
4. Pradeep, V.; Medioni, G.; Weiland, J. Robot Vision for the Visually Impaired. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, San Francisco, CA, USA, 13–18 June 2010; pp. 15–22.
5. Hesch, J.A.; Mirzaei, F.M.; Mariottini, G.L.; Roumeliotis, S.I. A 3D Pose Estimator for the Visually Impaired. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MI, USA, 11–15 October 2009; pp. 2716–2723.
6. Ulrich, I.; Borenstein, J. The GuideCane—A Computerized Travel Aid for the Active Guidance of Blind Pedestrians. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Albuquerque, NM, USA, 21–27 April 1997; pp. 1283–1288.

7. Ulrich, I.; Borenstein, J. The GuideCane—Applying mobile robot technologies to assist the visually impaired. *Proc. IEEE Trans. Syst. Man Cybern. Part A* **2001**, *31*, 131–136.
8. Shim, I.; Yoon, J.; Yoh, M. A human robot interactive system—RoJi. *Int. J. Control Autom. Syst.* **2004**, *2*, 398–405.
9. Melvin, A.A.; Prabu, B.; Nagarajan, R.; Illias, B. ROVI: A Robot for Visually Impaired for Collision-Free Navigation. In *Proceedings of the International Conference on Man-Machine Systems*, Penang, Malaysia, 11–13 October 2009; pp. 3B51–3B56.
10. Galatas, G.; McMurrough, C.; Mariottini, G.L.; Makedon, F. eyeDog: An Assistive-Guide Robot for the Visually Impaired. In *Proceedings of the International Conference on Pervasive Technologies Related to Assistive Environments*, Crete, Greece, 25–27 May 2011.
11. Se, S.; Lowe, D.; Little, J. Vision-based Mobile Robot Localization And Mapping Using Scale-Invariant Features. In *Proceedings of the International Conference on Robotics & Automation*, Seoul, Korea, 21–26 May 2001; pp. 2051–2058.
12. Chu, T.; Guo, N.; Backén, S.; Akos, D. Monocular camera/IMU/GNSS integration for ground vehicle navigation in challenging GNSS environments. *Sensors* **2012**, *12*, 3162–3185.
13. Ge, S.S.; Cui, Y.J. New Potential Functions for Mobile Robot Path Planning. *IEEE Trans. Rob. Autom.* **2000**, *16*, 615–620.
14. Ulrich, I.; Borenstein, J. VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Leuven, Belgium, 18–21 May 1998; pp. 1572–1577.
15. Ringdahl, O. Path Tracking and Obstacle Avoidance Algorithms for Autonomous Forest Machines. Master Thesis, Umea University, Umea, Sweden, 2003.

© 2012 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).