TeleBot: A Teleoperated Autonomous Robot

Shuvajit Das[1], Mohammed Moshiul Hoque[1], Sudipta Majumder[2]

[1]Department of CSE, Chittagong University of Engineering & Technology, Chittagong,Bangladesh

[2]Department of CSE, AhsanullahUniversity of Science & Technology, Dhaka, Bangladesh

Email address: dshuvajit@gmail.com, mmoshiulh@gmail.com, piash.aust@gmail.com

Abstract— Teleoperation means simply to operate a vehicle or a system over a distance. Teleoperated robots are controlled remotely by a human being. The remote control signals can be sent through a wire, through a local wireless system (like, Wi-Fi), over the Internet or by satellite. This paper describes the design of a teleoperated robotic system that can control remotely through a computer over a Wi-Fi link using TCP/IP. Specifically, the robot can control in four directions namely left, right, backward and forward from 120 meters far away. Evaluation results in terms of subjective and quantitative ways reveals that the proposed robotic system is functioning well.

***Keywords*— Teleoperation, Microcontroller, Robot, Wi-Fi, DC Motor.**

I. INTRODUCTION

Teleoperation indicates operation of a machine at a distance [1, 2]. Teleoperated robots are controlled remotely by a human being. The remote control signals can be sent through a wire, through a local wireless system (like Wi-Fi), over the Internet or by satellite. In a telerobotic system, a human operator controls the movements of a robot from some distance away. Signals are sent to the robot to control it; other signals come back, telling the operator that the robot has followed the instructions. The advent of new high-speed technology and the growing computer capacity provided realistic opportunity for new robot to reduce the human effort and realization of new methods of control theory of robot [3]. This technical improvement together with the need for high performance robots created faster, more accurate and more intelligent robots using new robots control devices, new drivers and advanced control algorithms. The teleoperated robot system can be used for different sophisticated robotic applications. Recently, intelligent machines are wirelessly controlled form remote places. Robots are commonly used instead of humans in many applications. Still there are numerous fields where robots in combination to the wireless communication can be deployed for better results and efficiency. More often researchers, explorers, soldiers and other people comes across places which cannot be accessed due to our physical limitations, unknown conditions of that place or some other reasons. Therefore, a robot which can be controlled wirelessly along with various supporting features can provide a good level of accessibility to us in risky or hazardous areas.

Teleoperated robot and human will co-exist with sharing and cooperating tasks according to the command provided by the human to the robot. A teleoperated robot can control remotely for performing such environments where human access is quite impossible or threats of human lives. This robot can be used in many important areas such as, in military applications for tracking underwater vehicles, for finding mine in fields, in anti terrorist activities, in medical application for surgery, in rescue operations, and so on. Assisted teleoperation can provide multiple types of assistance to a human user, such as collision avoidance, tremor filtering, targeting precision, path tracking, orientation control, dynamics compensation, and trajectory smoothing, and can provide this assistance in multiple ways, such as autonomous low-level control, information displays, and haptic feedback.

This paper describes on designing a teleoperated robot (hereafter called 'TeleBot') that can control through computer over Wi-Fi technology. We also develop TeleBot's control software to control it in four directions (i.e., front, back, left and right). Experimental evaluation confirms the effectiveness of the proposed robot's architecture in the controlled environment.

II. RELATED WORK

Teleoperated robot arms are widely used in space robotics, bomb disposal, remotely operated vehicles, and robotic surgery, but are usually direct-controlled by the human operator. Direct joint control is widely used but slow, tedious, and unintuitive. Haptic input devices, such as those used in robot surgery, are more intuitive, but are expensive, highly specialized, and have limited workspaces. In both cases, the user is completely responsible for guiding the robot's motion, which requires extensive training and constant attention in order to achieve low-level motions and to avoid collisions.

Gupta et al. [4] presents the design of a controller intended for teleoperation. It is capable of controlling an anthropomorphic robotic arm through a LAN or via the Internet. The user can control the robotic arm remotely and access its sensory feedback signals as well. Supervisory control assigns the user a managerial role over the robot's largely autonomous behaviour [5]. Collaborative control treats the user and robot as peers that must resolve conflicts using negotiation and dialogue [6]. Shared control gives the robot control of some known dimensions of a task to be handled at a fast update rate [7, 8]. Virtual fixtures are a common shared control technique for manipulator arms that uses haptic feedback to help users control end effectors along specified paths, surfaces, or orientations [9]. Adjustable autonomy addresses the issue of choosing autonomy levels appropriate to preference, trust, skill level, or the demands of a given situation [10, 11]. The UBot-5, [2] is a general purpose robot that has been adapted for use in assisting elderly patients by allowing doctors and caregivers to monitor them and to control the robot to perform some basic tasks. These robots are typically controlled by a surgeon or doctor and allow her to perform various tasks and treatments that she would not normally be able to do.

III. TELEBOT ARCHITECTURE

In this work, microcontroller (PIC16F877A) will be used to control the TeleBot's direction (clockwise and anticlockwise). The controlling device for the robotic controlling in this project will be a Microcontroller.

A. Hardware Setup

The data sent from CPU over Wi-Fi will be received by Wi-Fi module connected to Microcontroller. Microcontroller reads the data and decides the direction and operates the dc motors connected to it accordingly. Wireless transceiver is a special kind of device which is used to interface Wi-Fi with microcontroller. It converts a Wi-Fi Signal to UART (Universal Asynchronous Receiver/Transmitter) signal. Microcontroller is a small device which has CPU (Central Processing Unit), RAM (Random Access Memory) and EEPROM (Electrically Erasable Read Only Memory) built in it. Different Robots used different actuators. Here Motor is used as an actuator. Actuator controller is a small IC (Integrated Circuit) to control the speed of the actuator. Ethernet interface is a card built in the computer. Wireless router is a device that converts the Ethernet signal into Wi-Fi Signal and vice-versa. Sensor will give the user the information about the surroundings of the robot. The robotic system also used microcontroller programmer, DC motor, motor controller (L298), power source (4.5 volt battery), crystal oscillator, PCB wiring and some passive electronics. Fig. 1 shows the schematic diagram of the TeleBot system.

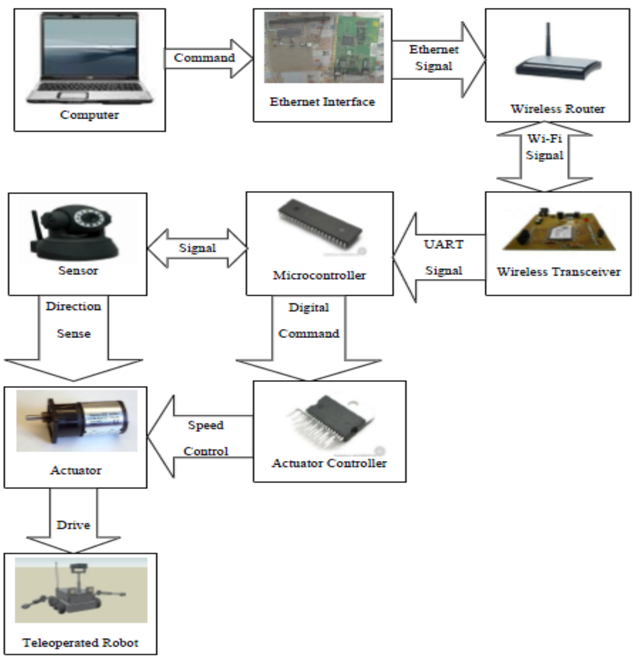

Fig.1 Proposed Schematic Diagram of Working Procedure.

The TeleBot is working as a closed-loop with real time control system. The dimensions of TeleBot are 25 cm (length) x 26 cm (width) x 12 cm (height). The diameter of the wheel is 5.5 cm. Fig. 2 illustrates a prototype of the TeleBot robot.

B. Software Modules

The Microcontroller is programmed used embedded 'C' language and the graphical representation of the robot's direction is programmed using Java Runtime Environment. The embedded software is actually a C compiler to convert the C program into assembly language. An interfacing software is used to simulate the microcontroller as a part of implementation. Controller software is written in Java and network socket of the Java will be used. The controlling software module is depicted in Fig. 3. In order to run the system, we have used Windows XP general purpose computer installed with several softwares: PIC kit 2 programmer for dumping code into micro controller, PIC-C compiler for embedded C programming (MikroC_PRO_PIC_2011_Build.4.60), and Proteus for hardware simulation (Lab-centre electronics Proteus 7.8 SP2).

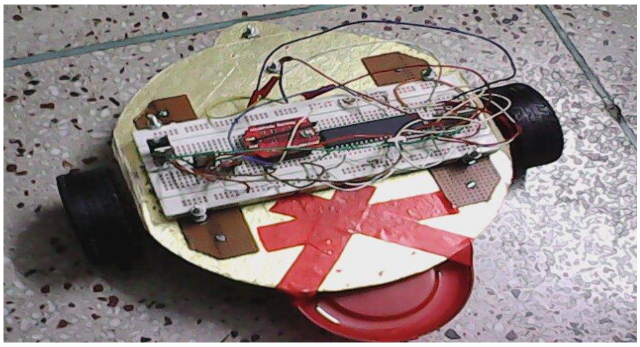

Fig. 2 A prototype of the TeleBot.

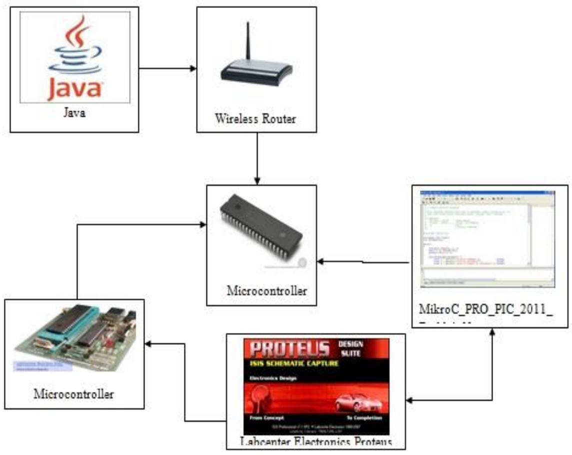

Fig. 3 Integration of various software modules.

IV. IMPLEMENTATION DETAILS

The implementation of the TeleBot carried out into three major steps: Wi-Fi link development, interface development (GUI) and integration of modules. We have used several hardwires to drive the system. Some of these are described in the following sections.

A. Microcontroller (PIC16F877A)

The microcontroller acts like the brain of the DC motor speed control system. The microcontroller chip that has been selected for the purpose of controlling the speed of DC motor is PIC16F877A manufactured by Microchip. Its size is small and equipped with sufficient output ports without having to use a decoder or multiplexer. It has portability and low current consumption. It has PWM inside the chip itself which allow us to vary the duty cycle of DC motor drive.

B. Differential Drive

By using two motors we can move our robot in any direction. This steering mechanism of robot is called as differential drive [12]. Table I shows the working patterns of this drive.

TABLE I
DIRECTION PATH OF EACH MOTOR FOR ROBOT MOVEMENT

Left Motor	Right Motor	Robot Movement
Straight	Straight	Straight
Stop	Straight	Left
Reverse	Straight	Sharp Left
Straight	Stop	Right
Straight	Reverse	Sharp Right
Reverse	Reverse	Reverse

C. L-298(DC Motor Drive)

The DC motor drive that will be used in this project is a dual full bridge driver, chip L298. The operating supply voltage of chip L298 is up to 46V and the total DC current up to 4A. The time to enable the chip L298 will be determined by the duty cycle pulse that sent from PWM in microcontroller.

D. RN-171-XV 802.11 b/g Wireless LAN Module

The RN-XV is the 802.11 b/g solution especially designed for customer who wants to migrate existing 802.15.4 architecture to a more standard TCP/IP based platform without having to redesign their existing hardware. Based on a pseudo standard footprint often found in embedded applications, the RN-XV module from Roving Networks allows for Wi-Fi connectivity using 802.11 b/g standards in legacy and existing designs that may have been based upon 802.15.4 standard.

The RN-XV module is based upon Roving Networks' robust RN-171 Wi-Fi module and incorporates 802.11 b/g radio, 32 bit SPARC processor, TCP/TP stack, real-time clock, cryptoaccelerator, power management unit and analogue sensor interface.

E. Software Implementation

For software implementation, MikroC_PRO_PI0C_2011_Build.4.6is used to program microcontroller in assembly language. Besides, Java Runtime Environment is used for use interface purpose and for monitoring the direction response of the system. A simulation software Proteus is used to give an overall view of the actual system.

F. Computer Interface (GUI)

The computer interface is designed in Java. It uses TCP/IP communication to transmit control signals to the router. The reason why we didn‘t opted for the UDP is that it doesn‘t have a flow control and also that TCP is a standard protocol for Wi-Fi. Java has a built-in component or an ActiveX. It enables us to communicate through LAN using TCP/IP protocol.

G. Microcontroller Design

Microcontroller used in this paper is a simple one that is (PIC 16F877A). Microcontroller will receive the desired destination command from user through PC that interface with Wi-Fi Shield. The actual speed will be compared with the desired speed and the correction will be done by microcontroller to always maintain the DC motor speed at the desired speed. An algorithm has to be developed to make the microcontroller to read the input and respond accordingly. The overall process can be represented by a flowchart as in Fig. 4. The flowchart is then translated into assembly language and compiled using MikroC, one of the PIC16F877A software development tool.

The microcontroller is connected with the router, when router receives a control signal it passes it on to the microcontroller which then handles the further operation. Microcontroller compared the received values with the ones already coded in it and does the corresponding operation. For example if it receives value 8 it sends a ‘1’ to a pin responsible for the forward movement of the robot. As for testing purposes a 3.3v coming from router were converted to 5v so now it cannot be fed directly to the microcontroller and it needs to be converted back so router is again used with the microcontroller.

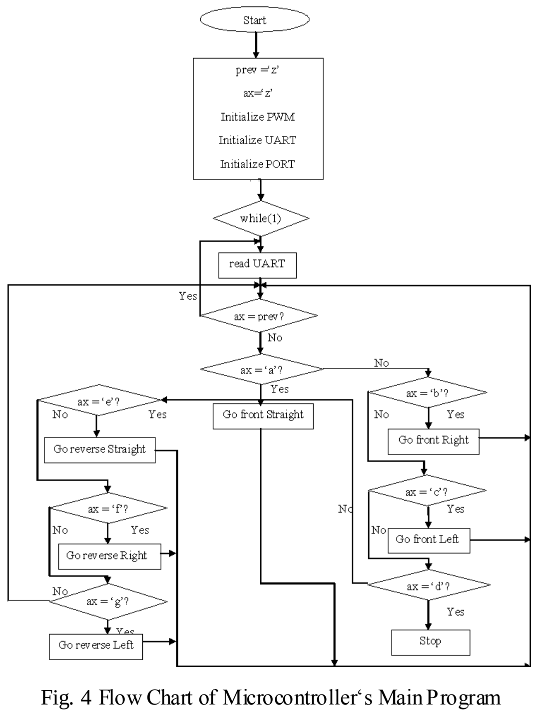

Fig. 4 Flow Chart of Microcontroller‘s Main Program

H. Proteus Software (Proteus 7.8 SP2)

Proteus is the UK Labcenter developed circuit analysis and physical simulation and printed circuit board design software which can simulate, analyse analogue circuits and integrated circuits. Proteus software provides lots of analogue and digital components and peripherals. In particular, it has the ability to interactively simulate microcontroller and its peripheral circuits as an integrated system. Fig. 5 shows the connection flow between various events created in Java Runtime Environment to perform various actions of the TeleBot.

I. TeleBot Circuit Design

The circuit implied in this project is a complex one. The total circuit design can be divided into three phase namely power circuit design, connection between microcontroller and Wi-Fi module and connection between microcontroller and motor controller. These circuits are also inter-linked with each other.

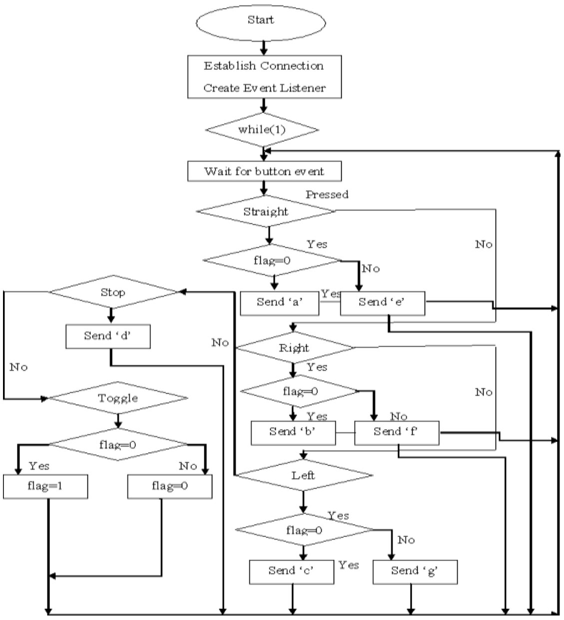

Fig. 5 Flow Chart of program in Java Runtime Environment

Fig. 6 shows the different view of our TeleBot system. Three types of view namely top view, front view and side view are depicted in Fig. 9.

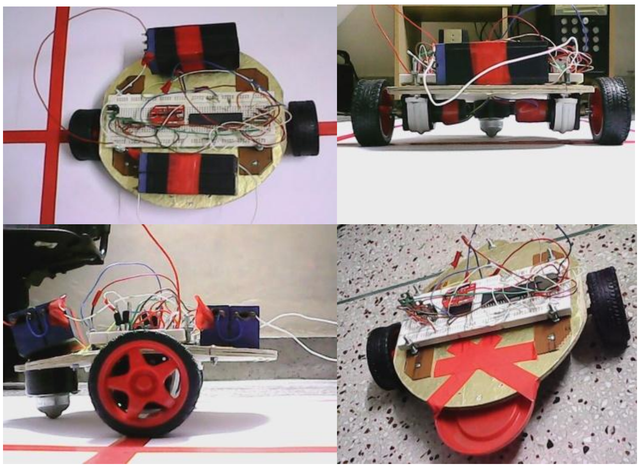

Fig.6 Different views of TeleBot: top, front, and side views.

1) Power Circuit Design:

There are three IC used in theseproject to construct the power circuit. They are IC LM7809, ICLM7805 and IC LM317T. Firstly, the 18V generated from the batteries (4 * 4.5 Volt Battery) becomes the input of IC 7809 which produces 9V as output which supplies 9V to the actuator controller (Pin 4 works as the Supply Voltage, V_S in IC L298). The output also becomes the input of IC 7805 which generates 5V as output which supplies 5V to V_{DD} (Pin 32) of MCU(PIC 16F877A) and to actuator controller to give the logic supply voltage (Pin 9 is used as the Logic Supply Voltage, V_{SS} in L298). The output of 7805 also becomes the input of the IC LM317T which produces 3.3V as output. The output of the LM 317T provides the input voltage of RN-171-XV Wi-Fi Module (Pin 1 is the V_{DD} of Wi-Fi Module).Thus, the power circuit design is completed.

2) Connection details of Circuit between Microcontroller and Wi-Fi Module:

There are total 20 pins in RN-171XV 802.11b/g Wireless LAN Module. Among them, 5 pins are used in this project. From them, 2 pins are needed to make connection with microcontroller PIC 16F877A.They are V_{DD}(Pin 1), TXD(Pin 2),RXD(Pin 3), GPIO 9(Pin 8) and GND(Pin 10). The TXD of Wi-Fi Module is connected with the RX (Pin 26) of MCU to receive the command transmitted from the Wi-Fi Device. Similarly, the RXD of the Wi-Fi Module is connected with the TX (Pin 25) of microcontroller to give the feedback response provided by the Wi-Fi shield. But, TX pin of microcontroller has to be voltage divided as MCU is operated at 5V while Wi-Fi is operated at 3.3 V. Three 220Ω (for voltage division) and one 1MΩ resistance (for ground reference) are used in this respect. The GPIO 9 (General Purpose Input-Output) of Wi-Fi is connected to its V_{DD} to enable the ad-hoc mode for creating its own network which is 3.3 V tolerant. If a wireless router is used via the Ethernet port of the Laptop, then GPIO 9 pin does not need to be connected. The MCLR (Pin 1) of microcontroller is connected with power supply, +5V(IC LM7805) via a 1MΩ resistance to clear the memory at the initial position and reset it after each command executed.

3) Connection details of Circuit between Microcontroller and Motor Controller:

There are 15 pins in DC motor Drive (ICL298). As we have used two actuators, so that input1 (Pin 5) and input2 (Pin 7) are the input of the first motor that are connected to the RD0 (Pin 19) and RD1 (Pin 20) of MCU. Similarly, the input3 (Pin 10) and input4 (Pin 12) are the input of the second motor that are connected to the RD2 (Pin 21) and RD3 (Pin 22) of microcontroller. The output of the first motor is obtained at the output1 (Pin 2) and output2 (Pin 3) while the output of the second motor is obtained at the output3 (Pin 13) and output4 (Pin 14). The Enable A (Pin 6) and Enable B (Pin 11) of IC L298 are connected to the CCP1 (Pin 17) and CCP2 (Pin 16) of PIC 16F877A to generate the PWM1 and PWM2 as output. The Current Sensing A and Current Sensing B of the motor controller are made grounded. The OSC1 (Pin 13) and OSC2 (Pin 14) of Microcontroller are used to generate clock pulse for the MCU. All the GND pin of all the ICs is connected to the ground of the battery.

J. Implementation Interface

There are two implementations in the TeleBot system. They are Hardware implementation and Software Implementation.

1) Hardware Implementation Interface:

Proteous Simulation software is used to build the hardware interface. In TeleBot system, as we mentioned before that connection circuit design and power circuit design are the two parts to implement. Connection circuits are developed not only between the microcontroller and motor controller but also between the microcontroller and the Wi-Fi module. As Wi-Fi module is not available as a functional symmetric design in Proteous, so that we have tried to use a second microcontroller as a Wi-Fi Module to provide UART signal between the two. Moreover, the microcontroller and the motor controller are connected to control the DC Motor direction by the input command given by the MCU becomes the output of the motor controller. Fig. 7 shows the Hardware Implementation Interface.

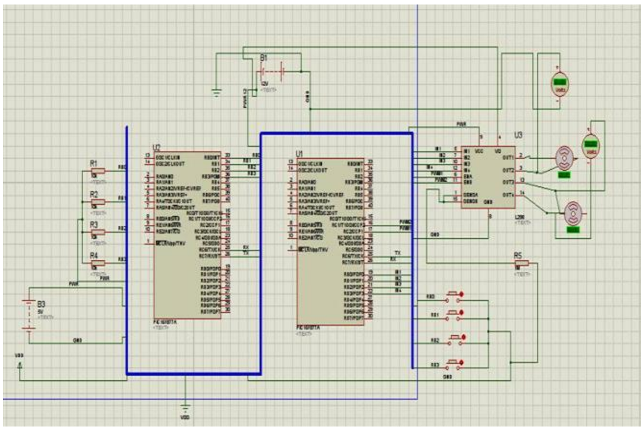

Fig. 7 Hardware implementation interface using Proteous simulator.

2) Software Implementation Interface:

The software interface is designed in Java Runtime Environment by creating socket using Java Network Programming and an interface window is designed in Java Swing Applet where some events are used to create the platform for various actions performed by the TeleBot. The software implementation interface window is depicted in Fig. 8.

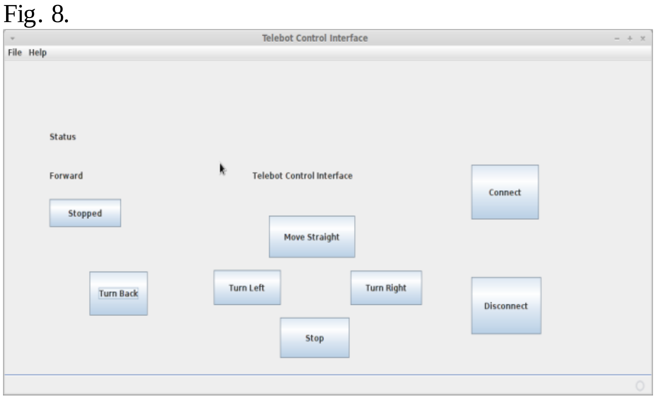

Fig.8 Software implementation interface window.

The various actions performed by the TeleBot can be possible by pressing the button of the interface window. Such as (i) *Connection Button Event:* creates the socket for connecting the Wi-Fi module to send various data to the microcontroller, (ii) *Move Straight Button Event:* after pressing this button, the TeleBot will move forward from its current position, (iii) *Turn Left Button Event:* By pressing this button the TeleBot will turn left from its current position, (iv) *Turn Right Button Event:* This button moves the Robot to Right from its current position, (v) *Turn Back Button:* This button performs two events. After pressing this button the Robot stops at first. Then if we want to change the mode either to move front or reverse we have to choose by clicking once again to the Turn back button thorough which the status of Robot's direction can be changed, (vi) *Stop Button Event:* This event stops the robot's movement from its running position and (vii) *Disconnect Event Button:* disconnects the TCP/IP connection between the software interface and the Wi-Fi device.

V. EVALUATION EXPERIMENT

In order to evaluate the proposed robotic system, we have simulated an environment to perform the experiment. We have evaluated the performance of our developed system in terms of two ways: quantitative and subjective.

A. *Experimental Setup*

We simulated an environment in the multimedia lab of CSE department, Chittagong University of Engineering & Technology (CUET). The room was 120x40 cm long. The robot was set to a location (i.e., at black spot indicates the robot current position) and control was done from that position. The experimenter was provided a command as input and the robot generates the movement according to this command. Fig. 9 shows the simulated environment for the TeleBot's movement.

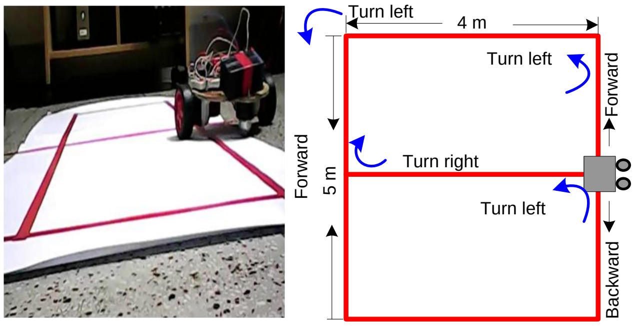

Fig. 9 Schematic set up for simulated environment.

B. *Quantitative Evaluation*

The quantitative evaluation includes the measurement of speed, load capacity, maximum distance travel by the TeleBot, and accuracy.

- *Speed*: The speed of the TeleBot per movement is measured by the speedometer. The Table II will give the illustration of the accuracy of speed. So the average speed of TeleBot obtained from the table is 0.03775 ms^{-1}.

TABLE II
SPEED CALCULATION FOR TELEBOT

No.	Movement Path	Speed(ms^{-1})
1	Move Straight	0.0378
2	Turn Left	0.0375
3	Turn Right	0.0381
4	Turn Back	0.0376

- *Load capacity:* The Robot's mass is 1.5 Kg including the power source. It can maintain the speed of average at this load. It was observed that the Robot stopped moving at 2.5 Kg. So its maximum load capacity is 2.5 Kg.
- *Maximum distance:* It can traverse 5.20 meter at one run * 10 times = 52 meter with the same power source.

- *Accuracy:* We tested our system by various input commands (for example *"Straight -> Left -> Back -> Right"*. Each action is performed 50 times. Most of the cases, we got the accurate response. Only for three cases, the system was not functioned properly. It may be happened due to the software malfunction and the Wi-Fi Shield could not interact with each other properly. We have used the Eq. (1) to calculate the accuracy of the robot performance.

$$\text{Accuracy, } E = (N_E / N_A) \times 100\% \quad (1)$$

Table III summarizes the results of the TeleBot system. By using this equation, we achieved the accuracy of the performance of TeleBot is 88%.

TABLE III
OVERALL SUCCESS RATE FOR THE TELEBOT'S OPERATION

Number of Total Command (N_T)	**Number of Total Actions performed (N_A)**	**Number of Actions that executed correctly (N_E)**	**Number of Incorrect Actions(N_I)**	**Success Rate (%)**
120	25	22	3	88

VI. CONCLUSION

When the technological state of the art has progressed, tasks carried out by remote human operators today could gradually be replaced with autonomous behaviour and local interaction. We evaluate the proposed TeleBot in simulated environment. The overall experimental result including subjective and quantitative experiment shows that the the robotic system is functioning quite well. We tried to deliver the most appropriate response the TeleBot should provide to the user. The feedback and the response comes from the user are so much satisfactory. There are various possible future enhancements of the existing system by incorporating metal detector, hurdle detection, auto rerun functionality, web interface application, computer vision sensor and so on that will not only increase its functionality but they will also increase the scope of its application

REFERENCES

[1] Teleoperation [Online]. Last day of access (September 25, 2013). Available: http://www.wikipedia.com/Teleoperation.htm

[2] A. M. Okamura, "Methods for haptic feedback in teleoperated robot-assisted surgery", in Ind. Rob., vol. 31, no. 6, pp. 499-508, 2004.

[3] Intelligent Robot controlled *by* Wi-Fi communication [Online]. Available: http://www.mycollegeproject.com.

[4] G. S. Gupta, S. C. Mukhopadhyay, and M. Finnie, *"Wi-Fi-based control of a robotic arm with remote vision"*, in Proc. International Conference on Instrumentation and Measurement Technology, pp. 557-562, 2009.

[5] W. Ferrel and T. Sheridan, "Supervisory control of remote manipulation". IEEE Spectrum, vol. 4, no. 10, pp. 81-88, 1967.

[6] T. Fong, C. Thorpe, and Charles Baur. Advanced interfaces for vehicle teleoperation: Collaborative control, sensor fusion displays, and remote driving tools. Autonomous Robots, vo. 11, pp. 77-85, 2001.

[7] D. A. Bell, S. P. Levine, Y. Koren, L. A. Jaros, and J. Borenstein, *Design criteria for obstacle avoidance in a shared-control system*, in Proc. of the RESNA, pp. 17-24, 1994.

[8] W. B. Griffin, W. R. Provancher, and M. R. Cutkosky, Feedback strategies for tele-manipulation with shared control of object handling force, Presence, vol. 14, no. 6, pp. 720-731, 2005.

[9] P. Marayong, M. Li, A. M. Okamura, and G. D. Hager, "Spatial motion constraints: Theory and demonstrations for robot guidance using virtual fixtures", in Proc. Intl. Conf. Rob. Automation, pp. 1270-1275, 2003.

[10] M. Desai and H. A. Yanco, Blending human and robot inputs for sliding scale autonomy, in Proc. IEEE Int. Workshop on Robot and Human Interactive Communication (ROMAN), pp. 537-542, 2005.

[11] D. Kortenkamp, D. Keirn-Schreckenghost, and R. P. Bonasso, "Adjustable control autonomy for manned space flight systems", in Proc. IEEE Aerospace Conference, pp. 629-640, 2000.

[12] K. N. Gupta, P .Agarwall and M .Agarwall, *Motor Driver L293D* [Online]. Available: http://www.rakeshmondal.info/L293D-Motor-Driver.