World Academy of Science, Engineering and Technology
International Journal of Mechanical and Mechatronics Engineering
Vol:11, No:8, 2017

Hand Controlled Mobile Robot Applied in Virtual Environment

Jozsef Katona, Attila Kovari, Tibor Ujbanyi, Gergely Sziladi

Open Science Index, Mechanical and Mechatronics Engineering Vol:11, No:8, 2017 publications.waset.org/10007702/pdf

***Abstract*—**By the development of IT systems, human-computer interaction is also developing even faster and newer communication methods become available in human-machine interaction. In this article, the application of a hand gesture controlled human-computer interface is being introduced through the example of a mobile robot. The control of the mobile robot is implemented in a realistic virtual environment that is advantageous regarding the aspect of different tests, parallel examinations, so the purchase of expensive equipment is unnecessary. The usability of the implemented hand gesture control has been evaluated by test subjects. According to the opinion of the testing subjects, the system can be well used, and its application would be recommended on other application fields too.

***Keywords*—**Human-machine interface, hand control, mobile robot, virtual environment.

I. Introduction

HUMAN-machine relationships mean the interaction between users and different devices, most likely computers, where the tool of connection between human and computer is the User Interface (UI), which contain both the software and the hardware. The human-machine interaction examines the human and the computer together.

The usability of devices used nowadays is more important than features available in them, furthermore, the creation of such control panels, like human-computer interfaces (HCI) are getting more and more important, overshadowing features more emphasized in the past.

The proper design of human-machine interaction may still have its limits, even unwanted misunderstandings might occur, so interface designers need to create such multimodality computer systems and interfaces that help minimizing these misunderstandings, and that are easier-to-use, their usage can be learnt without professional knowledge, achieving the smooth handling of interfaces. As a summary, in the future, the design of such an interface would be important, such that can be integrated into devices and enable good handling; nowadays, we still do not have a universal solution.

In order to solve the problem, human-machine interface based solutions are at the centre of researches on such crucial areas like modernization of nuclear power plants and improvement of their security [1]-[3], supporting the everyday life of disabled people by the remote control of equipment [4]-[7], control of vehicles even in real-time by remote control [8]-[10], and solving simulation tasks [11]. Besides that, we can also meet that on such fields, like the Internet of Things [12], or virtual reality [13], [15].

The reason for the research is to create such a human-machine interface based test system, which is able to move a mobile robot by gesture control. For achieving the goals of the research, the so-called Leap Motion gesture controller unit, having been developed by Leap Motion Inc. since 2010, was applied, and an interactive 'native' graphic programming environment, the Robotino View, the Robotino Sim Demo simulation software, that realistically models the motion of a mobile robot in a virtual 3D environment [16], [17], a self-developed software that processes, evaluates and transforms data arriving from the gesture controller unit into the information required to move the mobile robot. The developed virtual test environment also has many opportunities for educational aspects [18], [19]. At the end of the article, the evaluation of the usability of the designed test environment, and conclusions made from the results are published.

The article presents an attempt to integrate independently implemented software artifacts into one system, using existing parts: Leap Motion controller and Festo Robotino mobile robot device. Data from Leap Motion is used to navigate the Festo Robotino mobile robot.

II. Applied Devices and Technologies

In order to comply with the requirements of the research, modern architecture devices and technologies have been applied and introduced in detail below.

A. Leap Motion

There are several working analogical motion sensing solutions such as the Kinect, and the Xtion Pro gesture recognition could be comfortable, but these devices focus more on body action.

The Leap Motion Inc. has been developing the so-called Leap Motion controller since 2010, whose goal is to reform human-computer interaction; moreover, the Leap Motion is small, and has low power dissipation and high precision [14].

The Leap Motion can serve as an excellent replacement of Kinect as a fast development opportunity for desktop applications of motion sensing technology.

B. System's Design

The control unit's design is a rounded aluminum cuboid, equipped with glossy black cover and rubber sole, its size is

Jozsef Katona is with the University of Dunaújváros, Dunaújváros, CO 2400 Hungary (corresponding author, phone: +3625-551-605, e-mail: katonaj@uniduna.hu).

Attila Kovari, Tibor Ujbanyi and Gergely Sziladi are with the University of Dunaújváros, Dunaújváros, CO 2400 Hungary (e-mail: kovari@uniduna.hu, ujbanyit@uniduna.hu, sziladig@uniduna.hu).

The project is sponsored by EFOP-3.6.1-16-2016-00003 founds, Consolidate long-term R and D and I processes at the University of Dunaujvaros.

Open Science Index, Mechanical and Mechatronics Engineering Vol:11, No:8, 2017 publications.waset.org/10007702/pdf

smaller than an average computer mouse, it is approximately eight centimeters long. (Fig. 1) Thanks to the rubber sole, it sticks well on the table, and thus, there is no need to be concerned that the detector's own cable grabs it from its position.



Fig. 1 The 3-inch Leap Motion Controller with a glossy black face complimented by a smooth silver bezel

C. Communication from Leap Motion to Computer

The device connects to the computer via a USB 3.0 port, which provides its power supply and data exchange. The control unit's minimum system requirements contain AMD Phenom II or Nehalem architecture Intel processor and 2 gigabytes of RAM.

D. Operation Principle

The system applies three infrared LEDs and two monochrome infrared cameras to detect infrared light reflected from the hands, at 300 fps speed, they cover an approximately 1 meter diameter, almost semi-sphere shaped area, and the device can record not only the motion of our hand, but also the motion of each finger separately, so a computer can be controlled by smooth motoric motion.

The Leap Motion examines the environment on a much smaller area, but at a much higher resolution, thus enabling the separate tracking of fingers. If we perform hand and finger motions on the covered area, the system will recognize that as some kind of spatial position data, and execute the orders rendered to the gesture. The detector forwards the taken images to the computer, where the Leap Motion software processes the images of the two cameras by mathematical algorithms, so from the 2D sources, it can calculate 3D values regarding image position, this way managing the control. The program related to the device can be downloaded from the manufacturer's website, and following installation, Leap Motion can be used within minutes. Following the first switch on, the program guides through the basics of control, holding our hand above the detector, we can see its wireframe on the display, or we may draw by moving our fingers in the air (Fig. 2).



Fig. 2 Motion-controlled

E. Festo Robotino

Festo Robotino is a mobile robot system manufactured by Festo, which plays a major role in several researches. In Fig. 3, the Robotino itself is shown.



Fig. 3 Festo Robotino mobile robot

The completely functional mobile robot's central control unit is an embedded PC and a CompactFlash card, on which the Linux operation system specialized for the robot and the controller programs can be found. The control unit can be reached via Wireless LAN. The basics of controlling to all directions is enabled by the so-called omnidrive, which consists of three separate omni wheels placed into 120° angles, driven by a DC motor controlled by the omnidrive unit. Its advantage is that the robot is able to move from a specific starting point on any tracks, even rotating around itself. The omnidrive has got three inputs:

1) Vx [mm/s] (float) velocity;
2) Vy [mm/s] (float) velocity;
3) angle rotation velocity – omega [deg/s] (float), and three outputs, that define the velocity of each motor.

The block controlling the velocity of the motors controls the acceleration and brake of each motor using PID control.

The Robotino is equipped with several sensors, such as rubberized collision detector, infrared distance detector, encoder, VGA resolution camera, but it can be upgraded with any kind of sensors via the digital and analog I/O interface, or any other peripheries. The Robotino can be programmed in C++-ban, JAVA, .Net, MATLAB, but can be controlled in LabVIEW, or Robotino View programs. If properly programmed, it can autonomously perform the tasks.

F. Robotino View

The Robotino View (Fig. 5) is an interactive 'native' graphic programming environment, which is provided by Festo for the robot; it has got a user-friendly GUI and can be easily learnt through pre-defined examples.

The program can be downloaded for free from Festo's Didactic website. In the function block development environment control, image processing, logic, vector-algebra etc. objects are available, enabling complex control. In the Robotino View program, both blocks supporting image processing, oscilloscope and objects required for monitoring and controlling the change of detected signals in time are available. Instruction blocks may be executed sequentially (after each other) or iterated (repeatedly), and the execution of

the instruction depending on selection condition is also possible.

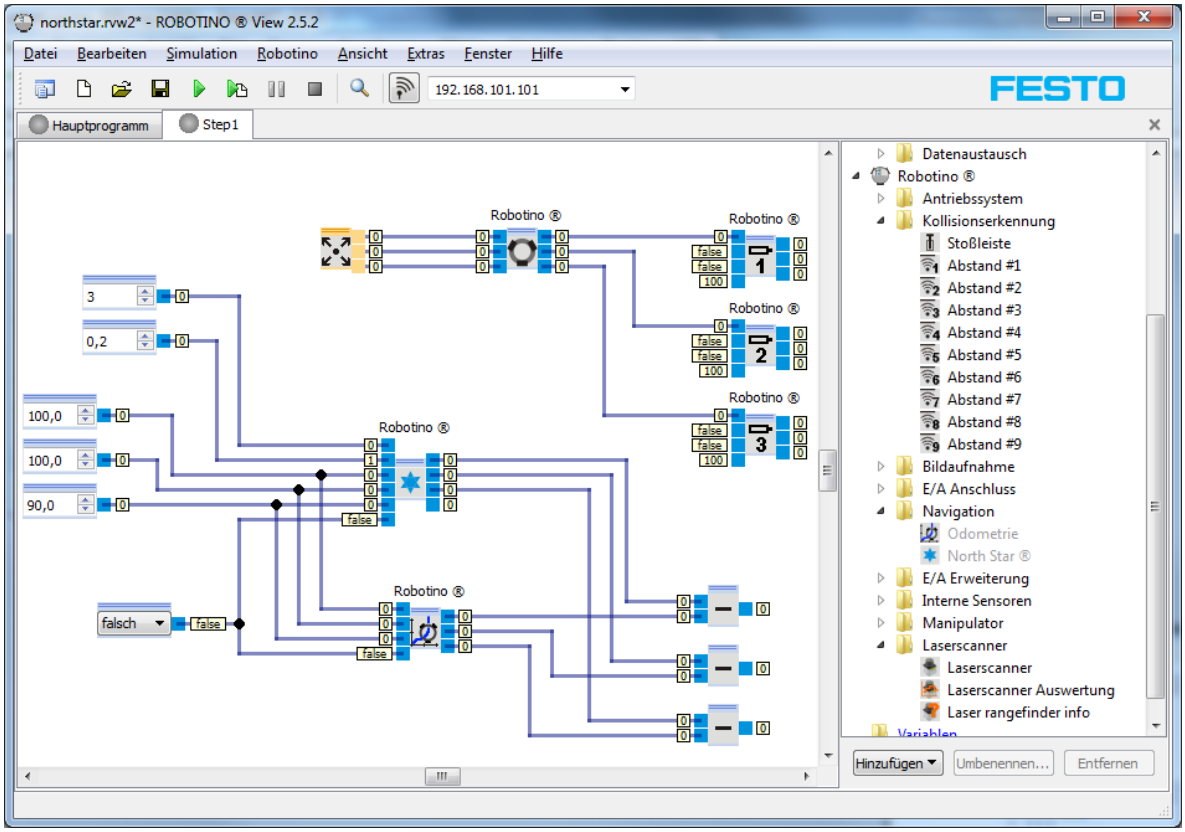


Fig. 4 Robotino View

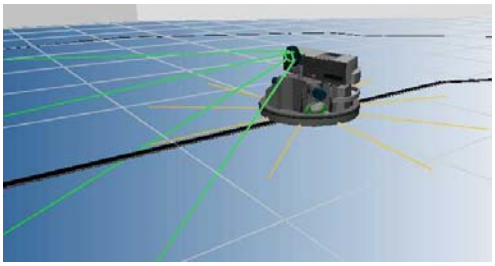


Fig. 5 Robotino Sim Demo

The program has two view options; the first one is the so-called "Subprogram view". Here we have to work with such components that control each task of the robot (e.g. motion, reception of built-in detectors, camera image processing etc.). The second view is the so-called "Main program view", where we can provide the quit conditions and execution sequences of different sub-programs. The development environment enables creating a data connection from external program via TCP/IP and UDP connection.

G.Robotino Sim Demo

The Robotino® Sim (Fig. 5) is a robot simulation software that realistically models the robot's motion in virtual 3D environment, which can be very useful during several developments. The Robotino simulation model contains everything as the real robot. In order to properly use the software, we need a video card with at least 128 MB RAM, supporting OpenGL.

III. Implementation the Mobile Robot Control Software

Among the goals described in the introduction, development of such an application can be found that enables the navigation of the Festo Robotino mobile robot by Leap Motion based hand gesture control. The implementation of data processing application has been made by C# high-level program language, while the application enabling the motion of the mobile robot in virtual environment according to evaluated data has been implemented in the Robotino View development environment.

The Leap Motion SDK specifies how and under what conditions they have received the data coming from the Leap Motion control unit. Using the official SDK and the driver of Leap Motion we were able to specify how process data provided by device control unit. The device sends hand action information periodically while it is turned on. Packages information is called a frame. With sensor information ID for instance Frame::hand() and Frame:finger() functions can be used or override if necessary. UDP server was used to forward the data to Robotino View for further processing.

Open Science Index, Mechanical and Mechatronics Engineering Vol:11, No:8, 2017 publications.waset.org/10007702/pdf

Open Science Index, Mechanical and Mechatronics Engineering Vol:11, No:8, 2017 publications.waset.org/10007702/pdf

A. Design

Following requirement definition, we created the software structures, data structures and algorithm descriptions to be implemented. The planning was not performed sequentially, so step-by-step, but in an iterative way, several versions were developed, and we were making continuous backsteps to correct former plans. To design the logic plan, we chose an easy-to-read and follow flow chart, while the UML (Unified Modelling Language) supporting object-oriented paradigm was applied in planning static and dynamic components in time. As the end-result of the planning process, such algorithms and data structures to be implemented became available - whose application enabled the effective production of application code.

B. The Application's Logic Plan at a Glance

As the first step of the planning, the system's logic plan was made using flow chart, as shown in Fig. 6.

C. The Application's Static UML Class Chart (Detail)

To create the static model, we applied the most commonly used chart type in UML, the class chart, where we showed the system components static items, and thus, we could easily examine static items during running the application. The simplified class chart of the application can be read in Fig. 7.

D. Implementation

The application's implementation performed during implementation and integration according to formerly created plans, in C# program language, in the Microsoft Visual Studio software development environment. The application runs in the background, for its current functionality, no control panel is necessary.

Integration is one of the important phases of development, when independently implemented and tested program parts are assembled into a complete system.

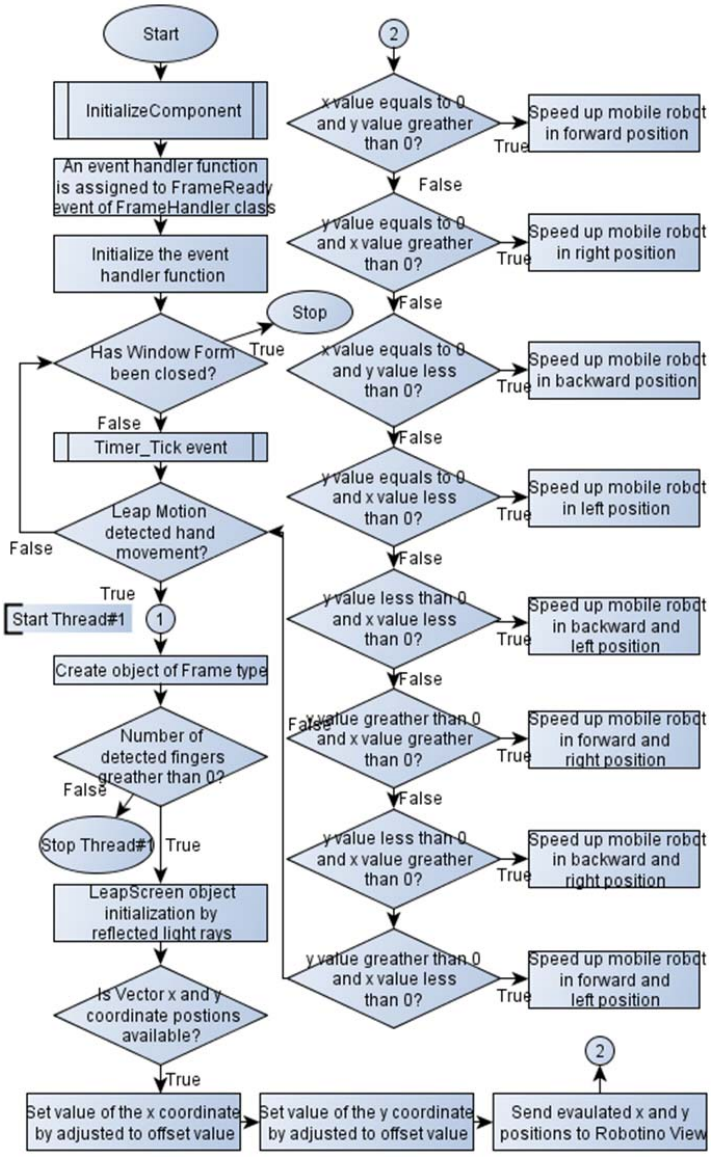


Fig. 6 The system's logic plan at glance

Open Science Index, Mechanical and Mechatronics Engineering Vol:11, No:8, 2017 publications.waset.org/10007702/pdf

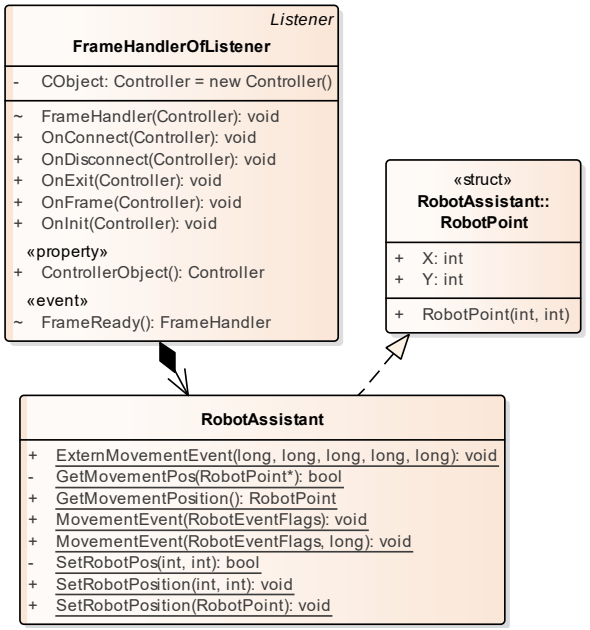


Fig. 7 The application's static UML class chart (part)

The now implemented application is built up from seven classes that are placed in one project. In the project, the addition of the so-called LeapCSharp.NET.3.5 DLL file as a reference was inevitable to manage data provided by the Leap Motion hardware. Furthermore, the DLL file embedded into the code base as a name space contains such class and function libraries already written by the manufacturer of the Leap Motion device that strongly support development. Following making the code base, the test environment was being tested.

IV. Results

The students were asked to fill in a questionnaire about the system. An eight-question survey was administered to students to elicit their opinion of the system, using a five-point Likert-type scale of "A": Strongly Agree; "B": Agree; "C": Unsure; "D": Disagree; and "E": Strongly Disagree).

In the examination of the implemented test environment, 14 test subjects participated, who were university students between the age of 18 years and 25 years.

Fig. 8 gives the mean of the results for each question, as asked of the university students. The students were asked about their experiences regarding the implemented system. Based on the answers given for the questions, it can be stated that the opinion of students about the implemented system and the gesture handle technology was very positive. In their opinion, the system was interesting, it helped them to use and learn the basics of Leap Motion to navigate a mobile robot. The system topic was interesting to 85% of the asked students and 100% strongly recommended that the grasping gesture should be adapted to different applications.

In the beginning of the testing, we asked the students to rate the system's usability on a scale ranging from 1 to 5, then repeat that after practicing for about 10-15 minutes. The result is shown on Table I.

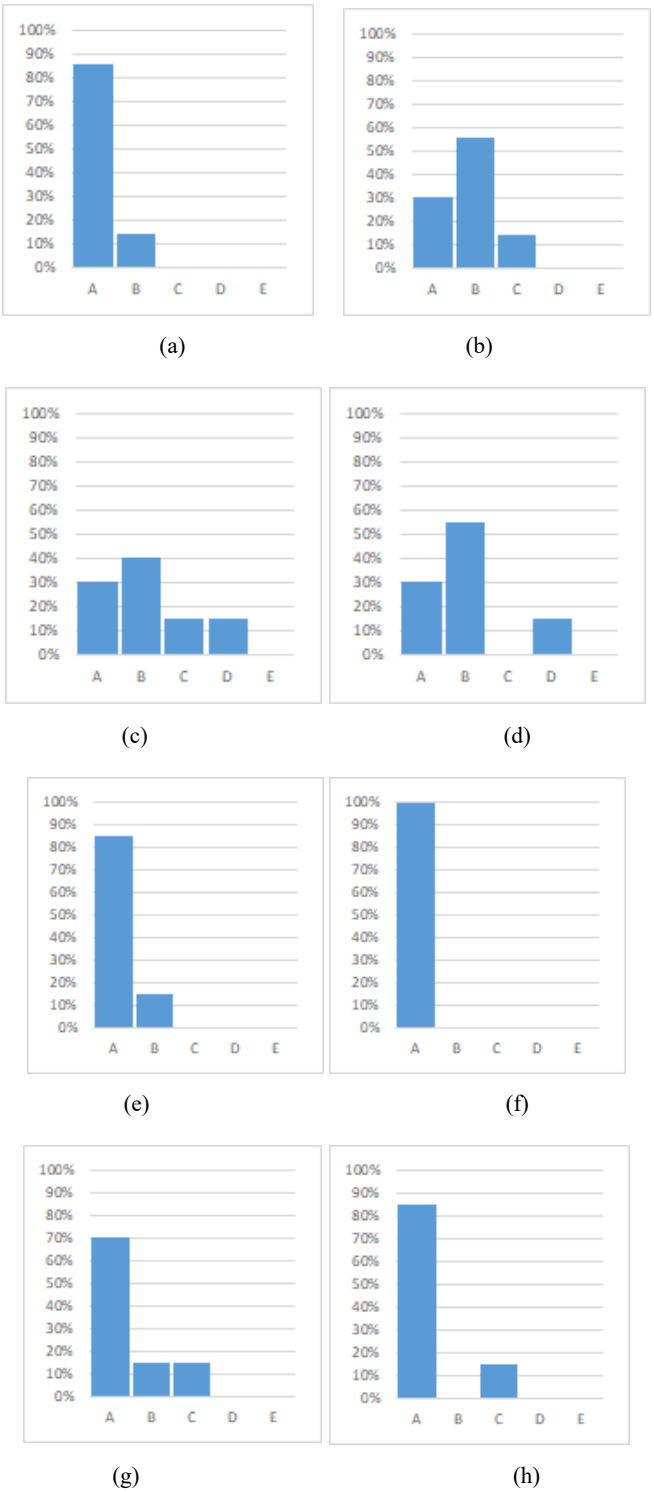


Fig. 8 Responses to the system questionnaire. (a) Q1: Did you find the implemented system interesting? (b) Q2: Was it difficult to learn the usage of Leap Motion? (c) Q3: Was mobile robot navigation system practical and efficient by using Leap Motion? (d) Q4: Is it possible to use Leap Motion instead of Kinetic to play games? (e) Q5: Can gesture handle adaptable in the daily life area? (f) Q6: Do you suggest that the grasping gesture should be adapted to different applications? (g) Q7: Were you satisfied with the system? (h) Q8: Do you think the gesture based control system can be more popular and attractive than traditional ways in the future?

TABLE I
MOBIL ROBOT MOVEMENT USABILITY TEST

Evaluation at the beginning		Evaluation after practice	
Test subjects	Usability[1]	Test subjects	Usability[1]
1.	4	1.	5
2.	3	2.	4
3.	3	3.	4
4.	2	4.	3
5.	5	5.	5
6.	3	6.	4
7.	4	7.	4
Average	**3.50**[2]	**Average**	**4.14**[2]

[1]The maximum mark was 5. 1 = unusable, 2 = hardly usable, 3 = usable, 4 = well-usable, 5 = excellent usable
[2]Average of the complete sample

We can get the conclusion from the results that according to the seven questioned test subjects, the test environment is usable, with a bit better result than average. There were such test subjects, who had difficulties in moving the mobile robot in the beginning, but others could easily move the mobile robot right at the beginning. Following 10-15 minutes of practice, according to the opinion of the test subjects, the system's usability improved by 0.72 points, so the 'usable' rating changed to 'well-usable'. Of course, the use of the system was not perfect though, the systems needs further fine tuning.

V. CONCLUSION

The application of a hand gesture controlled HCI was introduced in the article regarding the control of a mobile robot. The control of the mobile robot was implemented by the Leap Motion hand position detection system. The mobile robot's control was tested in realistic virtual environment. The implemented manual gesture control, as the usability of HCI based control, was examined by test subjects. According to the feedback given by the test subjects, the system can be handled well following a little practice. After performing the examination, according to the feedback given by the test subjects, the system's application also on other application fields would be useful. As a summary, by the development of IT systems, human-computer interaction is being developed to an even greater extent, and in the future, new communication methods will be available in human-computer interaction.

REFERENCES

[1] Y. Lin and W. Zhang, "A function-behavior-state approach to designing human-machine interface for nuclear power plant operators", IEEE Transactions on Nuclear Science, vol. 52, no. 1, pp. 430-439, 2005.
[2] Seong Soo Choi, Jin Kyun Park, Jin Hyuk Hong, Han Gon Kim, Soon Heung Chang and Ki Sig Kang, "Development strategies of an intelligent human-machine interface for next generation nuclear power plants", IEEE Transactions on Nuclear Science, vol. 43, no. 3, pp. 2096-2114, 1996.
[3] P. Lagari, A. Nasiakou and M. Alamaniotis, "Evaluation of Human Machine Interface (HMI) on a Digital and Analog Control Room in Nuclear Power Plants Using a Fuzzy Logic Approach", International Journal of Monitoring and Surveillance Technologies Research, vol. 4, no. 2, pp. 50-68, 2016.
[4] I. Rezazadeh, M. Firoozabadi, H. Hu and S. Golpayegani, "Co-Adaptive and Affective Human-Machine Interface for Improving Training Performances of Virtual Myoelectric Forearm Prosthesis", IEEE Transactions on Affective Computing, vol. 3, no. 3, pp. 285-297, 2012.
[5] M. Bosnak and I. Skrjanc, "Embedded Control System for Smart Walking Assistance Device", IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 25, no. 3, pp. 205-214, 2017.
[6] C. Cipriani, J. Segil, J. Birdwell and R. Weir, "Dexterous Control of a Prosthetic Hand Using Fine-Wire Intramuscular Electrodes in Targeted Extrinsic Muscles", IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 22, no. 4, pp. 828-836, 2014.
[7] B. Ravani, M. Gabibulayev and T. Lasky, "Improvement of a Human-Machine Interface (HMI) for Driver Assistance Using an Event-Driven Prompting Display", IEEE Transactions on Control Systems Technology, vol. 19, no. 3, pp. 622-627, 2011.
[8] P. Alvarado Mendoza, A. Lindgren and A. Angelelli, "Ecological interface design inspired human machine interface for advanced driver assistance systems", IET Intelligent Transport Systems, vol. 5, no. 1, pp. 53-59, 2011.
[9] C. Pickering, "The search for a safer driver interface: a review of gesture recognition human machine interface", Computing and Control Engineering, vol. 16, no. 1, pp. 34-40, 2005.
[10] S. Park and T. Sheridan, "Enhanced Human–Machine Interface in Braking", IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, vol. 34, no. 5, pp. 615-629, 2004.
[11] D. Santana, C. da Silva Santos and H. Hernandez Figueroa, "Human-Computer Interface Techniques to Design and Evaluate an Electromagnetic Simulator", IEEE Latin America Transactions, vol. 12, no. 4, pp. 725-732, 2014.
[12] H. Saini and R. Daruwala, "Human Machine Interface in Internet Of Things system", 2016 International Conference on Computing Communication Control and automation (ICCUBEA), 2016.
[13] M. Gertz, D. Stewart and P. Khosla, "A human machine interface for distributed virtual laboratories", IEEE Robotics & Automation Magazine, vol. 1, no. 4, pp. 5-13, 1994.
[14] Jiang, Y. C., "Menacing motion-sensing technology, different leap motion." PC. Fan 11, pp. 32-33, 2013.
[15] Gy. Molnár and Z. Szűts, Visual Learning - Picture and Memory in Virtual Worlds, In: András Benedek, Kristóf Nyíri (ed.) Beyond Words: Pictures, Parables, Paradoxes. 260 p. Frankfurt: Peter Lang Verlag, 2015. pp. 153-161. (ISBN:978-3-631-66385-1)
[16] Gy. Molnár, Challenges and Opportunities in Virtual and Electronic Learning Environments, In: Szakál Anikó (ed.) SISY 2013: IEEE 11th International Symposium on Intelligent Systems and Informatics: proceedings. Budapest: IEEE Hungary Section, 2013. pp. 397-401. (ISBN:978-1-4799-0303-0)
[17] I. Horváth, Innovative engineering education in the cooperative VR environment, 7th IEEE Conference on Cognitive Infocommunications CogInfoCom 2016, 16-18 October 2016, Wroclaw, Poland, pp. 000359 - 000364, doi: 10.1109/CogInfoCom.2016.7804576
[18] I. Horváth, The education of disruptive technologies in the innovative engineering training, Australian Journal of Intelligent Information Processing System, Vol.14, No.4. 2016.
[19] Dr Molnár Gy., The role of electronic and virtual learning support systems in the learning process, In: Szakál Anikó (ed.) IEEE 8th International Symposium on Applied Computational Intelligence and Informatics: SACI 2013. (IEEE) New York: IEEE, 2013. pp. 51-54. (ISBN:978-1-4673-6397-6)

Open Science Index, Mechanical and Mechatronics Engineering Vol:11, No:8, 2017 publications.waset.org/10007702/pdf