Global Journal of Computer Science and Technology: D
Neural & Artificial Intelligence
Volume 21 Issue 1 Version 1.0 Year 2021
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Development of a Portable IP-Based Remote Controlled System for Mobile Robot

By Daramola O. A., Obe O. O. & Oriolowo A.
Federal University of Technology Akure

Abstract- The use of Mobile Robots to interact with objects in remote locations has proved to be useful in areas not easily accessible or too dangerous for humans. Various means have been used to remotely operate or control Mobile Robots. These range from wired connection to Wireless connection like radio frequency signal and more recently internet controlled Mobile Robot using the TCP/IP protocol stack. However, the problem of remote control dependence on the Mobile Robot Platform or configuration has made it difficult to switch controllers between Mobile Robots. In this work, a portable IPbased remote control system has been designed and implemented to remove the constraint imposed by the Mobile Robot's platform in choosing the control interface. The system developed was built on three loosely coupled components working together to ensure a high degree of Control interface portability. The Mobile Robot Gateway component was used to receive and send data from the Mobile Robot.

Keywords: mobile robot, remote control, command hub, rest architecture, TCP/IP, IP-based.

GJCST-D Classification: I.2.9

DEVELOPMENTOFAPORTABLEIPBASEDREMOTECONTROLLEDSYSTEMFORMOBILEROBOT

Development of a Portable IP-Based Remote Controlled System for Mobile Robot

Daramola O. A. [α], Obe O. O. [σ] & Oriolowo A. [ρ]

Abstract- The use of Mobile Robots to interact with objects in remote locations has proved to be useful in areas not easily accessible or too dangerous for humans. Various means have been used to remotely operate or control Mobile Robots. These range from wired connection to Wireless connection like radio frequency signal and more recently internet controlled Mobile Robot using the TCP/IP protocol stack. However, the problem of remote control dependence on the Mobile Robot Platform or configuration has made it difficult to switch controllers between Mobile Robots. In this work, a portable IP-based remote control system has been designed and implemented to remove the constraint imposed by the Mobile Robot's platform in choosing the control interface. The system developed was built on three loosely coupled components working together to ensure a high degree of Control interface portability. The Mobile Robot Gateway component was used to receive and send data from the Mobile Robot. The Command Interface component enables the user to issue commands to control the Mobile Robot. The command hub component is a REST-based service over HTTP saddle with the role of relaying commands between the control interface and the mobile robot gateway. C, C#, Python, and JavaScript were used at different levels to accomplish different tasks during the implementation phase. Apache Cordova and Ionic framework were used to develop a cross-platform mobile application for the control interface while MS SQL Server 2012 was used as the backend storage. The time complexity of the entire system was evaluated and has a value of O(n) which means the system executes in linear time.

Keywords: mobile robot, remote control, command hub, rest architecture, TCP/IP, IP-based.

I. Introduction

Mobile robots were defined by Posadas et al (2008) as physical agents that move and interact continuously while embedded in a dynamic environment. A mobile robot is also described as a situated and embodied agent endowed with mobility (Obe and Dumitrache, 2012). Control of mobile robots is categorized into three types namely, autonomous control, semi-autonomous control, and Tele operation. Autonomous control implements various control algorithms that control mobile robots in their environment without human intervention. Semi-autonomous control allows a user to instruct the mobile robot on what to do but the robot decides how the task is carried out. This mode of control is supervisory. In a Tele operated control, the mobile robot is entirely controlled by the user. Oxford dictionary (2015) defines "remote control" as the control of a machine or apparatus from a distance utilizing radio or infrared signals transmitted from a device. Teleoperation is often used in place of Remote control in research or technical environments. Teleoperation means controlling or doing work at a distance (Wichmann et al, 2014). The meaning of the distance, however, can vary. The distance can be physical, such as an operator controlling a robot at a remote location. The distance could also be a change in scale, for example, a surgeon using teleoperation to conduct surgery at the microscopic level. In 2014, Wichmann et al. reported that teleoperation systems are designed using a master-slave system model where the control system depends heavily on the mobile robot platform.

Various communication technologies have been used to achieve teleoperation for a mobile robot. These technologies include Bluetooth, Infrared, WIFI, GSM, Internet of Things (IoT) (Chikurtev, 2019), and Internet (Luimula et al., 2007). While some of these technologies impose a constraint on the range of operation, GSM and the Internet have proven to overcome such barriers (Chin et al, 2003, Ankit, 2014 and Juang&Juang, 2016). The Internet which is based on TCP/IP enables connectivity of billions of devices worldwide, giving access to communication, data, pictures, videos, and even real images of distant environments (Siegwart and Saucy, 1999). The advancement in the development of internet applications has made it the ideal testing ground for sophisticated new applications, such as video-conferencing and remote control systems (Oboe & Fiorini, 1997). However, only a few examples of real physical interaction with distant places are available at the moment. Goodrich and Schultz (2007) defined two categories of interaction, remote and proximate. Remote interaction refers to the situation where the human and the robot are separated spatially or even temporally (i.e., Opportunity Mars rover), while with proximate interaction the humans and the robots are collocated.

This research work aims at creating a portable IP-based remote-controlled system for mobile robot interaction. According to James (1997), a software unit is portable (exhibits portability) across a class of environments to the degree that the cost to transport and adapt it to a new environment in the class is less than the cost of re-development. Portability in general

Author α σ ρ: Federal University of Technology Akure, Ondo State Nigeria. e-mails: ooobe@futa.edu.ng, oadaramola@futa.edu.ng, adeoriolowo@gmail.com

terms is the usability of an object or component of the system in multiple environments without any modification in the internal structure.

Internet Protocol (IP) is a protocol in the Internet Layer of Transport Control Protocol/Internet Protocol (TCP/IP) model. It hides the underlying physical network by creating a virtual network view (Lydia et al, 2006). It is an unreliable, best-effort, and connectionless packet delivery protocol. Best-effort means that the packets sent by IP might be lost, arrive out of order, or even be duplicated. IP was designed with the assumption that a higher layer of the protocol will address these anomalies. According to Lydia et al., (2006), one of the reasons for using a connectionless network protocol was to minimize the dependency on specific computing centers that used hierarchical connection-oriented networks. Available remote control systems require specialized hardware and software that are heavily dependent on the mobile robot's platform or architecture. In this work, an architecture is proposed to eliminate the dependency of the remote control system on the robot's platform.

There are issues with existing mobile robot remote control systems which include but are not limited to, operation range, portability, and the multi-interface option of the system. Various researchers have addressed some of these problems, however, the problem of switching a remote control system from one mobile robot to another with little or no reconfiguration has not been dealt with. In Chin et al. (2003), a Real-Time Remote Control Architecture Using Mobile Communication was proposed. GPS, GSM, and GIS were applied in the development of real-time communication for remote data transfer which solved the range of operation problems, but the system lacked portability. Jose et al. (2004) illustrated a Java/Matlab-Based Environment for Remote Control System Laboratories. Matlab/Simulink and the Quanser Win Con environment were used to develop a control system using the HTTP server to process client requests. The client program is a java applet written in Java. The range of operation issue was eliminated by the architecture but failed to address portability and multi-interface options. In Mobile Robot Temperature Sensing Application via Bluetooth, Abdullah and Poh (2011) developed a control system using the KC-21 Bluetooth module and PIC16F877A microcontroller for remote temperature sensing. The system has a limited range of operations. Ankit et al. (2014) in Controlling of Remote Robot through mobile phone using DTMF Signal, developed a remote control system using microcontroller, CDMA modem, and DTMF signal. The system lacks portability and multi-interface options. Almali et al. (2015) developed a Wireless Remote control for Mobile robots operating in dangerous or narrow places for human beings. A 433MHz RF transceiver module was used to establish a connection between the mobile robot and the computer controlling it. The system has a limited operating range and also lacked portability.

This research work addresses the issue of portability and multi-interface options for mobile robot remote control systems. This will facilitate the use of one remote control system with a multi-interface for different mobile robots on several platforms, this will give room for using a robust interface for the control system at a particular time.

II. Related Literature

Oboe and Fiorini, 1998 in "A design and control environment for internet-based telerobotics describe the environment for the design, simulation, and control of internet-based force-reflecting telerobotics using a segment of the network to connect the master to the slave. Simulation of the complete telerobotic system and emulation using a planar force-reflecting master and a virtual slave uses a MatLab-Simulink program interfaced with a set of dedicated routines for internet modeling. The issues in the variable time-delay system were addressed by using the delay parameters acquired from the network probe to design the controller. However, the work does not address multiple interfaces that could be used as the master or controller. Lung et al., 2002; designed an internet-based human-assisted robotic controller system but the security of the web page was not considered as anyone that stumbles on the web application can control the robot. Chin et al., 2003 adopted the use of G3 (global positioning system (GPS), global system for mobile (GSM), and geographic information system (GIS)) system in developing real-time communication and remote control systems for robot real-time remote control, navigation, and surveillance. The designed system is not portable as it was only implemented on a Windows platform. Visual Basic, which is the choice of programming language, is not supported on other operating systems. Jose et al., 2004; developed a Java/Matlab-Based Environment for Remote Control System Laboratories, illustrated with an Inverted Pendulum - a novel environment that provides 24-hours-a-day access to a Web-based lab for the remote control of different didactic setups. A detailed description of an environment for the teleoperation of real Lab via the Internet, was achieved however, the design cannot be replicated on robots with different platforms due to the heavy dependence on Matlab/Simulink.

Abdullah & Poh, 2011; Mobile Robot Temperature Sensing Application via Bluetooth developed a Bluetooth-based control system for remote measurement of temperature from the robot's surrounding environment. The range of operation is limited to ten (10) meters. Esteller-Curto et al, 2012; in the Proposal of a REST-based architecture server to control a robot proposed the use of a REST-based architecture server for the control of a robot. The use of proprietary protocol between the robot and the server limits the portability of the control architecture to other robot platforms. Pravin and Shalini, 2013 developed a remote control system based on the hand gesture of the user. The gesture is captured by an accelerometer sensor capable of tracking coordinate values of X, Y, and Z axes as the sensor are tilted. Radiofrequency 433Mz is used as a communication link. The remote control has a limited range of about 100 meters radius. A remote control system for mobile robots using DTMF tone from mobile phones was developed by Ankit et al., 2014. The system has no visual feedback to enable the user to monitor the movement and position of the robot within its environment. Adamides et al., 2014; worked on the development of a suitable user interface for teleoperated agricultural spray robots using multiple camera feedback, keyboard, and PS3 gamepad. The communication link is based on RF signals and has a limited range of operations. The interface design was restricted to the keyboard and Human Interface device (Gamepad) for robot control. Dutta and Zielinska, 2015 identified the challenges concerning IoT-aided robotic applications, with particular reference to their technological and scientific implications. The work is entirely theoretical. No implementation is discussed for the proposed architecture.Amali et al., 2015; in Wireless Remote control of a Mobile Robot designed a mobile robot serving dangerous and narrow areas for humans. The mobile robot consists of a mobile platform and a 4-DoF robot arm with a gripper. This work established the development of a mobile robot with teleoperation capability however, operation range is limited and integration with other platforms was not considered.

III. Method

a) System Overview

The proposed portable IP-based remote control system for Mobile Robots is aimed at solving two basic problems usually encountered with common Mobile Robot remote control systems. The first problem is the communication range between the Mobile Robot and its remote control interface. This range limit is entirely dependent on how far the communication link can transmit or receive data sent from both ends (usually 10 Meters for Bluetooth, 100 Meters for WIFI). The second problem is the portability of control interfaces with different platforms to provide different user interfaces that can be used to control different Mobile Robots. To solve the above-mentioned problems, the new system is divided into three functional parts namely: Command hub, Control Interface, and Mobile Robot Gateway.

b) System Architecture

Figure 1 shows the entire architecture of the Mobile Robot Remote Controlled System. The system components run separately on different devices, performing different roles in achieving the overall goal of the system. The components are also loosely coupled to ease upgrade and maintenance without affecting the operation of the other ones.

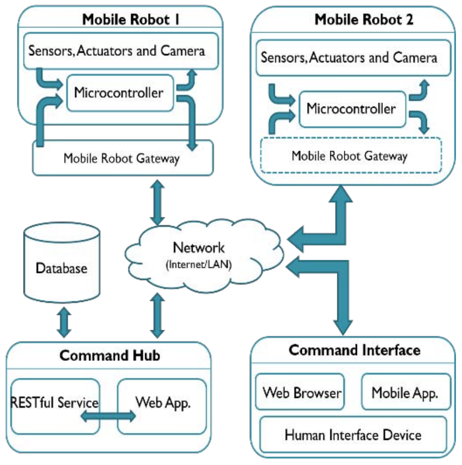


Figure 1: The System Architecture

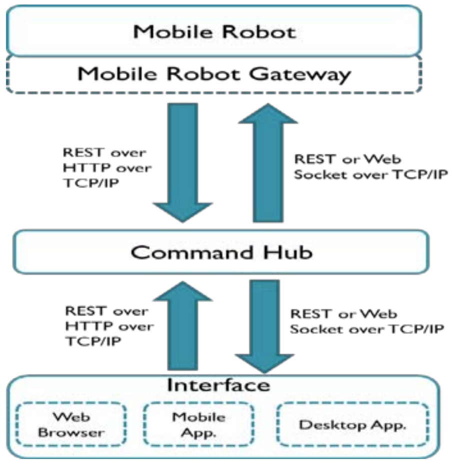

Mobile Robot
Mobile Robot Gateway
REST over HTTP over TCP/IP
REST or Web Socket over TCP/IP
Command Hub
REST over HTTP over TCP/IP
REST or Web Socket over TCP/IP
Interface
Web Browser
Mobile App.
Desktop App.

Figure 2: The System Communication Protocol

i. Command Hub Design

The Command Hub is at the center of the entire Remote control system for mobile robots. It is one of the three components required to develop the full system as shown in figure 2. This component acts as the major middleware between the Control Interface and the Mobile Robot Gateway. It is responsible for handling communication between the Gateway and the Control Interface. Digging deep into what made up the Command Hub, it is a REST-Based Service that runs a remote Webserver. It exposes a REST API and resources which can be accessed through Uniform Resource Identifier (URI) and corresponding HTTP Verbs (i.e. GET, POST, PUT, and DELETE). Every action performed by the user on the Control Interface is mapped by an HTTP request to the command Hub. The Command Hub then receives the request, processes it, and responds appropriately. From the request received to the response given, there are three different modules responsible for carrying out the tasks required to transform the request to a corresponding response as shown in figure 3. The modules are:

Input/output module: This module is the first and the last point of contact for both Control Interface and Mobile Robot Gateway. It is responsible for intercepting any request made to the Command Hub, determine the resources that are requested, and then pass control to the appropriate controller fit to handle such requests. The module is also responsible for taking results from any controller and generates the output response which is either a stream of byte when using Web Socket or JSON/XML data format whenever web Socket is not available.

Command Controllers: This module carries out various important tasks required to correctly and efficiently process all the requests received from both the control interface and the gateway. This module is made up of different controllers to handle different requests coming from the input/output module. Controllers available in the module include Command controller (responsible for generating commands based on the type of request received), Motor Controller (handles requests related to Robot's movement control), Proximity controller (governs the request related to proximity sensors), and vice versa. All the controllers present in the module interact with the data access layer to get, insert, update, or delete records from the database. This module is also responsible for constructing the appropriate response to be sent back to the client's requests. Responses are delivered in two ways which are determined by the module. The module makes use of Web Socket to push new or updated content to the client (Control Interface and Gateway), if the client supports Web Socket, a push notification is used to stream the response back to the client. If the client does not support Web Socket, the traditional HTTP response is used to have a payload of either JSON or XML depending on the content negotiation from the client.

Data Access Layer: This module interface directly with the database. It contains all the necessary stored procedures and SQL statements required to retrieve, update, insert, and delete any record from the database. This module is entirely controlled by the controllers.

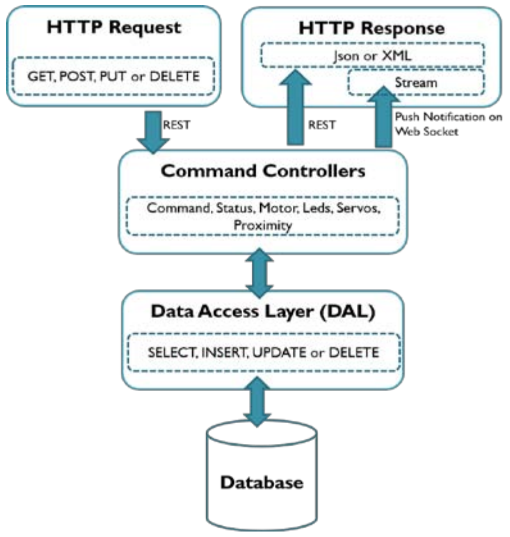


Figure 3: Command Hub Architecture

ii. *Control Interface Design*

The control interface component in the system is the one that interacts with the user and based on the operation performed on it, it will create and send an appropriate HTTP request to the command hub respectively. The control interface performs two basic functions. The first function is to take users' actions and transform them into a valid HTTP request required by the command hub. The second function is getting telemetry data from the command hub to create visual feedback for the users. The structure of this component is therefore organized along with its functions. Two modules are provided, one to constantly fetch new or updated data from the command hub to update the telemetry view of the mobile robot's vitals while the other module handles user interactions with the control interface and makes appropriate HTTP requests to the command hub.

Different Control Interfaces are designed based on the two modules explained earlier to expand the choice of control for each Mobile Robot in the system. Available control interface includes:

Cross-Platform Mobile Interface: This type of interface is meant for users that prefer to control their Mobile robots from a smartphone. The interface wasdesigned with Apache Cordova, HTML 5, CSS 3, and JavaScript. This ensures that the resulting interface can be used on Windows Phone, Android Phone, and iPhone as well.

Web Application Interface: This type of interface is available to users that want to control their Mobile Robots through a web browser. The web browser is used to access Web applications deployed alongside the command hub to provide the control interface. This Control interface is designed with ASP.Net MVC 4, HTML 5, CSS 3, and JavaScript. A variety of web browsers are supported.

Desktop Interface: In this type of control interface, the traditional Desktop application is used to provide the interface to the users. C# programming language is used to develop this interface.

iii. *Mobile Robot Gateway Design*

This is the component that interfaces directly with the Mobile Robot. It plays a major role in connecting the robot to the remote control interface through the command hub. There are two different types of Mobile Robot Gateway namely, Internal Mobile Robot Gateway and External Mobile Robot Gateway. The internal Mobile Robot Gateway runs locally on the robot's operating system. This type of Gateway is only supported by Mobile Robots capable of running scripts, executable programs and whose hardware supports networking and direct connectivity to the Internet. The Gateway is implemented as part of the required software running locally on the robot's platform. The Gateway runs directly on the Mobile Robot's control board as shown by Robot 2 in figure 1. External Mobile Robot Gateway does not run on the robot platform, it runs on a remote computer connected to the Mobile Robot through Bluetooth or a WIFI device. This type of Gateway is meant for Mobile Robots that do not have shields or devices to directly connect to the Internet thereby using the remote computer's Internet connectivity for its operation. In the case of External Mobile Robot Gateway, a remote computer connected to the Internet is placed within the communication range of the Mobile Robot, the computer shares its internet connection with the Mobile Robot. The Gateway is also executed on the remote computer to serve as the conduit through which communication is established with the control interface via the command hub. Robot 1 in figure 1 makes use of the External Mobile Robot Gateway.

Gateway Operation Circle

The Mobile Robot Gateway operates in a circle of three operations namely Sense/Listen, Think, and Act. Figure 4 shows the interaction of Gateway operations in a single circle.

- Sense/Listen Operation:- in this operation, the Gateway awaits new commands from the command hub through an active Web Socket if available or through an HTTP request to the REST API exposed by the command hub. If the Mobile Robot is in autonomous or semi-autonomous mode, values from the proximity sensors are fetched by the Gateway. Any data acquired from this operation is then passed on to the THINK operation in the circle.
- Think - analysis of data passed from the Sense/Listen operation is carried out during this operation. It is mainly responsible for command interpretation and conversion into a simpler form that could be handled on the Mobile Robot easily. After the interpretation and conversion task has been carried out, the result is sent to the ACT operation to take necessary action.
- Act: - this part is responsible for generating the control sequence based on the input received from the THINK operation. It determines the destination of the resulting control sequence (i.e. Mobile Robot or Command hub). The control sequence now determines the behavior the Mobile Robot will exhibit. This operation is also responsible for sending telemetry data to the command hub. After this operation is carried out, control is returned to the Sense/Listen operation

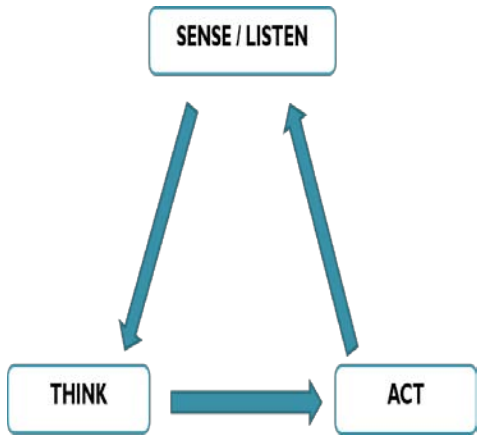


Figure 4: Mobile Robot Gateway Operation Circle

iv. *Command Design*

One of the roles of the Command Hub is the generation of commands to be sent to the Mobile Robot. These commands are generated based on the interaction of the users with the control Interface. Each action performed by the user on the control interface corresponds to a specific command meant for the Mobile Robot to execute. When an action is performed on the control interface, a request is sent to the REST API (Application Programming Interface) exposed by the Command Hub. The Command Hub now maps the request to the corresponding command to be generated and sent to the Mobile Robot. For example, if the user presses the "Move Forward" Button on the Control Interface, an HTTP POST request is sent to http://Command-Hub-DomainName/api/move/1, then "f" command is generated and sent to the Mobile Robot. The "Command-Hub-DomainName" is the domain name of the server hosting the Command Hub. Table 1 shows the comprehensive list of all Uri, HTTP verbs, and associated commands mapped to them.

Table 1: Command-Uri Mapping

URI	HTTP VERB	COMMAND	DESCRIPTION
/api/move	GET	W	Get Wheels State
/api/move/1	POST	F	Move forward
/api/move/2	POST	b	Move Backward
/api/move/3	POST	r	Move Right
/api/move/4	POST	l	Move Left
/api/move/5	POST	s	Stop
/api/speed	GET	v	Get Robot Speed
/api/speed/value	POST	v,value	Set Robot Speed to value
/api/leds	GET	d	Get all LED Status
/api/leds/id	GET	d,id	Get the Status of LED id
/api/leds/id/status	POST	d,id,status	Set the state of LED id with status
/api/servo	GET	c	Get Status of all Servo
/api/servo/id	GET	c,id	Get state of Servo id
/api/servo/id/pos	POST	c,id,pos	Set the position of Servo id to position
/api/motor	GET	m	Get the state of all Motor
/api/motor/id	GET	m,id	Get state of motor id
/api/motor/id/pwm	POST	m,id,PWM	Set the PWM signal of Motor id
/api/prox	GET	p	Get state of all proximity sensor
/api/prox/id	GET	p, id	Get state of proximity sensor id

v. *Mobile Robot Design and Configuration*

At the hardware level, two robots were designed to validate the workability of the remote control system whose architecture was shown in figure 1. These Mobile Robots have different platforms and configurations each of which corresponds to the two types of Mobile Robots in figure 1. The Mobile Robots are:

SleekBot V1: This was designed to use Arduino Nano R3 as its main control board. All other actuators and sensors are connected to the main control board. In this configuration, a Bluetooth module is used to communicate with an external Mobile Robot Gateway. Figure 6 shows the breadboard schematics of the robot's hardware configuration.

SleekBot V2: This is the second Mobile Robot, it was designed to use Raspberry Pi 2 Model B as its main control board while Arduino Nano R3 was used as a slave control board. In this configuration, a USB camera with a two degree of freedom is connected to the mainboard and Internal Mobile Robot Gateway was used because the main control board is capable of running executable programs and it is also capable of connecting to the internet directly. Figure 7 shows the breadboard schematics of SleekBot V2.

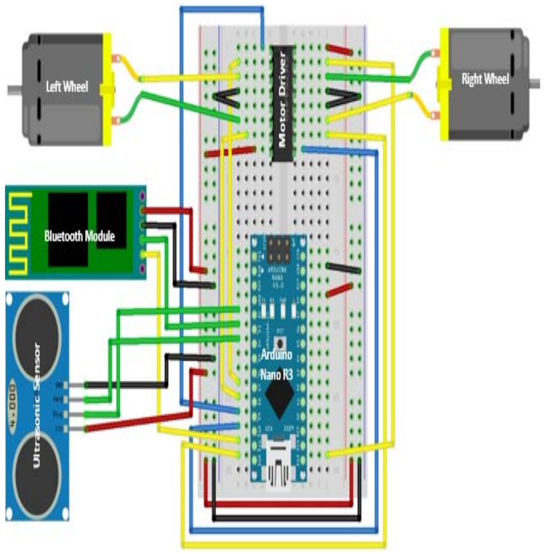


Figure 6: SleekBot V1 Breadboard schematics

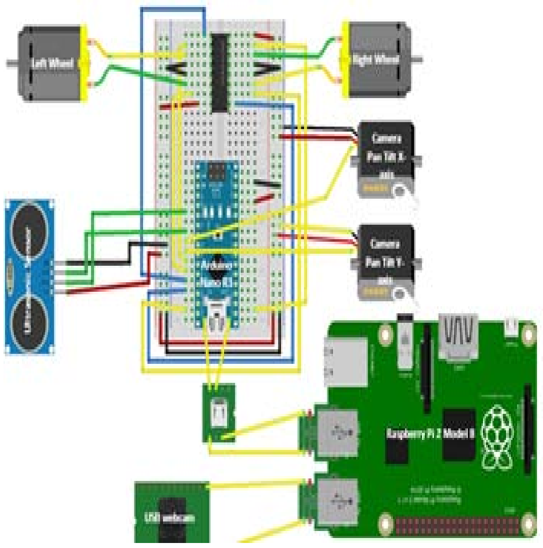


Figure 7: SleekBot V2 Breadboard Schematics

IV. Result and Discussion

a) System Implementation

The system was implemented in three stages, each stage corresponds to the deployment of each component that made up the system.

i. *Mobile Robot Gateway Stage*

Two types of Mobile Robot Gateway were implemented. The specific implementation used was determined by the robot's ability to connect directly to the internet. In a scenario where the Mobile Robot can connect directly to the internet, a python program was used to implement the Gateway. The second implementation was used when the Mobile Robot connects to the internet through another computer. A desktop application written in C# was used to implement the Gateway on the computer with internet connectivity.

ii. *Command Hub Stage*

At this stage of implementation, a web API written in C# was used to implement a REST-Based service that acts as the command hub for the system. The service was deployed to Internet Information Service (IIS) with MS SQL Server 2012 to act as the backend storage for the system.

iii. *Command Interface Stage*

HTML 5, JavaScript, and CSS 3 were used with Apache Cordova and Ionic framework to create a cross-platform mobile application for Mobile Robot remote control. Also at this stage, a web application written in C# and ASP.Net MVC 4 was used to implement a web-based Remote control Interface

b) Implementation Tools

i. *Hardware*

Arduino Nano R3: this is a programmable development board based on the ATmega328 microcontroller chip. It has 32 KB flash Memory, 2 KB SRAM, 1 KB EEPROM, 14 digital I/O pins of which 6 provide PWM (Pulse Width Modulation output), 8 analog input pins, and 16 MHz clock speed. It was used to develop the SleekBot V1 Mobile Robot.

Raspberry Pi 2 Model B: this is a credit card size computer capable of running a trim-down version of the Linux operating system. It has a quad-core processor with a clock speed of 900 MHz, 1 GB RAM, and 40 GPIO (General-purpose Input Output) pins. This was used with Arduino Nano R3 to develop SleekBot V2 Mobile Robot

ePuck Mobile Robot: this a differential drive educational mobile robot based on a dsPIC30F6014A microcontroller chip running at 60 MHz clock speed. It has 8 KB RAM, 144 KB flash Memory, 3D accelerometer, VGA camera, 8 infra-red sensors, Bluetooth for wireless communication with a computer, speaker and LED. It was used as one of the Mobile Robots used in testing the workability of the developed remote control system.

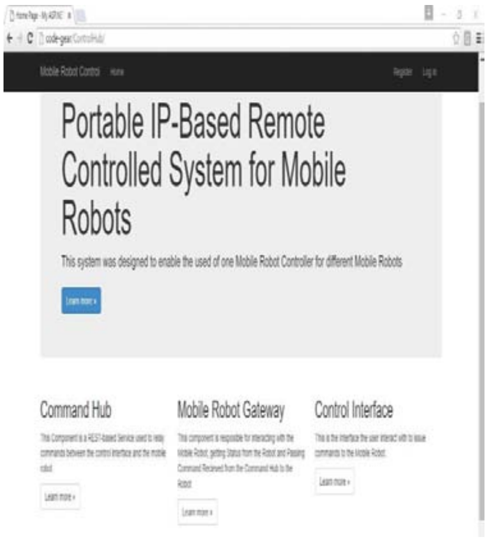
Software

Programming Languages: C, C#, Python, JavaScript
Interface Design: HTML 5, CSS 3, Win form
Backend: MS SQL Server 2012
Framework: .Net Framework, Apache Cordova, Ionic Framework
Web Server: Internet Information Service (IIS) 8

ii. *User Interface Documentation*

Home Page

For web interface users, the system provides a landing page where navigation to other modules is accessed. Figure 8 shows the home page layout.



Portable IP-Based Remote Controlled System for Mobile Robots

This system was designed to enable the used of one Mobile Robot Controller for different Mobile Robots

Command Hub

This Component is a REST-Based Service used to relay commands between the control interface and the mobile robot

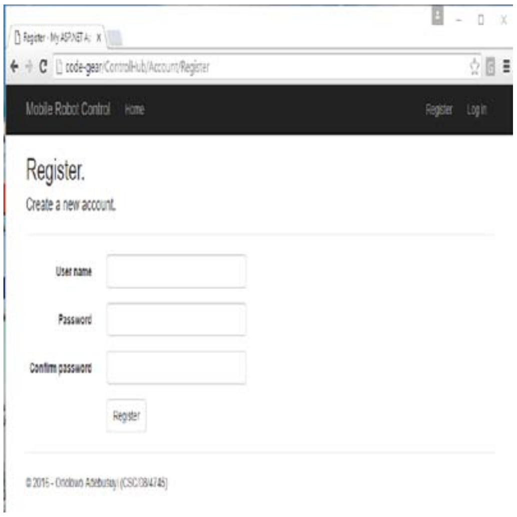
Mobile Robot Gateway

This component is responsible for interacting with the Mobile Robot, getting Status from the Robot and Passing Command Recieved from the Command Hub to the Robot

Control Interface

This is the interface the user interact with to issue commands to the Mobile Robot.

Figure 8: Home Page



Register.

Create a new account.

User name

Password

Confirm password

© 2015 - Onidowo Adebusayi (CSC/08/4745)

Figure 9: Web Registration Interface

Global Journal of Computer Science and Technology (D) Volume XXI Issue I Version I Year 2021

User Registration

This module is used to register all users making use of the control platform. All valid users of the system need to be authenticated before they are granted access to control any mobile robot remotely. Figure 9 shows the user registration interface for web interface users while figure 10 shows the registration interface for mobile users.

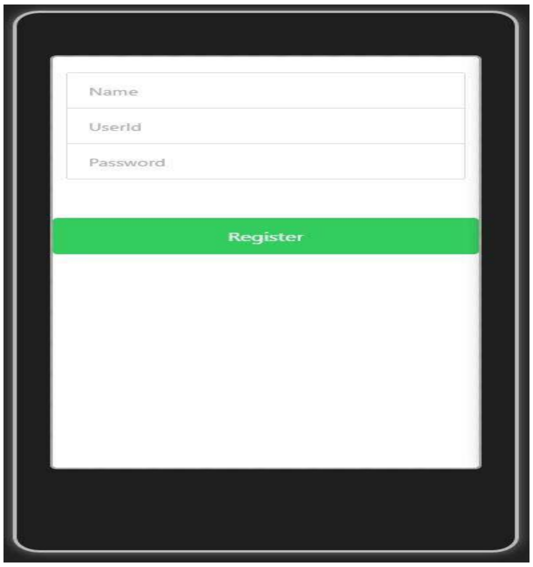
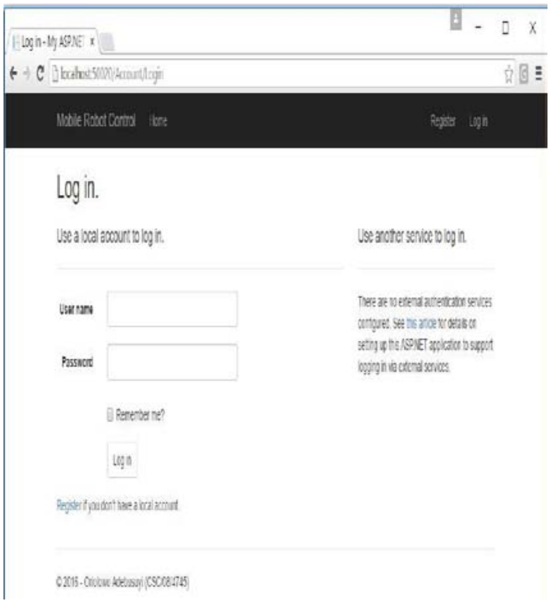


Figure 10: Mobile Registration Interface

User Login

This module is used for authenticating all users making use of the system. All users must pass through this module to use the system. Figure 11 and figure 12 show the login user interface (UI) for web and mobile users.



Log in.

Use a local account to log in.

User name

Password

Remember me?

Log in

Register if you don't have a local account

Use another service to log in.

There are no external authentication services configured. See this article for details on setting up this ASP.NET application to support logging in via external services.

© 2016 - Ololuwa Adebusayi (CSC/08/4745)

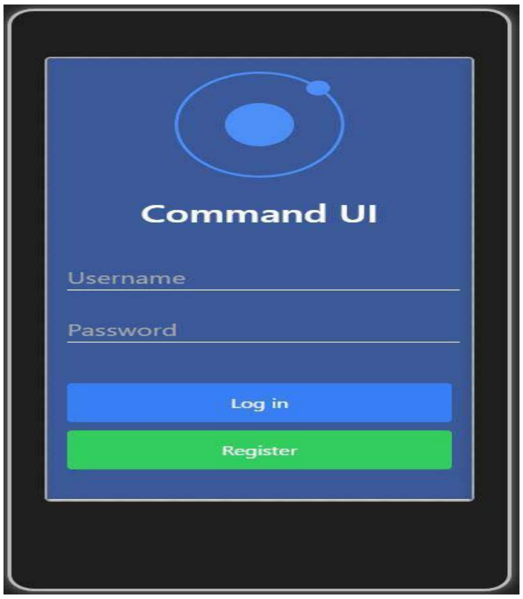


Figure 11: Web UI

Figure 12: Mobile UI

Main Menu

The main menu provides navigation to important modules the user can use to perform different tasks in the system. Figure 13 shows the main menu for mobile interface users

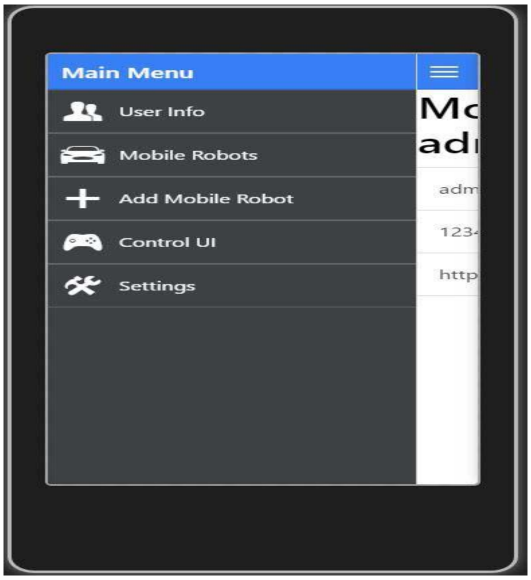
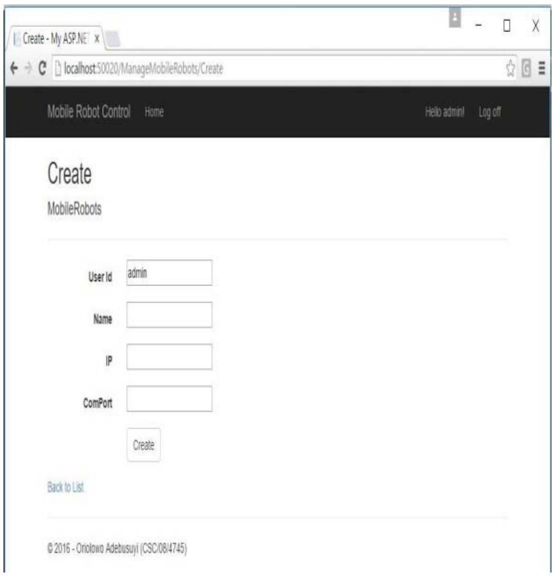


Figure 13: Main menu


Create - My ASP.NET

localhost:50020/ManageMobileRobots/Create

Mobile Robot Control Home Hello admin! Log off

Create

MobileRobots

User Id: admin

Name:

IP:

ComPort:

Create

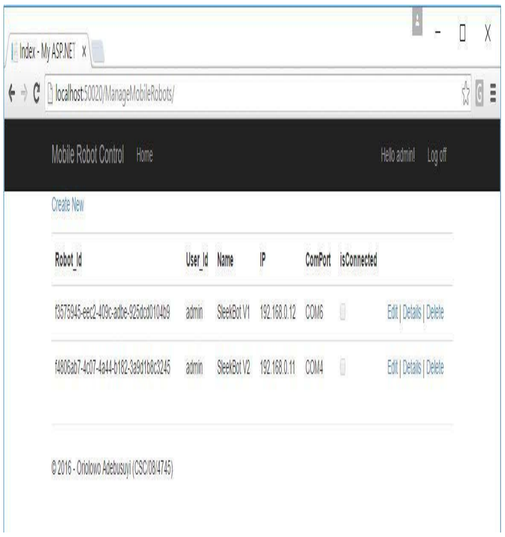
Back to List

© 2016 - Oriolowo Adebusuyi (CSC/08/4745)

Figure 14: Mobile Robot Registration

Mobile Robot Registration

To make use of any robot on the platform, the user needs to register the robot. Figure 14 shows the interface dedicated to the registration. After registration, all registered Mobile Robots by the user is displayed in the Mobile Robot List as shown in figure 15.


Index - My ASP.NET

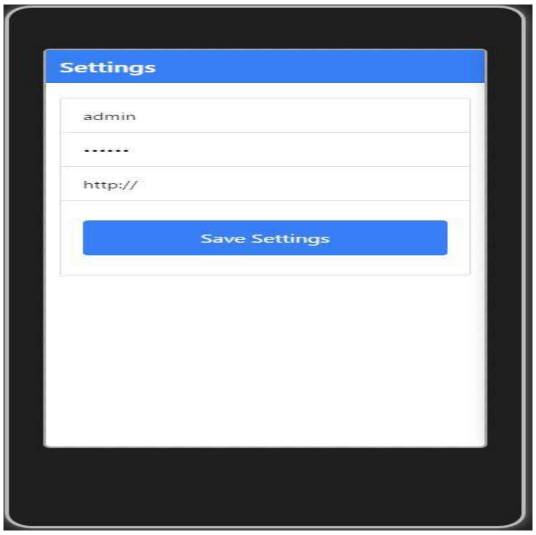
localhost:50020/ManageMobileRobots/

Mobile Robot Control Home Hello admin! Log off

Create New

Robot_Id	User_Id	Name	IP	ComPort	isConnected	
f3575945-eec2-409c-adbe-925dcd0104b9	admin	SleekBot V1	192.168.0.12	COM6	☐	Edit | Details | Delete
f4806ab7-4c07-4a44-b182-3a9d1b8c3245	admin	SleekBot V2	192.168.0.11	COM4	☐	Edit | Details | Delete

© 2016 - Oriolowo Adebusuyi (CSC/08/4745)


Settings

admin

••••••

http://

Save Settings

Figure 15: Registered Mobile Robots *Figure 16:* Settings Interface

Settings

The settings module is used to configure the parameters required for the interface to work properly. Figure 16 shows a settings Interface for mobile Users

Remote Control Interface

This interface provides the necessary widget to remotely control mobile robots. Actions performed on this interface translate to a command to be executed on the mobile robot. Figure 17 and figure 18 show the remote control interface for web and mobile interface users.

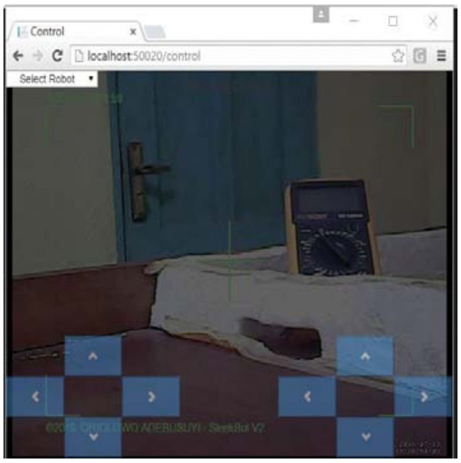


Figure 17: Remote control for web UI

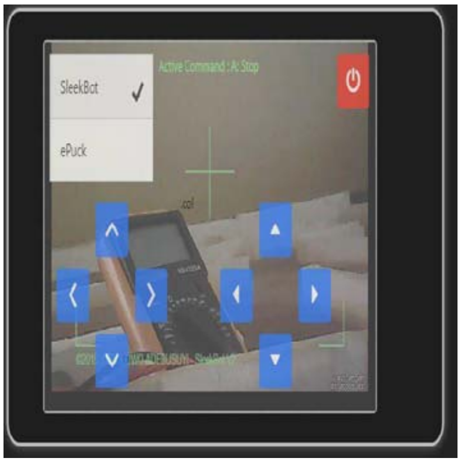


Figure 18: Remote Control for Mobile UI

c) System Evaluation

Algorithm for Command Hub

```
Initialize Command Hub
While interrupt NOT available
    Get HTTP request from client
    Map request to command controller
    Construct response message
    Return response message
End While
```

Time Complexity of Command Hub algorithm

```
Number of Operation    = 1 + 1 + N(1 + 1 + 1+ 1)
                       = 2 + N(4)
                       = 2 + 4N
Number of Operation    = 2 + 4N
In Big O notation, the algorithm executes in linear time i.eO(n)
```

Algorithm for Command Interface

```
Initialize Command Interface
While UserAction != null
    Construct command object
    Send command object to Command Hub using    HTTP Request
    Get Response for the HTTP Request
    If Response != null
        Process Response
        Update Interface
    End If
End While
While Interrupt NOT available
    Get Status from Command Hub
    Update Interface
End While
```

Time Complexity of Command Interface algorithm

```
Number of Operation    = 1 + 1 + N(1 + 1 + 1+ 1 + 1)  + N(1 +1)
                       = 2 + N(5) + N(2)
                       = 2 + 4N + 2N
Number of Operation    = 2 + 6N
In Big O notation, the algorithm execute in linear time i.eO(n)
```

Mobile Robot Gateway Algorithm

```
Initialize Gateway
While new Command IS available
```

```
        Get Command from Command Hub
        If Command != null
                Map command to the Mobile Robot Kinematics
                Execute Command
        End If
End While
While Interrupt NOT available
        Get Mobile Robot Status
        Send Status to Command Hub
End While
```

Time Complexity of Mobile Robot Gateway algorithm

Number of Operation $= 1 + N(1 + 1 + 1) + N(1 + 1)$

$= 1 + N(3) + N(2)$

$= 1 + 3N + 2N$

Number of Operation $= 1 + 5N$

In Big O notation, the algorithm executes in linear time i.eO(n)

The Time Complexity of the Entire System

Number of Operations $= 2 + 4N + 2 + 6N + 1 + 5N$

$= 5 + 15N$

Number of Operations $= 5 + 15N$

The Time Complexity of the entire system expressed in Big O notation = O(n)

V. Conclusion

The goal of this research work to develop a portable IP-based multi-interface remote-controlled system for mobile robots was achieved. The system offers the use of a single remote control device across different mobile robots and the use of a multi-interface for a single mobile robot.

The solution architecture is based on three loosely coupled components performing various tasks at different levels to collectively achieve a single aim of having a portable control system. At the core of the system is a REST-based web service handling communication between the user interface and the mobile robot. Various programming languages were used at different levels to achieve the overall goal of the system. A cross-platform mobile application, Web application, and the desktop client was developed to serve as the system's user interface.

Implementation of the system was carried out on three different mobile robots based on different platforms. The system was evaluated based on the time complexity of the algorithm used in its components. The result shows that the system time of execution is linear (O (n)).

The system developed is not without its weakness, hence the need to improve some parts that are currently inefficient in its mode of operation. These include:

1. The visual Feedback(Video Streaming) uses Motion Jpeg standard which has unacceptable lag time for network connection speed lower than 256 Kbps
2. iWeb Control UI does not work well with the Internet Explorer web browser
3. Better implementation algorithm to reduce the system time complexity from O(n) to O(1)

References Références Referencias

1. Abdullah, M.F.L., and Poh, L.M. (2011). Mobile Robot Temperature Sensing Application via Bluetooth. International Journal of Smart Home Vol. 5, No. 3, July 2011
2. Adamides, G., Katsanos, C., Christou, G., Xenos, M., Papadavid, G., and Hadzilacos, T. (2014). User interface considerations for telerobotics: The case of an agricultural robot sprayer. In Second International Conference on Remote Sensing and Geoinformation of the Environment (RSCy2014) (pp. 92291W-92291W). International Society for Optics and Photonics.
3. Almali, M. N., Gürçam, K., Bayram, A. (2015). Wireless remote control of a mobile robot. International Journal of Scientific Research in Information Systems and Engineering (IJSRISE),
4. Ankit, J., Hemant, Y., Raj, K. Y., and Rajendra, S. (2014). Controlling of Remote Robot through mobile phone using DTMF Signal. International Journal of Science, Engineering and Technology Research (IJSETR), Volume 3, Issue 4, April 2014.
5. ATIS (2001). Telecom glossary "bot". Alliance for Telecommunications Solutions. Retrieved from http://www.atis.org/tg2k/_bot.html
6. Bekker, M. G. (1960). Off-The-Road Locomotion. Ann Arbor: University of Michigan Press.
7. Bekker, M. G. (1969). Introduction to Terrain Vehicle Systems. Ann Arbor: University of Michigan Press.
8. Chikurtev, D., Rangelov, I., Yovchev, K., &Chivarov, N. (2019). The communication system for remote

control of service robots. IFAC-PapersOnLine, 52(25), 186-191. https://doi.org/10.1016/j.ifacol.2019.12.470

9. Chin, E. L., Chih-Ching, L., An-Sang, H., and Chih-Chen, W. (2003). A Real-Time Remote Control Architecture Using Mobile Communication. IEEE transactions on instrumentation and measurement, vol. 52, no. 4, August 2003.
10. Dutta, V., and Zielinska, T. (2015). Networking technologies for robotic applications. arXiv preprint arXiv:1505.07593.
11. Esteller-Curto, R., Cervera, E., Del Pobil, A. P., Marin, R. (2012). Proposal of a REST-based architecture server to control a robot. In Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference (pp. 708-710).
12. Gates, B. (2007). A robot in every home. The leader of the PC revolution predicts that the next hot field will be robotics. Sci Am 296:58–65
13. Goodrich, M. A., and Schultz, A. C. (2007). Human-robot interaction: a survey. Foundations and Trends in Human-Computer Interaction, 1(3), pp 203–275.
14. Harris, T. (2007). How Robots Work. How Stuff Works. Retrieved from http://science.howstuffworks.com/robot.htm
15. Hockstein, N. G., Gourin, C. G., Faust, R. A., Terris, D. J. (2007).History of robots: from science fiction to surgical robotics. Journal of robotic surgery, 1(2), 113-118.
16. James, D. M. (1997). Bringing Portability to the Software Process. West Virginia University, Dept. of Statistics and Computer Science.
17. Jose, S., Sebastián, D., Rafael, P., Fernando, M. (2004). A Java/Matlab-Based Environment for Remote Control System Laboratories: Illustrated With an Inverted Pendulum. IEEE transactions on education, vol. 47, no.
18. Juang, S. Y., &Juang, J. G. (2016). Remote control of a mobile robot for indoor patrol. Applied Sciences, 6(3), 82. https://doi.org/10.3390/app6030082.
19. Lambèr, R., and Rinie, V.E. (2015). A Literature Review on New Robotics: Automation from Love to War. International Journal of Social Robotics 2015: 295.
20. Lung, N., Wyatt, S. N., Vincenzo, L. (2002). An Experiment in Internet-Based, Human-Assisted Robotics. Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference, Vol. 2, pp. 2190-2195.
21. Luimula M., Saaskilahti K., Partala T., Pieska S., Alaspaa J. and Lof A. (2007), "Improving the Remote Control of a Mobile Robot Using Positioning and Ubiquitous Techniques," 2007 IEEE International Conference on Automation Science and Engineering, Scottsdale, AZ, 2007, pp. 1027-1033, DOI: 10.1109/COASE.2007.4341744.
22. Lydia, P., David, T. B., Chuck, D., Jason, F., Wei, L., Carolyn, M., Nicolas, R. (2006). TCP/IP Tutorial and Technical Overview. International Technical Support Organization. ISBN 0738494682.
23. Nicholas, W. (2009). Robots in space. Retrieved from http://www.universetoday.com/43750/robots-in-space
24. Obe, O.O., and Dumitrache, I., (2012). Adaptive Neuro-Fuzzy Controller with Genetic Training for Mobile Robot Control. International Journal of Computer Communications and Control, No.1, Vol.7, 149-156
25. Oboe, R., and Fiorini, P. (1998). A design and control environment for internet-based telerobotics. International Journal of robotics research, vol 17(4), pp 433-449
26. Oboe, R., and Fiorini, P.(1997). Internet-Based Telerobotics Problems and Approaches: Advanced Robotics, 1997. ICAR '97. Proceedings., 8th International Conference.
27. Oxford dictionary (2015, June 13). remote control - definition of remote control in English from the Oxford dictionary. Retrieve from http://www.oxforddictionaries.com/definition/english/remote-control
28. Oxford Dictionary (2016). Robot - definition of a robot in English from the Oxford dictionary. Retrieved from http://www.oxforddictionaries.com/definition/english/robot
29. Robot Institute of America in (1979) Robotics - Encyclopedia - Business TermsRetrieved from https://www.inc.com/encyclopedia/robotics.html
30. Polk, I.(2005). RoboNexus 2005 robot exhibition virtual tour. Robonexus Exhibition 2005. Retrieved from http://www.virtuar.com/click/2005/robonexus/index.htm
31. Posadas, J.L., Poza, J.L., Simo, J.E., Benet, G., Blanes, F. (2008). Agent-based distributed architecture for mobile robot control. Engineering Applications of Artificial Intelligence 21 (2008) 805–823.
32. Pravin, V., and Shalini, T. (2013). Accelerometer Based Hand Gesture Controlled Robot. International Journal of Science and Research (IJSR), ISSN (Online): 2319-7064
33. Robin, R.M. (2000). Introduction to AI Robotics. Massachusetts Institute of Technology. ISBN 0-262-13383-0
34. Robotiq (2014). Industrial Robots: 5 Most Popular Applications. Retrieved from http://blog.robotiq.com/bid/52886/Industrial-robots-5-most-popular-applications
35. Royakkers, L., and van-Est, R. (2015). A literature review on new robotics: automation from love to war. International journal of social robotics, 7(5), 549-570.

36. Siegwart, R., and Saucy, P. (1999, May). Interacting with mobile robots on the web. In IEEE International Conference on Robotics and Automation (ICRA).
37. Sparrow, R (2007). Killer robots. J ApplPhilos 24(1):62–77. Talos Encyclopedia Mythica. Retrieved April 29, 2016, from Encyclopedia Mythica Online. http://www.pantheon.org/articles/t/talos.html
38. Turkle, S (2011). Alone together. Basic Books, New York, Why we expect more from technology and less from each other
39. Wichmann, A., Okkalioglu, B. D., and Korkmaz, T. (2014). The integration of mobile (tele) robotics and wireless sensor networks- A survey. Computer Communications, 51, 21-35.
40. Zunt, D.(2007). Who did actually invent the word "robot" and what does it mean? The Karel Capek website. Retrieved from http://capek.misto.cz/ english/robot.html.