Development of a Mobile Robot as a Test Bed for Tele-Presentation

Diogenes Armando D. Pascua
School of Computer Studies
Mindanao State University-Iligan Institute of Technology
Iligan City,Philippines
diogpascua@must.edu.ph

Sherwin A. Guirnaldo
Dept. of Mechanical Engg., College of Engineering
Mindanao State University
Marawi City,Philippines
sheronell@yahoo.com

***Abstract*— In this paper a human-sized tracked wheel robot with a large payload capacity for tele-presentation is presented. The robot is equipped with different sensors for obstacle avoidance and localization. A high definition web camera installed atop a pan and tilt assembly was in place as a remote environment feedback for users. An LCD monitor provides the visual display of the operator in the remote environment using the standard Skype teleconferencing software. Remote control was done via the internet through the free Teamviewer VNC remote desktop software. Moreover, this paper presents the design details, fabrication and evaluation of individual components. Core mobile robot movement and navigational controls were developed and tested. The effectiveness of the mobile robot as a test bed for tele-presentation were evaluated and analyzed by way of its real time response and time delay effects of the network.**

***Index Terms*—— telepresence, teleoperation, remote sensing, Skype, Teamviewer.**

I. INTRODUCTION

A.BACKGROUND

Telepresence has been the hype of the modern day fictional films like the *Surrogate*, *Avatar*, *Sleep Dealer* and *Gamer.* In Surrogate for example, a person will be assigned to a robot replica of oneself which one has to control via his mind in a recliner seat at home. The much younger, stronger, better looking replica will then do your biddings in the world. The real technology behind such fantastical fiction is grounded both in far-out research and practical robotics. In the present day real world telerobots is presently limited to physical interfaces-through wireless internet connections, video cameras, joysticks, and sometimes audio. Humans move robots around at the office, in the operating room, underwater, on the battlefield, and on Mars. Examples of advances in today's telepresence robots are the RP-7 of InTouch Health Systems and QB of Blackwell's Anybots [5]. RP-7 is a mobile robotic platform that enables the physician to be remotely present. Through the integration of key technologies, RP-7 can remove time and distance barriers and effectively extend the physician's reach to manage patient care. The Robot's visualization system consists of a camera, microphone, and a speaker. Mobility and navigation is possible via a holonomic drive system and an array of infrared sensors. Physicians teleoperate the robots through the internet by seating in a control station-a computer with a joystick and a webcam. The wide angle display of the robot cameras presents the doctor with the view of the robot's environment for navigation and to examine patients and converse with the hospital staff. Blackwell's QB on the other hand is a general purpose remote telepresence robot. It is a Wi-Fi enabled, vaguely body-shaped wheeled robot with an ET-looking head that has cameras for eyes and a display in its chest that shows an image of the person it's standing in for. You can slap on virtual-reality goggles, sensor gloves, and a backpack of electronics to link to it over the Internet for an immersive telepresence experience. Or you can just connect to the robot through your laptop's browser.

Telerobots, teleoperators, and remotely operated vehicles belong to a class of machines used to accomplish a task remotely, without the need for human presence on site. They are typically used in situations that are too hazardous to human health or survival, like deep water, outer space, or toxic environments. A growing number of telerobots is used for applications where it would be too expensive or too time-consuming to send humans, for example in telemedicine or tele maintenance that requires highly trained individuals with special skills. Sheridan [11] defines a telerobot as a machine with sensors of the environment and devices to perform mechanical work. The human operator supervises the telerobot through a computer intermediary. The operator communicates to computer information about goals, plans, and orders relative to a remote task, getting back integrated information about accomplishments, difficulties and sensory data. The telerobot executes a task based on the information received from the human operator plus its own artificial sensing and intelligence.

The cost of a telerobotic system can be considerably reduced by using personal computers and prevailing standard software for most of the computing tasks[1]. Another way to reduce costs for some applications is to use the free Internet for communication between the computer that the operator interacts with and the computer that controls the robot. Telepresence robots on the other hand are specialized types of networked telerobots that offer the operator some form of both visual and tactile feedback giving him a sense of as if he is at the actual site of the robot.

Compared to plain robotic systems, in which a robot executes a motion or other program without further consultation of a user or operator, telerobotic systems provide information to and require commands from the user. Their control architectures can be described by the style and level of autonomy.

Using standard internet technology for telerobotic applications offer the advantage of low cost deployment. There is no longer a requirement for expensive purpose built equipment at each operator's location. Almost every computer connected to the internet can be used to control a teleoperable device. The downside is the limitation of the varying bandwidth and the time delays.

The internet offers the infrastructure for communication but still the operator requires software that displays the user interface and communicates with the telerobot over the internet. Powerful browsers are freely available and often updated with increased functionality. User interfaces can be developed using a web browser only can be achieved. Operator visual feedback can also be realized using readily and freely available Skype software, freeing the developer from developing its own visual feedback.

Existing commercial telepresence robots are way far expensive ranging from 6000 to 15000 dollars. Some have their own dedicated web servers to manage control, and visual feedback commands thus adding to the cost overhead by charging users a monthly subscription fee[5]. This research was conceptualized bearing in mind the cost savings when an elementary telepresence robot will be developed from off the shelf components and doing away with a dedicated web server.

Having a dedicated web control infrastructure however leads to a much effective telepresence robot by ensuring an almost zero communication downtime, thus a significant reduction in delays, by efficiently rerouting IP packets to other routes. Since the robot to be designed do away with this type of infrastructure, a controller/robot communication system must be developed with the ultimate goal of minimizing the effects of transmission delays.

Telepresence robots are useful in various areas such as remote presentation, teleconferencing, telemedicine: remote diagnosis, remote treatment of patients with the aid of a medical staff and remote consultation, military applications, remote surveillance, teleoperation , advertising and remote education Due to these benefits, a low cost telepresence robot from off the shelf components was developed. A heightened perception of the presence of the speaker in the remote area was sought and for safety purposes, a control scheme with minimal delay effects was developed.

Most of the existing telepresence robots are either private projects or commercial ones. Each of the robots have their own advantages and disadvantages and this study would like to fill in the gaps left by these telepresence robots. In this study, a robot was developed that somewhat mimics the properties of these established telepresence robots but has some inherent characteristics that somehow complements the shortcomings of these previously reviewed robots. First of all a robot made up of locally available materials was built and thus leads to a low cost platform. The blueprint for the robot was laid out as simple as possible so that a generalized model can be easily duplicated. The design of the hardware was open sourced and the sources of the components were easily obtained locally. The electronic controls components of the robot come in kits that can be obtained in local stores. So in general, a telepresence robot that can be built using modular components was developed. In terms of robot control software, most of the previously reviewed robots have their own proprietary software and most of them are closed source. In this particular telepresence robot, it was decided to use free and open source software. This decision was based on the notion that open source and free software can contribute to the robot's low price and a generalized robot can be synthesized by just anyone with the right means. The previously reviewed telepresence robot are considered complex in terms of control and software and the designed robot however was not. Moreover, the teleconferencing component of the robot came from the freely available Skype software. Overall, a telepresence robot designed with modular components and free software was developed. This means that anyone with sufficient knowledge in electronics and computer programming can develop their own telepresence robot since the components can be obtained readily.

To solve the problem stated thus we have done the following:

1. Designed and constructed a web controlled telepresence robot platform with the following features
    a. A tracked wheel differential drive motion capable robot with motion control system incorporating two sets of optical wheel encoders for dead reckoning linear displacement measurements and speed control using pulse width modulation.
    b. Incorporated a webcam, microphone sets and an LCD display for two way audio and video transmission between the robot and remote controller.
    c. Employed three sets of ultrasonic distance sensor for the robots obstacle avoidance system.
    d. Incorporated on the robot a digital compass for direction sensing based on the 4 cardinal directions.
    e. Incorporated on the robot a 3 axis accelerometer for inclination measurement on the three cartesian axis.
    f. Use of the Skype teleconference software for the audio and video information transmission between the robot and the remote controller.
    g. Low cost and sourced from Commercial off the Shelf (COTS) materials.
2. Developed a purely local robot control using a processing GUI whose control is ported to the operator via TEAMVIEWER VNC.
3. Tested the effectiveness of the robot as a telepresence agent based on the evaluation of the time delay between the transmission and execution of control commands.
4. Developed and optimized control algorithms for the robots navigational control.

II.SYSTEM DESIGN

To reduce complexity, the telepresence robot was a passive terminal with minimal autonomy. A little intelligence however was incorporated through the sensors in a way wherein the robot can have control in case sensors detect obstacles. Mobility was limited to positioning controls, although the prototype can be designed for much general and complex teleoperation task, modelling, calibration and testing for the prototype would be done for remote tele-presentation only. The software used for teleconferencing was limited to Skype and robot control was achieved through forwarding the remote desktop to the controlling computers screen using Teamviewer. The Processing programming language was adopted in developing the robot controllers GUI.

The choice of the type of robot was based on some existing commercial telerobot designs. The robot first and foremost must be able to move around in the remote environment thus a suitable mobile platform was designed. Since the robots task was to interact with other people on the remote site as naturally as possible, a human sized robot was designed.

The robot can be remotely controlled by a remote PC. Fig. 1 shows the overall scheme of the system. Basically, the system consist of two computers, one top of the robot and other will be the remote controller station. These PC's were both connected to the net and communicates with each other via Skype and Team viewer VNC package.

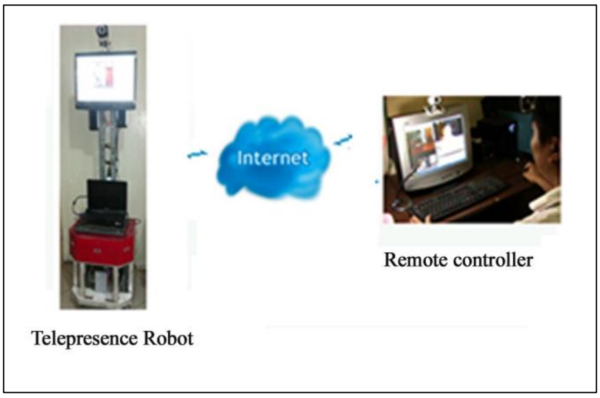

Figure 1.The telerobotic system showing the telepresence robot and its remote controller station

Video, audio data were relayed from the remote PC to the controlling PC in the robot and vice versa. In this setup, the computer in the remote robot runs a java controller program whose inputs were controller commands and sensor data. The controller commands were implemented by normal inputs consisting of buttons, sliders and textbox in a Graphical User Interface (GUI). Outputs were implemented as text labels and graphical elements like gauges and arrows. This GUI were virtually transported on the remote controller station, appearing on the controllers monitor and thus controlled remotely. Fig. 2 shows the Graphical User Interface (GUI) in the remote controllers end.

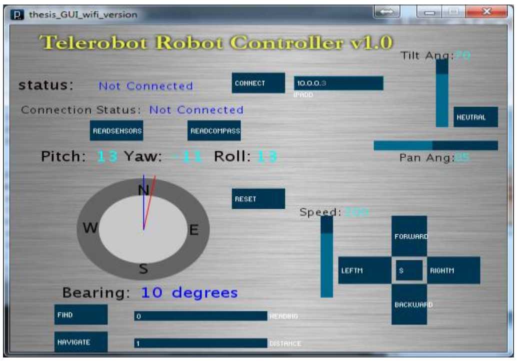

Figure 2.The robot Graphical User Interface

A. Hardware Design

The hardware of the robot platform is shown in Fig. 3 and Fig. 4. The remote robots skeletal system was made up of a combination of 1x1 square aluminum bars comprising the robot frame and a main vertical post made up of a 15/8 by 1 5/8 14 gauge slotted angle bar. The post acts as a backbone to support the visual feedback system consisting of the 14 inch LCD monitor and the camera system. Nuts and bolts of various sizes were utilized as fasteners. Acrylic plastic sheets with thickness of 2.5 mm and 3 mm was used as coverings. A tracked industrial platform acts as the robot base. The main controller of the robot was a microcomputer with an Intel-based processor. This computer processes the video feed from the robot mounted camera, processes the audio feed from the microphone , outputs the received video information from the controlling PC to a robot mounted LCD, outputs the received audio signal from the controlling PC to the robot speakers, control and monitor the communication link between the controller and the robot, and perform motion and tactile commands to the robot as well as process the robots sensors relaying feedback signals to the controlling PC. The robot moves via a differentially driven track wheels which are in turn

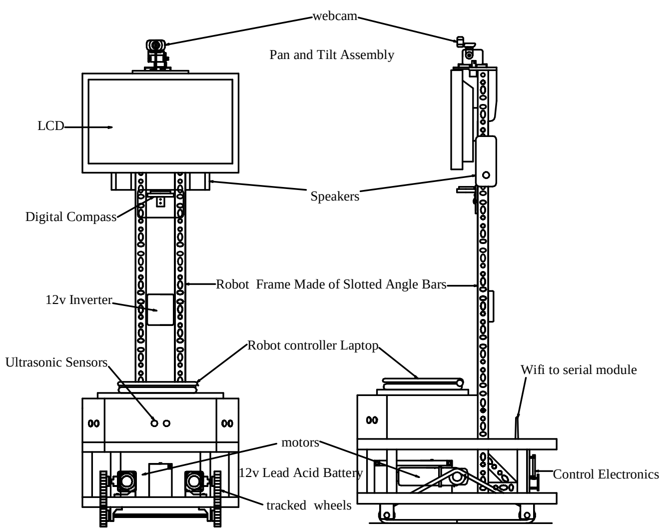

Figure 3.The remote Robot System

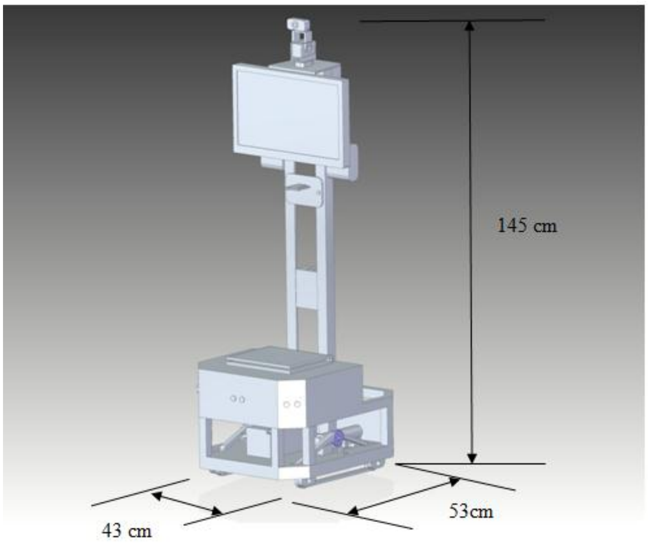

Figure 4. The remote robot system dimensions

controlled by an arduino microcontroller via motor controller boards. Slave microcontrollers were employed to process sensor data and a pan and tilt mechanism for the web camera system. Robot navigation and tactile commands were processed by the main microcontroller. Obstacle avoidance was achieved by the use of three ultrasonic sensor arrays spaced symmetrically around the body of the robot. Connectivity on the robot site was provided through 3G internet service via a 3G wireless router installed in the remote site. The overall block diagram of the system is described in Fig 5.

Two Nissan mt3-12 high torque DC geared motors were employed to drive two threaded drive chains. The drive chains are configured for a differential drive robot base. The two motors were controlled by two 6.0 Ampere H-bridge motor driver kits from E-gizmo Mechatronix Central. These kits control the direction and speed of both motors. The axel connected to the shaft of the motor was coupled to an optical encoder wheel. Two sets of optical incremental wheel encoder system were developed for the two motors using infrared reflective sensor combined with the wheel encoder mounted on the motor shaft. These absolute encoders determine the relative distance travelled by the robot at a certain time difference. A Phillips KMZ52 based electronic compass acts as a sensor for azimuth position. The output of this sensor was used as an input to the motor controller system.

A proportional integral control scheme was employed for the mobile robot to control its displacement and heading. The output signal of the PI controller will be a PWM signal which will be the input to the motor controllers. Two sets of PWM controllers were employed, one for the left motor and one for the right motor. Fig.6 shows the PI controller implementations for the robot.

The controller station consists of a laptop PC equipped with a web cam, a microphone and speakers. Broadband Internet access must be provided for this PC using any of the existing services in the Philippines like DSL, 3G, 4G or Wimax .This controller PC must have Skype, Teamviewer , Java Runtime installed. The hardware of the controller station is shown in Fig.7.

The controller PC will be the one responsible controlling the remote PC mounted on the robot. The two computers must communicate via the internet using Skype and The VNC software Teamviewer.

B. System Software Design

The video and audio transmission between the robot and the controller was handled by the Skype Software. Both the Robot PC and The Controller PC have Skype installed, each with their corresponding Skype accounts registered. Banking on the popularity of this software, it was assumed that this part of the design was already been taken cared off

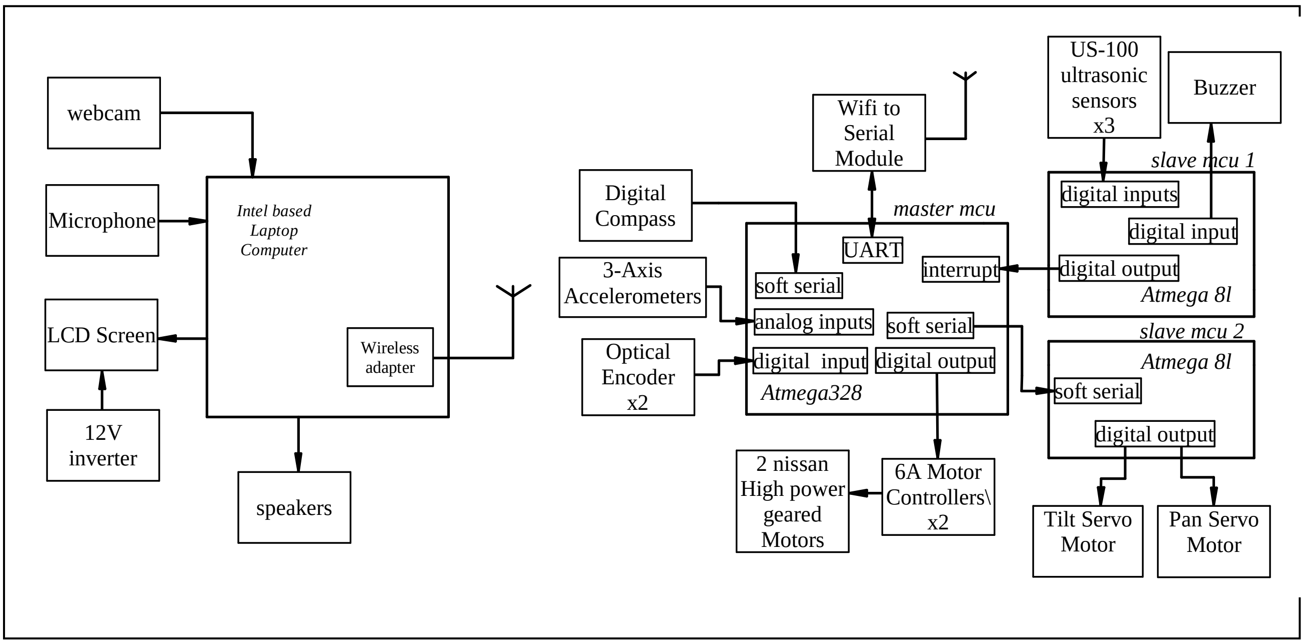

Figure 5. Overall block diagram of remote robot

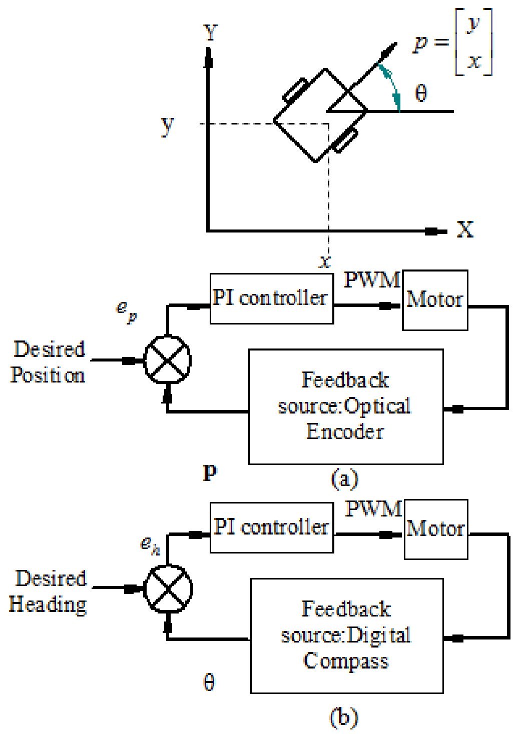

Figure 6. Two PI controller implementation. (a) for position (b) for angular heading

Figure 7. Controller hardware setup

since the software has already been proven in the internet in terms of affectivity and efficiency. Albeit to say, the system therefore was highly dependent on Skype in terms of reliability of connection and transmission delays of the video and audio information.

On the remote robot computer, a Graphical User Interface was implemented using the java based Processing language. This interface was actually a TCP client which connects with the wifi server module of the robot. These client sends control signals to the server and at the same time receives feedback data from the server. This GUI was then virtually transported on the controller monitor through the Teamviewer software. That is, the remote controller screen has a display of the remote robots PC desktop.Through Teamviewer, the remote controller PC can control the GUI on the robot PC. The effectivity of sensing and executing control commands via this setup was dependent on the transmission

Figure 8. The overall robot system hardware: (a)overall robot system;(b)LCD video system and speakers;(c)Web camera mounted on pan and tilt servo mechanism; (d)Control electronics consisting of microcontrollers, motor controllers and power conditioning circuits.

delay on the internet and it was investigated through a series of experiments.

The Graphical User interface permits the control of the movement of the robot. Two methods of robot control movements were allowed, manual direct control and semiautomatic. In the manual control, 5 directional buttons corresponding to forward, left, right, backward, stop were used. Clicking on the desired a particular button effects the desired movement of the robot . Eight bit values ranging from 0 to 255 can be used to implement PWM speed values to the motors. While the robot is executing a particular robot movement, the clicking of another movement button will stop the robot. One has to click the button again to effect the desired movement. Once the robot is moving, the ultrasonic sensors are active. When an obstacle is directly in front of any of the three ultrasonic sensors, the robot will stop its movement and sends the “obstacle detected” robot status on the robot GUI. One can in anytime stop the movement of the robot by sending any commands to the robots GUI. In the manual method, the robot implements the command and wait strategy wherein the robot executes the sent command and then waits for the next command. Some commands have a definite duration in terms of execution and once the control routine is finished, the robot stops and waits for the next command. Examples for this are the pan and tilt commands for the camera.

In the semiautomatic mode of control, sliders and buttons are implemented. Two possible movements are implemented in this method:

1. distance traversal
2. angular heading seeking

In the distance traversal, the user uses the distance slider to enter the required distance of travel. One then clicks the navigate button and the robot moves forward to the required travel distance. This movement is implemented using a Proportional Integral controller algorithm where the inputs are the actual travel distance and desired travel distance. The output of the controller will be the PWM signals on the two motors driving the track wheels. The actual travelled distance was measured using an optical encoder.

In the angular heading seeking movement, one uses the heading slider to choose the desired heading. The heading angle correspond to the earth magnetic pole directions with 0 degrees as north, 90 degrees as east, 180 degrees as south and 270 degrees as west. To lock to the desired heading, one clicks the FIND button and the robot turns towards the desired angle. The robot turns towards the desired angle using a PI controller where the inputs are the desired angle and the actual heading angle. The output of this controller will then be the PWM signals that drives the robot motors. The implementation of the PI controller in this system is complemented with a logical system wherein the choice of turning direction is dependent on the amount of angle to be traversed. The system is programmed wherein the turning angle to be traversed will be the one which entails lesser turning distance. For example if the present heading angle is 90 degrees and the desired angle is 180 degrees, there are two possible scenarios for implementing this heading seeking:

1. Clockwise at an angular displacement of 90 degrees
2. Counterclockwise at angular displacement of 270 degrees (360-90).

For this example, the robot selects the clockwise movement for it entails lesser angular displacement to be covered.

The actual heading angle was measured by a digital compass sensor whose output is a serial data sent to the robot controller. The actual desired angular headings of the robot are displayed on the GUI using a circular gauge with two dials. The red dial is for the actual heading and the blue dial is the desired heading. Whenever the robot moves, this actual heading dials location is updated. One can also manually update the actual angular position by clicking the READCOMPASS button. The Robots Pan and Tilt Camera can be controlled via the Tilt Angle and Pan Angle sliders in the GUI. The Pan angle slider value corresponds to the amount of angular degrees the camera pans. These angular values can range from 0 to 180 degrees. The tilt angle slider values correspond to the amount of degrees the camera tilts. These angular values can range from 0 to 180 degrees. By clicking the neutral button, one can reset the camera to point to a default viewing position which is the usual orientation of a webcam. Neutral position corresponds to a pan angle of 85 degrees and a tilt angle of 90 degrees. An overall GUI for the robot controller was employed on the robot. This GUI serves as a frontend to a telnet session between the robot PC and The Wi-Fi to serial converter server. The Wi-Fi to serial converter

There were three arduino microcontroller board utilized in the implementation of the control of the robot. One Atmega 328 based master microcontroller board acts as the master controller and two Atmega8l based microcontroller board acts as slaves. One of the slave module acts as the controller for the pan and tilt servo motor assembly. This slave microcontroller waits for serial command coming from the master microcontroller. These commands are the pan and tilt angle for the servos. The other slave microcontroller controls the three US -100 ultrasonic sensors constantly getting its distance reading. When the distance of the obstacle on any of the three sensors was below the threshold level, this microcontroller sends a low level signal on an output pin. This pin was connected to an interrupt pin of the master microcontroller. The master microcontroller therefore had an interrupt whenever a low level signal was present on this pin.

The master microcontroller performs numerous task. These tasks were:

1. Communication with the WIFI serial module.
2. Controlling the two 6 A h-bridge motor controller.
3. Receives encoder counts from the two optical encoders.
4. Receives serial data from the digital compass.
5. Reads the output of the three axis accelerometer.
6. Executes the overall control loop for the control of the robot.

Fig. 9 shows the overall flow chart for the main controller loop of the master microcontroller firmware.

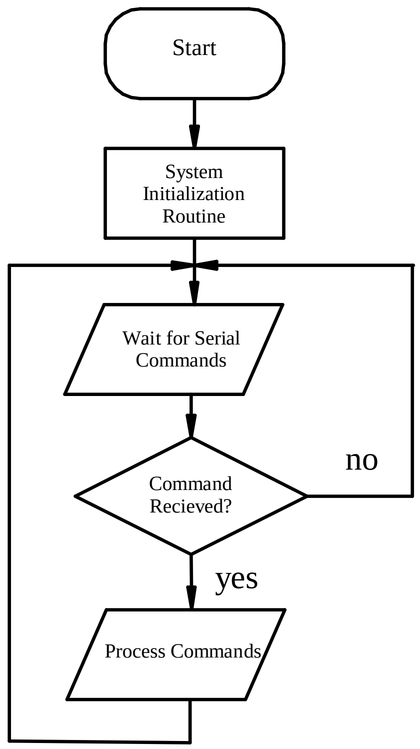

Figure 9. Flowchart for the main controller loop

The initial step in the main microcontroller code was the initialization routine. Here every piece of hardware interface was initialized, I/O pins data direction are set depending on the peripheral connected to it. Figure 10. is the flowchart for the initialization routine. The initialization is implemented on the setup( ) function of the main microcontroller code.

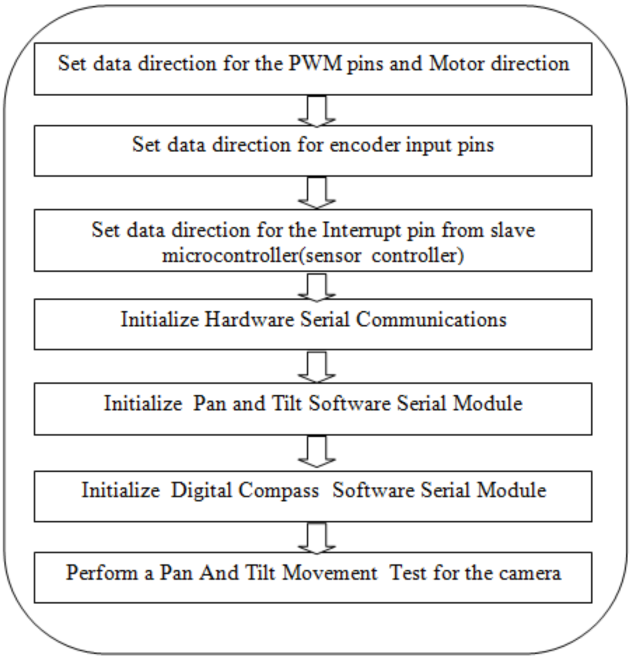

Figure 10. . Flowchart for the initialization routine

The bulk of the main loop of the master controller's firmware was a blocking wait loop. The program waits for serial data coming from the wifi serial module. The data coming from the wifi module were actually commands coming from the main graphical user interface. As mentioned previously, the graphical user interface processes commands entered by the user as relayed through the internet via Teamviewer. The commands received by the GUI were then encapsulated into a TCP packet which was then relayed to the serial wifi module. The serial wifi module then decapsulates the commands from the TCP packet into a stream of serial data. These serial data will now be the serial commands the main controller loop processes. Commands are formatted as a series of 4 bytes The wait loop in the main loop waits for this and upon reception, the first byte then was used as an argument in a switch case block in the code. Different commands then are executed based on the decoded first byte. Likewise if a particular command needs a response from the robot, the said response was sent to the wifi serial module as a serial character stream. This serial stream was then converted to a TCP packet which will then be transmitted to the GUI client on the controlling PC. An example of this was the string "ready" which was transmitted every time a command was successfully executed by the robot.

III.TESTING AND EVALUATION

The main goal of this study was to develop a basic telepresence robot whose affectivity was dependent on how accurately it performs the tasks assigned by the controller. The study implements the objective method of evaluating the performance of the telerobot through a series of tasks that it has to complete at the shortest time possible, thus *task completion time and reaction time* to a remote stimuli were the metrics used. Task completion time was measured and averaged while the reaction time was measured by getting the time difference between the completion time in the local station and the time of completion as perceived in the remote station. Two tasks have been designed, one is the traversal of a square path and the other one is the traversal of a straight path.

The traversal of the square path employs the shared continuous means control. In this method a "command and wait strategy" was implemented by which the robot was sent commands one at a time. One command must be finished first before another command was sent. The robot on the other hand monitors obstacles along the path, and once an obstacle was detected, it stops the present maneuver of the robot. The next step for the robot was then to wait for the next command and execute it. The path was traversed 10 times and the time of completion was recorded through *Wireshark*, an open source network sniffing software. The perceived time of completion was also measured on the controlling computer through *Wireshark* by looking at the timestamp on the TCP transmission of the last command coming from the remote machine. The testing area would be a tiled surface with an outline drawing of the path to be traversed. Two paths have been defined for testing, one is a square path with a side dimension of 1.2 m and the other is a straight line with a length of 1.6m.

In the experiment, the robot performs its usual startup routine whereby it connects to the controller station via internet. Skype and Teamviewer were run on both the robot and the controlling station. The robots controller GUI was then forwarded to the controller screen so that it can remotely control the robot. Skype provides the visual feedback to the controller PC so that it can effectively maneuver the robot. Once connection was established, the robot then traverses the paths. While the robot was traversing the paths, Wireshark monitors the wireless data packets leaving and entering the robot PC. These data packets are then sent remotely to the controller stations PC for monitoring. Wireshark in the controller station also monitor its wireless data packets particularly the timestamps on each TCP packet. Through the timestamps in the TCP packets, the time of completion and time delays was calculated and recorded. In calculating the time delay, the tasks starting time and ending time were extracted from the TCP stream trace and packet display list of the TCP transmission between the robot PC and the wifi module. Fig. 11. shows software modules and the data flow during the experiments.

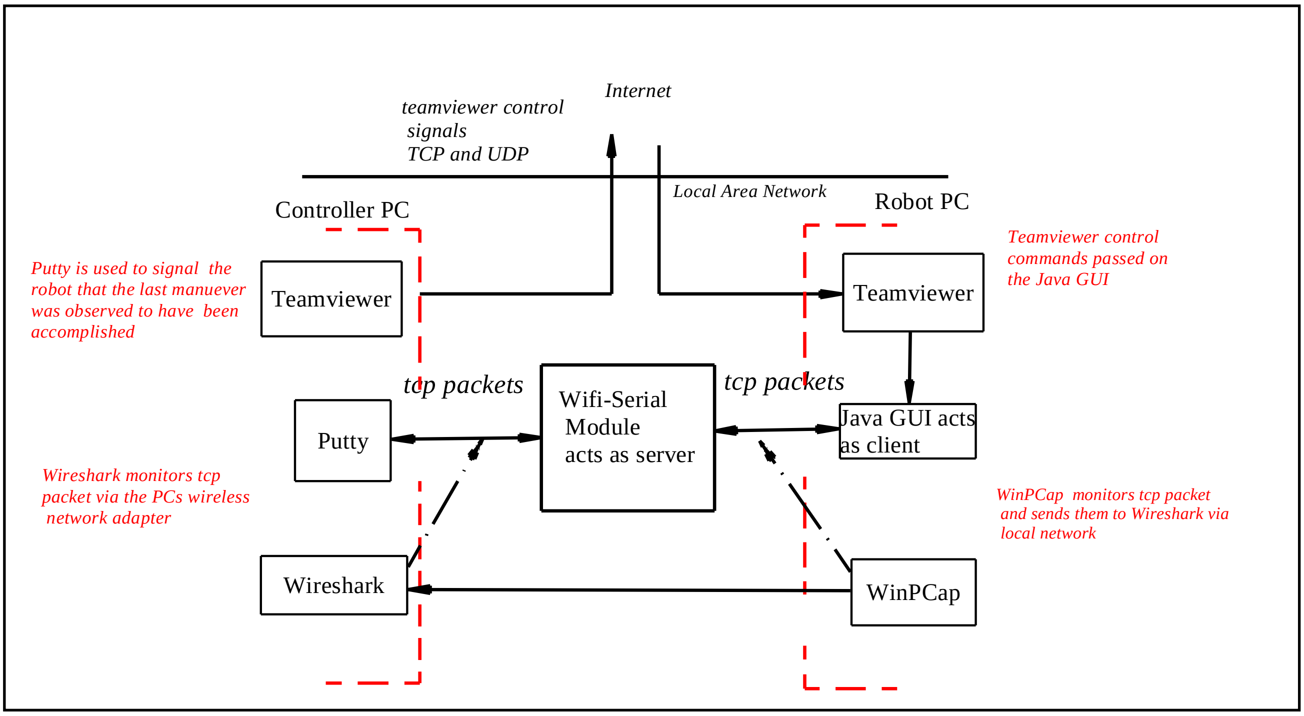

Figure 11. Software modules and the data flow during the experiments

IV. Conclusions and Recommendations

The goal of the study was to develop a human sized tracked mobile robot suitable for telepresence applications. A mobile robot equipped with sensors and tele-presentation capabilities were developed. The robot was built from off the shelf components so that the overall cost of the system is as minimal as possible. The robot design process was implemented through the manufacture and testing of individual components. Core mobile robot movement and navigational programs were developed and tested. The effectiveness of the mobile robot as a test bed for tele-presentation were evaluated and analyzed by way of its real time response and time delay effects of the network. Iterative process of software creation, testing and debugging were done to come up with an optimized code suitable for both mobility and remote robot control. The physical hardware components of the robot were developed first through an iterative process of mixing and matching. Once the hardware components were developed, an iterative process of software development was implemented in order to fit the desired function of the robot.

The mobility of the robot has been proven effective through a series of mobility test scenario where the robot has been controlled. The use of Skype has also been proven effective in sending the audio and video information between the robot and its controller. The use of Teamviewer as the controller medium for control signals deemed effective provided a good quality of service is expected from the internet provider. The use of Wireshark deemed effective in capturing the time of execution of task as well as time delays with accurate resolutions of up to 10 ms. A test was conducted which resulted to an average time delay of execution of less than 2 ms in both tests. This test was done not taking into account the quality of service for the internet connection. Overall, the design of the robot leads to a low cost system; as such materials were coming from off the shelf items.

A Series of time of task completion test and time of execution delay was conducted through the use of Wireshark network sniffing software. The researcher suggests therefore that more test be done on this aspects taking into account the different quality of service at different times of the day. As for the robot control, the researcher recommends further investigation by implementing control software that is not dependent on the commercial Teamviewer software. A pure client and server program for robot control is therefore recommended for further study.

As for mobile robot mobility, wheel slippage has been observed due to the nature of the wheels employed on the project. This resulted to severe errors on the optical encoder system which limits its effectivity on smaller distances. Due to this, the researchers recommend the use of other means of robot locomotion such as wheels. Odometry using optical encoders deemed erroneous when large distances was traversed, thus the use of MicroElectroMechanical Modules (MEMS) such as accelerometers as position sensors are recommended.

REFERENCES

[1] R. Aracil, M. Buss, and S. Cobs, “The human role in telerobotics”. Springer Tracts In Advanced Telerobotics, 2007, vol 31, pp. 11-22.

[2] J. Borestine, and H.R. Everet, Where am I, Sensors and Methods for Robot Positioning. Michigan:UM Press, 1996.

[3] T. Bräunl, Embedded Robotics. Berlin:Spinger Science, 2006.

[4] M. Buss, and G. Schmidt, "Control problems in multimodal telepresence systems," Advanced Control Highlights Europe Control Conference, 1991, pp. 65–101.

[5]. Caroll, T. "Then and now:telepresence," Servo, Vol 9, pp. 76-80, March 2011.

[6] J. FENG, Introductory Robotics. NewYork:Prentice Hall, 1194.

[7] D. Geer, "Remote presence robots put physicians within reach," Servo,vol 8, pp. 76-88,June 2009.

[8]. G. McComb,Robot builders bonanza. New York:Mc Graw Hill, 2002.

[9]. G. Niemeyer, C. Preusche, and G. Hirzinger, "Telerobotics," Springer Handbook of Robotics, 2008, pp.741-754.

[10] W. Schloerb,Handbook of Robotics.New York:Prentice Hall, 2002.

[11] T.B. Sheridan, Telerobotics, Automation and Human Supervisory Control. New York: The MIT Press,1992.

[12]. R. Siegwart, and I.R. Nourbakhsh, Introduction to Autonomous Mobile Robots. London : MIT Press,2007.

[13]. D. Song, K. Goldberg, and N.K.Chong, "Networked Telerobotics," Springer Handbook of Robotics, 2008,pp. 759-769.