VOL. 9, NO. 12, DECEMBER 2014 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

OPEN PROTOCOL FRAMEWORK FOR TELEPRESENCE ROBOT

Petrus Santoso and Handry Khoswanto
Department of Electrical Engineering, Petra Christian University Siwalankerto, Surabaya, Indonesia
E-Mail: petrus@petra.ac.id

ABSTRACT

Research on telepresence robot has a lot of unexplored potential. Many researches have been done to pursue development on telepresence robot. Most of the researchers usually buy a commercial telepresence robot or build a special purpose one. Usually the needed protocol is developed from scratch. The developed system seldom has interoperability between each other and the process can be costly or lengthy. To expedite research process on telepresence robot development, this paper tries to present an open protocol framework for telepresence robot. The framework will consist of architecture definition, service specification and protocol specification. An implementation of open protocol framework has been simulated. A working hardware prototype that implemented the framework also has been built and tested. The result shows that the open protocol framework is useful and easy to implement. All the implemented protocol functions working properly. The function call of the SPs can be configured according to the needed parameters.

Keywords: telepresence, robotic, protocol.

INTRODUCTION

Telepresence refer to the ability to be present or participate in a distance location without actually being there physically. Telepresence involves many technologies such as robotic, computer network and signal processing. The role of robotic is to represent the physical entity of a telepresence system. Computer network deals with the underlying communication technology. Signal processing enables the sophisticated presentation of the communicating entities.

A lot of research has been done in the field of telepresence. Problems that have been addressed including cost optimization [1], usage of network bandwidth[2], ergonomic of the interface[3], use case of the system[4], [5], enhancement of presence[6], [7].

To do their research, the researchers usually buy a commercially available telepresence robot or build a special one. Most of the research usually isolated from other research in term of interoperability. To ensure interoperability there is a need of open protocol specification. Beside interoperability, open protocol framework will provide advantage on the acceleration of research development. A good framework will surely help researchers on doing their work.

A similar work on developing a framework of telepresence robot has been done by Peter Corke *et al*. [8] Their work based on a well known tools of ROS and Skype. The developed robot allows the remote user to interact with people near the robot and also to view maps, robot sensory data, robot pose and to issue high-level motion commands. Based on this experience, this paper want to define a generalized open protocol framework to overcome interoperability issue.

The objective of this paper is to propose an open protocol framework for telepresence robot research. To pursue the objective, an exploration has been performed. The exploration steps consist of a design process of a generalized framework of telepresence robot, a simulation of depicted design and a protoype implementation of the robot. The proposed framework will consist of architecture definition, service specification and protocol specification.

Design of the Open Protocol Framework

The design steps of the open protocol framework are outlined as follow. The first step is the requirement analysis process. The results of requirement analysis are used to define the architecture of the system. Architecture design will be followed by the service design. Service design presents all of the Service Elements needed by the system. The last step is outlined the whole system protocol. The protocol will implement all of the Service Primitives on each Service Elements.

On the requirement analysis, the following results are obtained. The telepresence robot should have the capabilities as follow:

- Can be controlled remotely
- Can move on constant speed at a predefined setting
- Having capability of video monitoring or surveillance
- Having capability of video call communication/conference
- Having obstacle detection feature or other autonomous features optionally

To fulfill those requirements, a certain components of the system need to be presented. Control capabilities will be provided by a certain controller and a communication module. To move on constant speed, the system will have motor and motor driver and power supply. As of video monitoring, surveillance and communication; the functions will be provided by a device with a camera. Obstacle detection feature or other autonomous features will need certain sensors as required. Based on the requirement analysis, the basic architecture of the system is obtained. Figure-1 depict the architecture of the system.

VOL. 9, NO. 12, DECEMBER 2014 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

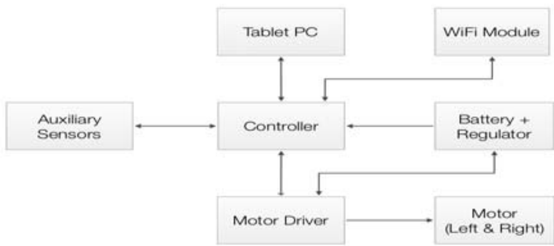

Figure-1. System Architecture.

The main system component will be the controller and the tablet. Controller will deal with the movement of the robot, including autonomous features (if present). The tablet will handle communication process during a conference or monitoring session using the WiFi module. Besides the two main components, there are other components such as battery, motor driver, sensor, etc (according to the requirements).

The service design is depicted on Table-1. It shows the list of acquired Service Elements based on requirement analysis. The Service Primitives (SP) is grouped into 4 Service Elements, namely: Robot Movement, Monitoring, Video Call and Auxiliary Service.

Table-1. Service Primitives.

Service Element	Service Primitive
Movement	Forward_Req
	Backward_Req
	Left_Req
	Right_Req
	Stop_Req
	Move_Ack
Monitoring	Video_Req
	Video_Ack
Video Call	Call_Req
	Disc_Req
	Add_Recipient_Req
	Remove_Recipient_Req
Auxiliary	Send_Robot_Status
	Send_Aux_Req
	Get_Aux_Req
	Aux_Ack

Movement Service Element deals with the movement of the robot. Monitoring Service Element handles video monitoring capability. Video Call Service Element handles the video call communication process. Auxiliary Service Element is prepared to accommodate other features such as obstacle detection or other autonomous features of a telepresence robot.

Based on the depicted service design, the protocol can be constructed. To give the outline of the protocol, the simplified version is depicted on Figure-2 (State Diagram of the Protocol).

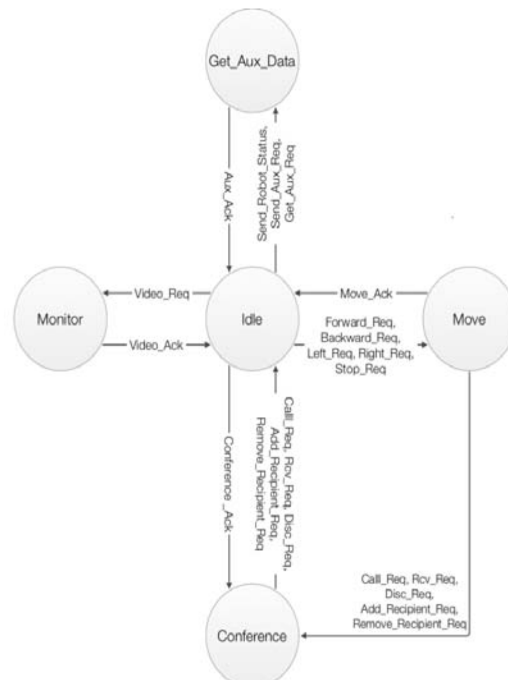

Figure-2. State Diagram of the Protocol.

Experiment

To ensure the usage scenario feasibility, some experiments have been done.

The first experiment is by implementing the protocol using VPL of the Microsoft Robotics Developer Studio (MRDS). The robot implemented in the simulation is a two-wheel differential drive robot based on the MARK Reference Platform [9]. MARK Reference Platform is a platform design specified by Microsoft. The specification is intended to provide guidance on developing robot that can be simulated using MRDS. A screen capture of the running simulation is depicted on Figure-3.

ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

Figure-3. Simulation.

There is a limitation on the simulation process. The Monitoring Service Element and Video Call Service Element cannot be simulated because the limitation on the available hardware.

Robot Movement Service Element can be simulated properly. The movement of the robot is controlled via a joystick hardware attached to the computer. Protocol command can be interpreted properly, and the robot can move accordingly in the simulated environment.

Auxiliary Service also simulated using a Simulated Kinect component in the Microsoft Robotics Developer Studio. Robot can receive the information from Simulated Kinect and perceived the depth data.

The second experiment is by implementing a two-wheel differential drive telepresence robot. A mechanic platform based on the simulation has been made. The platform is made utilizing acrylic material and an aluminium pole for tablet stand. The base of the platform has 40 cm diameter. The height of the aluminium pole is 150 cm. The robot also makes use of acrylic material for its wheel. The diameter of the wheel is 15 cm.

The electronic controller uses an Arduino Uno as a controller board. The tablet uses in the prototype is a Galaxy Table-3 Lite tablet. Motor that drives the robot is a pair of GM-42 DC Motor with EMS 30A H-Bridge Driver. Auxiliary Service is not implemented yet.

The prototype implementation is depicted on Figure-4.

Figure-4. Prototype Implementation.

A pseudocode example of implemented protocol functions is shown on Figure-5. The examples represent the pseudocode for Movement Service Element implemented in the Arduino Uno as a controller board.

```
Switch (Cmd):
   Forward_Req:
       PWM_Value = Param[0]
       digitalWrite(M1,High)
       digitalWrite(M2,High)
       analogWrite(E1,PWM_Value)
       analogWrite(E1,PWM_Value)
       Move_Ack(Forward)
   Backward_Req:
       PWM_Value = Param[0]
       digitalWrite(M1,Low)
       digitalWrite(M2,Low)
       analogWrite(E1,PWM_Value)
       analogWrite(E1,PWM_Value)
       Move_Ack(Backward)
   Left_Req:
       PWM_Value = Param[0]
       digitalWrite(M1,High)
       digitalWrite(M2,Low)
       analogWrite(E1,PWM_Value)
       analogWrite(E1,PWM_Value)
       Move_Ack(Left)
   Right_Req:
       PWM_Value = Param[0]
       digitalWrite(M1,Low)
       digitalWrite(M2,High)
       analogWrite(E1,PWM_Value)
       analogWrite(E1,PWM_Value)
       Move_Ack(Right)
   Stop_Req:
       PWM_Value = 0
       digitalWrite(M1,Low)
       digitalWrite(M2,Low)
       analogWrite(E1,PWM_Value)
       analogWrite(E1,PWM_Value)
       Move_Ack(Stop)
```

Figure-5. Pseudocode Example.

As of the result of the simulated version of the robot, the prototype also gives similar result. All the implemented protocol functions can be interpreted properly. Movement of the robot, video monitoring and video call feature are work well in the real environment.

The prototype also shows that the robot can move with constant speed (as stated in the requirements). The speed measurement result is depicted on Figure-6.

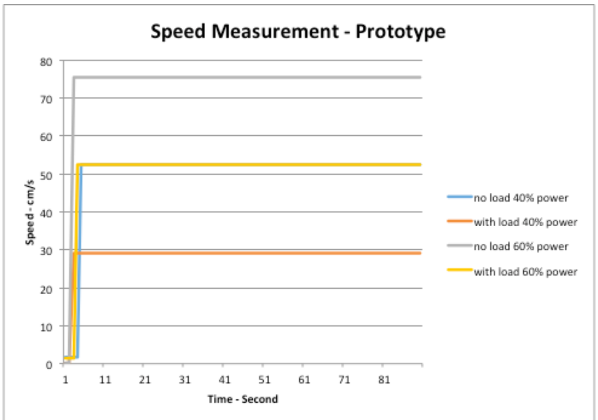

Figure-6. Speed Measurement Result.

The result of speed measurement shows that after a certain startup time, the robot can move with a constant speed. On 40% power with loaded condition, the robot can move at 29.12 cm/s. On 60% power with loaded condition the robot can move at 52.3 cm/s.

The result of experiment either in the simulation or prototype implementation gives a satisfying confirmation that the implemented protocol function properly.

DISCUSSIONS

To be accepted as a standard, the frameworks still need further testing and polishing. Improvement on the robot feature is open for discussion. A complete specification of the service and protocol should be discussed. There is also a need of protocol validation to ensure the robustness of the design.

After complete protocol specification is achieved, a library of functions can be implemented to simplify development on the telepresence robot platform.

CONCLUSIONS

The depicted robot in this paper is a work in progress. Based on the experiment result it can be shown that the open protocol framework can be implemented easily. The function call of the SPs can be configured according to the needed parameters.

Further works still needed to complete the Open Protocol Framework specification. Comment and suggestion very much expected. In the end, we hope that the proposed framework can be developed further and in the end can be accepted as a standard to enable faster development on telepresence robot.

ACKNOWLEDGEMENT

This Research is funded by Competitive Research Grants fund of Indonesian Higher Education Directorate under contract no. 32/SP2H/PDSTRL_PEN/LPPM-UKP/IV/2014.

REFERENCE

[1] D. A. Lazewatsky and W. D. Smart. An Inexpensive Robot Platform for Teleoperation and Experimentation.

[2] K. Bijlani, P. Venkat Rangan, S. Subramanian, V. Vijayan, and K. R. Jayahari. 2010. A-VIEW: Adaptive bandwidth for telepresence and large user sets in live distance education. In Education Technology and Computer (ICETC), 2010 2nd International Conference on. 2: V2-219.

[3] B. Keyes. 2007. Evolution of a Telepresence Robot Interface. University of Massachusetts Lowell.

[4] K. M. Tsui, M. Desai and H. A. Yanco. 2010. Exploring Use Cases for Telepresence Robots. no. August. pp. 11-18.

[5] K. M. Tsui, A. Norton, D. Brooks, H. A. Yanco, and D. Kontak. Designing Telepresence Robot Systems for Use by People with Special Needs.

[6] H. Nakanishi, Y. Murakami, D. Nogami and H. Ishiguro. 2008. Minimum Movement Matters : Impact of Robot-Mounted Cameras on Social Telepresence. pp. 303-312.

[7] H. Nakanishi, Y. Murakami, and K. Kato. 2009. Movable Cameras Enhance Social Telepresence in Media Spaces. pp. 433-442.

[8] P. Corke, K. Findlater, and E. Murphy. 2012. Skype : a communications framework for robotics. In Proceedings of Australasian Conference on Robotics and Automation.

[9] Microsoft Robotics Group. 2012. Reference Platform Design. V1.0. [Online]. Available: http://go.microsoft.com/fwlink/?LinkID=228540&clcid=0x409.