

Customized embedded system design for lower limb rehabilitation patients

Chae-eun Lee*, Inn-chae Yeo*, Inshil Doh**

*Dept of Computer Science, Ewha Womans University

**Dept of Cyber Security, Ewha Womans University

nuguziii@ewhain.net, ic0715@ewhain.net, isdoh1@ewha.ac.kr

Abstract—There are a number of researches about combining games with rehabilitation training to induce the interest of the patients. However, related studies mainly focus on upper body rehabilitation patients and have not dealt with accuracy of sensors. In this work, we aimed to improve the reliability of game results by implementing four step algorithm that minimizes the errors caused by sensors for lower body rehabilitation patients. Furthermore, we implemented a customized game system for the patient by collecting patient data and allowing the embedded smart car to move in proportion to the patient's movement.

Keywords— Lower limb rehabilitation, Embedded system, User customized, Correction algorithm

I. INTRODUCTION

In modern Korean society, the number of elderly people is increasing due to the aging trend of population. Accordingly, patients with gonarthrosis, which is one of the chronic symptoms of the elderly, are also increasing. These symptoms are lowering the quality of their life by restricting activities necessary for daily life [1]. Similarly, children with cerebral palsy are also experiencing discomfort due to movement disorders. Both the elderly people and the children suffer from diseases that restrict walking, which is considered to be more important than any other abilities in that it is related to independence and quality of life [2]. At this point, rehabilitation therapy is essential to improve the strength of the patient, and bring better life [3][4]. However, general rehabilitation treatments are not enough to attract the patient's interest in that they should repeat the same behavior for several times. In order to make up for this problem, there have been a lot of researches that developed the rehabilitation embedded system combining with game. These studies mostly focused on the patients who need the upper extremity rehabilitation and assumed that the sensor extracts the correct value [5]. In this study, however, we focused on the patients who have inconvenience in walking. Also, we implemented the algorithm that minimizes the error caused by the sensor to supplement the limitation of the existing studies. Furthermore, we realized a customized game system for the patients.

Our paper is structured as follows: Section 2 describes system design consideration and the implemented algorithm based on it. Section 3 describes the system development environment and verifies the system implementation results.

II. REHABILITATION SYSTEM DESIGN

A. Design Consideration and Contents Processing Method

The purpose of our study is to develop a rehabilitation assist system using an Arduino-based embedded smart car. We considered three design factors: interest, customization, and achievement. First, "Tricky mode" was designed for user's interest and it also enhances the visual impact. Second, the rehabilitation training time can be selected and the movement of the smart car is changed according to the user. Third, when the momentum is saved, the user can check the record with the graph through the app.

The detailed game rules are shown in Figure 1, Distance implies the distance between the user and the smart car. When the user approaches within 60cm, the smart car selects direction randomly, moves away from the user, and the success value increases. In addition, status of the smart car can be detected by LED and Buzzer.

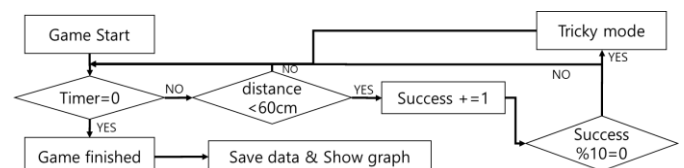


Figure 1. Game contents overall algorithm

B. Operation Algorithm

1) Analysis of Ultrasonic Sensor Values

In this study, Smart Car's ultrasonic sensor was used to implement 'hide and seek' which is the key function in our contents. Figure 2 shows the location of the 12 ultrasonic sensors attached to the HBE-Smart Car. The maximum value that can be measured by each ultrasonic sensor is 250 cm, and it is received once every 0.125 sec.



Figure 2. Ultrasonic sensors attached to smart car

The test was performed to see the change of the value of each ultrasonic sensor when the user approached the Smart Car's 9th sensor. Figure 3 shows the ultrasonic sensor value (cm) taken 10 consecutive times, and it was reconfigured to reflect various situations that occurred through several tests. As a result of the test, the values of 250 and 9 appeared sporadically like 5th sensor, and it was found that the sensor values were not accurate depending on the surrounding environment. On the other hand, the 9th sensor value that the user is approaching is decreasing at a constant interval.

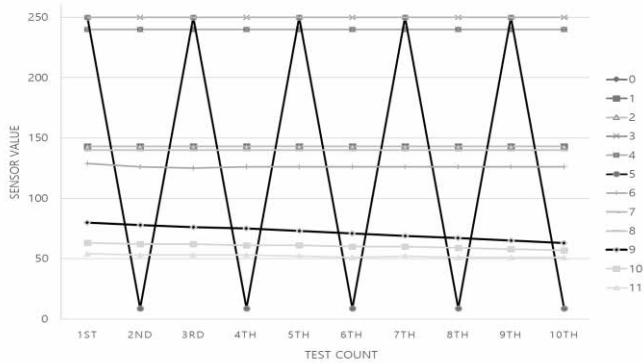


Figure 3. The sensor value of each Ultrasonic sensor

We wanted to avoid the unnecessary movement of the Smart Car as shown on the 5th sensor and improve the accuracy of the Smart Car's movement. Therefore, we designed an algorithm that minimizes the error by identifying the tendency of the sensor value to decrease, such as the 9th sensor.

2) Ultrasonic Sensor Measurement Algorithm

Considering these various variables, our study implemented 4 steps of ultrasonic sensor measurement algorithm as shown in Figure 4. The first step is to save the values and assign weights. Table 1 is introduced to explain this.

TABLE 1. TERMINOLOGIES USED IN ALGORITHM

RX_buf [12]	An array that moves already received values to RX_before and receives the newly incoming values from # 0 to 11 of the ultrasonic sensors
RX_before [12]	An array that stores the values of sensor # 0 ~ 11 of ultrasonic sensors from RX_buf
score[12]	An array that stores each sensor's score weighted by the difference between RX_before and RX_buf
score_max	Store the largest value in the score array
score_std_value	This is criteria value which is 25 for deciding whether the score_max value is appropriate for moving to the 3rd step.
RX_std_value	This is criteria value which is 60 for deciding whether the user is located close enough to the smart car.
RX_min	Store the smallest value among the ultrasonic front sensors (#0 ~ 6) when obstacle is detected by the rear sensors
avoid_front_wall()	The function to execute if there is an obstacle in front of the smart car when the user coming from behind
control_Speed()	The function that changes the speed of the smart car according to the user's approaching speed

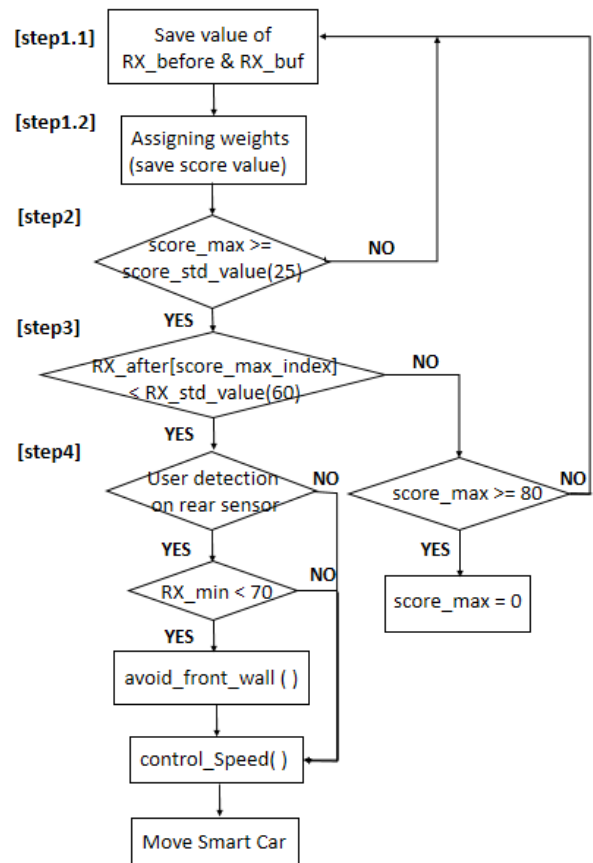


Figure 4. Flow Chart of Ultrasonic Sensor Measurement Algorithm

TABLE 2. METHOD OF GIVING WEIGHTS

	0	1	2	3	4	5	6	7	8	9	10	11
RX_before	240	143	143	250	240	250	129	250	140	80	63	54
RX_buf	240	143	143	250	240	9	126	250	140	78	62	53
RX_before - RX_buf	0	0	0	0	0	241	3	0	0	2	1	1
score	0	0	0	0	0	1	5	0	0	5	5	5

In Table 2, RX_before and RX_buf imply sensor values, and score implies weight. The reason for weighting is to compensate for the sporadic occurrence of inaccurate values such as 250 or 9(cm) in ultrasonic sensor. At this time, if the decreasing trend of the sensor value shows a meaningful result, we considered the user is approaching to a specific sensor, so a higher weight/score is given.

In the first step, the values measured by the ultrasonic sensor are stored in the RX_before and RX_buf arrays in the order. When the difference value from the previous sensor exceeds 3(sharply decrease), the value of 1 is stored in the score array. Conversely, when the difference value between the previous sensor is less than 3(gradually decrease), it is determined that the data value is reliable, and a value of 5 is stored in the score array. When the newly sensed value comes in, it is stored in

RX_buf and original RX_buf values move to RX_before. The above process is repeated and the accumulated value is stored in the score array.

The second step is to check whether the largest weight value exceeds a certain level. That is, the algorithm is performed for largest value in score array only when its value exceeds 25. This is because, the higher the score is, the more likely the sensor values decrease with regular interval.

The third step is to check whether the user is within 60cm. This step is introduced to exclude the highly weighted sensor even if it is far from the smart car.

The final step is to check for obstacles in front of the smart car before it runs away from the user to prevent a collision. When the user is detected by the rear sensor, smart car identifies the closest sensor from the wall among five front sensors. Check if the distance between closest sensor and the wall is 70cm or less. If satisfied, execute the function to avoid the collision with the wall and let the smart car escape from the user.

C. Customized Algorithm for Arthritis Patients

1) Speed Variation of Embedded Smart Car

When defining the speed of the smart car according to the user's approaching speed, we bundled four sensor values into one to minimize sporadic errors in the sensor, such as 9 or 250. Figure 5 was introduced to explain why we chose to bundle four sensors instead of two or eight sensors. The response time in Figure 5 implies the time delayed while returning from the control_Speed() function when the smart car has to be moved. When two sensors were combined, the accuracy rate was low. In contrast, when eight sensors were combined, the response time was slow. Therefore, we decided to make 4 sensors in one cluster.

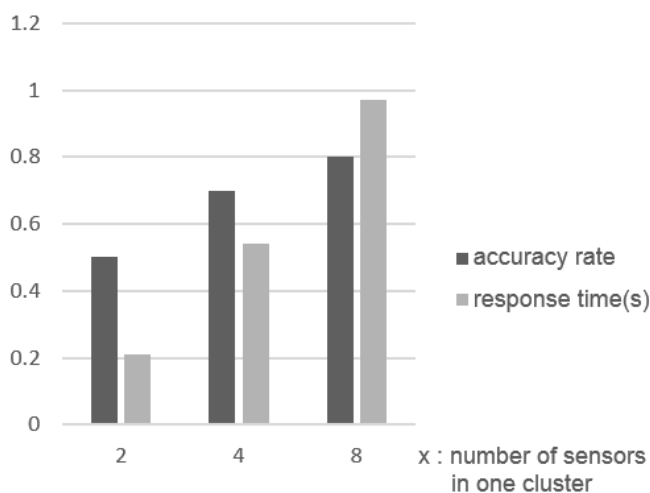


Figure 5. comparison of accuracy rate and response time among 3 cases : bundling 2,4, and 8 sensors

When setting the speed of the Smart Car, we focused on the difference between bundled values. Suppose that the eight sensor values that come in sequentially are a, b, c, d, e, f, g,

and h. The difference of the sensor values is calculated as $(a + b + c + d) - (e + f + g + h)$. The small value implies that there is little difference in the change of the approaching speed of the user. So we let the smart car also move slowly. That is, the smart car moves in proportion to the difference of sensor values.

2) Measurement of User's Momentum

The measurement of the user's momentum is based on the fact that the distance traveled by the Smart Car is proportional to the distance traveled by the user. That is, the user's momentum is assumed to be equal to the moving distance of the Smart Car. The method of measuring the moving distance of the Smart Car is as follows.

The signal from the Encoder mounted on the rear wheel of the Smart Car (pulse) is connected to the ATmega2560 pin. The right motor was used. The number of signals from the right rear wheel was counted and stored in the variable, Encoder_value_R and the 390 counts is one rotation. The calculated momentum formula is " $((\text{Encoder_value_R}) / 390 * (\text{double}) 50) / 100(\text{cm})$ ", and the user can visually check his / her momentum through the graph.

III. IMPLEMENTATION OF REHABILITATION EMBEDDED SYSTEM

A. System Development Environment



Figure 6. HBE-SmartCar

To implement this system, HBE-Smart Car embedded system device, a car-type robot having ATmega2560 as main board was used. Detailed specifications can be found in the Table 3 and Figure 6.

TABLE 3. HBE-SMARTCAR HARDWARE AND SOFTWARE SPEC

	item	specification
Hardware	Size / Weight	245mm x 380mm x 70mm / 5kg
	Sub-System Main Controller	Atmega2560 (Google ADK Platform with Arduino Mega2560)
	Connectivity - Bluetooth	On-Board Bluetooth (FB155BC)
	Digital Voltmeter	ATmega8
Arduino Firmware	Arduino IDE	AndroX Studio, Arduino IDE, ArduBlock
	OSEK/VDX IDE	Cygwin, WinAVR
OSEK/VDX Firmware	Function Test Firmware	Motor/Encoder, Ultrasonic, Infrared, LED, Compass Sensor, gyro sensor, acceleration sensor, Buzzer, UART/Bluetooth
	Intelligent Robot Test Firmware	Remote control between Smart device with Bluetooth and HBE-SmartCar. Obstacle avoidance self-driving with ultrasonic sensor. Autonomous vehicle recognition with vision. Specified path driving with Infrared sensor

Programming environment was developed using AndroX Studio IDE of Arduino platform. After building the development environment with the Android device Nexus7, we implemented the application with the block-based tool, MIT App Inventor program. Block diagram can be shown in Figure 7.

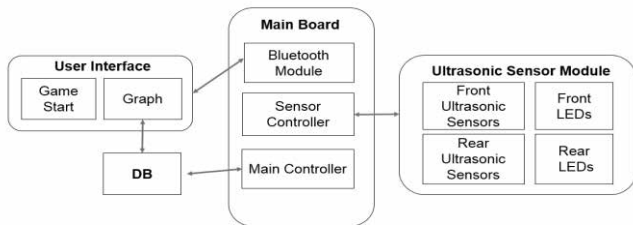


Figure 7. Block Diagram of System Overview

B. Rehabilitation System Implementation

Since this study aims to rehabilitate patients with degenerative arthritis, we made it as simple and accessible as possible. As shown in Figure 8, the main screen consists of the "Game Start" menu (second image), and the "Graph" menu(third image), which shows the rehabilitation training data at a glance. In Graph menu, you can check the last 5 records.

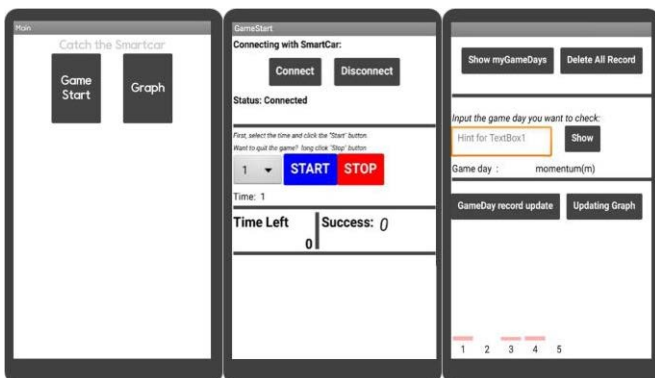


Figure 8. Application Screen

The overall sequence of game content is shown in Figure. 9.

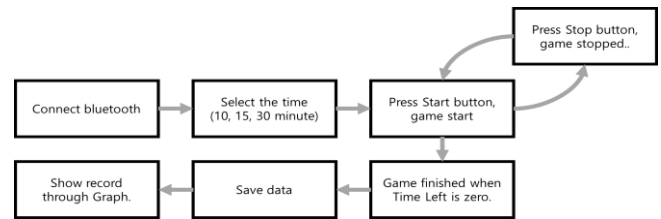


Figure 9. Overall Game Flow Chart

C. Experiment Results

Figure 10 is a graph showing the difference of performance between the system applying the ultrasonic sensor measurement algorithm and the system not applying it. At this time, the more accurate the smart car recognizes the location of the sensor that the user is approaching, the higher the accuracy rate is given. As the number of movements of the Smart Car increases, the accuracy rate is cumulatively represented by the vertical axis of the graph.

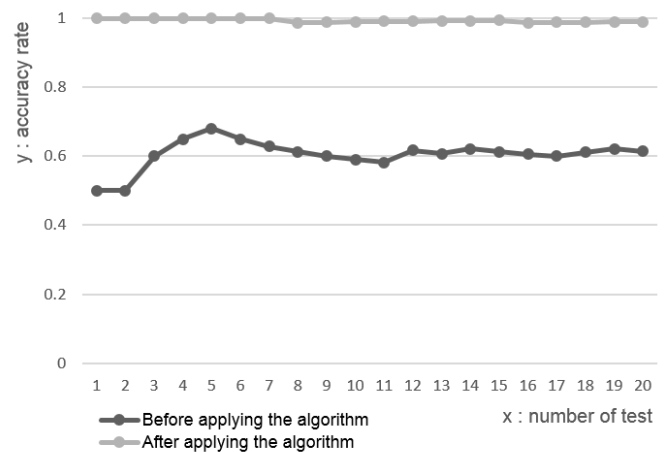


Figure 10. Comparison between the system that does and does not apply algorithm

IV. CONCLUSIONS

In this study, we implemented a system for rehabilitation of degenerative arthritis patients utilizing embedded using Smart Car. The patients are able to participate in rehabilitation treatment with interest through the game. Also, by applying the patient-customized system, the smart car moves according to the patient's speed, and the patient can easily check his or her records through application. In addition, we applied weighted algorithm to minimize errors caused by sensors and showed the accuracy of the result has been improved.

In the future, we will build a social network among patients through applications by producing servers, create more customized systems by applying the deep learning concept. In addition, by attaching various sensors to the patient's legs, the walking posture can be corrected more effectively.

Womans University, Seoul. Her research interests include wireless network, wired/wireless network security, and IoT security.

ACKNOWLEDGMENT

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the Seoul Accord Vitalization Program(IITP-2017-2012-0-00442) supervised by the IITP(Institute for Information & communications Technology Promotion). Inshil Doh is the corresponding author.

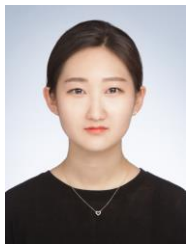
REFERENCES

- [1] Yun, Cheonggu, An, Changsik, "The Effect of Exercise Program on Pain and Quality of Life for Patients with Knee Osteoarthritis: A Systematic Review and Meta-Analysis" , J Muscle Jt Health Vol. 21 No.3, 173-183, December, 2014
- [2] Lee, Byoung Hee, Kim, Seong Yeol, Seo, Hyun Du, Yoo, Ha Na, "The Clinical Usefulness of Augmented Reality Based on Exercise Program for Spatiotemporal Parameters in Children with Cerebral Palsy" , Journal of Special Education & Rehabilitation Science Vol. 48, No. 4, Pp.211-230, 2009
- [3] Chae-Gil Lim, Mi-Kyoung Jun, " The Effects of Rehabilitation Exercise Program on Muscle Strength and Pain in Elderly with the Knee Osteoarthritis" , J Kor Soc Phys Ther 2006; 18(6): 59-65
- [4] JINWAN Kim, " The Effect of Robot-Assisted Game Training on Upper Extremity Function in Stroke Patient " ,Dept. of Medicine Graduate School, Dong-A University Busan, Korea, 2014
- [5] Sung-Wook Shin, Sung-Hoon Jeong, Sung- Teak Chung, C.J.Lim, "Balance ball controller and Rehabilitation Contents Development using Smart Multisensor," Korean Society For Computer Game, Vol. 27, No.3, 30 September 2014, pp. 135-141



Chae-eun Lee Author

Born in Korea on March 14, 1996.
Department of Computer Science, Ewha Womans University, Korea (undergraduate)
Primary interests: Embedded System, Data Analysis



Innchae Yeo Author

Born in Korea on July 15, 1995.
Department of Computer Science, Ewha Womans University, Korea (undergraduate)
Primary interests: Embedded System, Artificial Intelligence



Inshil Doh Author

Inshil Doh received the B.S. and M.S. degrees in Computer Science at Ewha Womans University, Korea, in 1993 and 1995, respectively, and received the Ph.D. degree in Computer Science and Engineering from Ewha Womans University in 2007. In 2008 and 2009, she took the post-doc program at Seoul national university. From 1995-1998, she worked in Samsung SDS of Korea to develop a marketing system. She was a research professor of Ewha Womans University in 2009~2010 and of Sungkyunkwan University in 2011. She is currently an assistant professor of Software department at Ewha