REMOTE SURVEILLANCE VIA WIRELESS-CONTROLLED MOBILE ROBOTS

TAREK M. SOBH, UNIV. OF BRIDGEPORT, USA, SOBH@BRIDGEPORT.EDU
RAJEEV SANYAL, UNIV. OF BRIDGEPORT, USA, RAJEEVS@BRIDGEPORT.EDU
BEI WANG, UNIV. OF BRIDGEPORT, USA, BEIWANG@BRIDGEPORT.EDU

ABSTRACT

This work addresses visual remote surveillance through the World Wide Web. A mobile robot built at the RISC lab is controlled via the Internet with the help of images obtained from a network camera. The user specifies the desired position by utilizing the real time web based visual interface. The autonomous robot moves to that location avoiding obstacles.

KEYWORDS: Autonomous mobile robot, web-based, visual surveillance.

1. INTRODUCTION

The RISC team at University of Bridgeport has developed a web based wireless robot for remote applications. The mobile robot "UBROBO" allows remote control over the Internet using a Java-enabled web browser. A network camera installed in the lab provides real time feedback to the user regarding the robot's position.

A variety of web based remote control interfaces for robots has been developed over the last few years. Xavier [1] can be advised by Web users to move to an office and to tell a joke. Rhino [3] and Minerva [2] were deployed as interactive museum tour guide and could be operated over the web. The robots mentioned above operate in known environments with pre-stored map during their navigation. UBROBO can operate in unknown environments provided a camera is installed in the area of operation.

2. SYSTEM ARCHITECTURE

The server hardware in the system architecture shown in figure 1 is a Pentium based PC with Windows 2000 Professional running the web server with JSP support. It is composed of three major modes: a web interface, a wireless interface and a remote robot (UBROBO) (figure 2). A user logged on to the web server can observe the robot's current location with the images obtained by a network camera installed in the lab. The network camera (AXIS) is installed on the ceiling of the robotics lab, facing the floor to provide the user with a top view of the lab. To command the robot, the user clicks on the desired location visible on the image. As soon as the user clicks, the web server first plans the robot path based on the obstacles present. It then transmits the path to be followed to the robot via the wireless link. Once the robot receives the command, it moves to the desired location. If the robot detects any obstacle it signals to the PC. Path is planned by the server till robot reaches the desired position. A camera temporarily installed on the robot transmits the images to the web server.

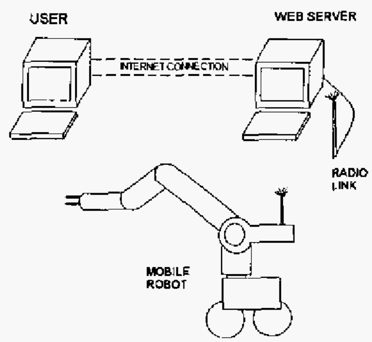

Figure 1: System Configuration

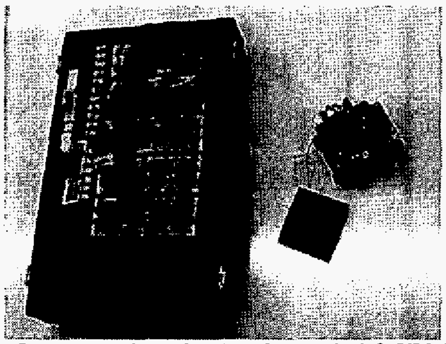

Figure 2: System Overview, with wireless interface on the left, UBROBO on the right

3. UBROBO ARCHITECTURE

3.1 Hardware Structure

The physical structure of UBROBO can be seen in the images shown in figure 3-6. UBROBO is equipped with ATM 103 MCU, 4 sets of infrared sensors and emitters for object detection and avoidance, 2 sets of infrared sensors and emitters for wheel encoding, a PC video camera (LEGO) and a RF transceiver for wireless connectivity with the web server. Differential drive mechanism has been adopted for locomotion. The robot is powered by two 5V DC shunt motors. To maintain balance, pivots are placed on the four corners of the robot. The wheels have horizontal projections at the edge of the rim. These projections are used by an infrared emitter and a sensor for the purpose of encoding. The PC video camera is placed on top of the rectangular frame of the body in the front portion. For the purpose of obstacle avoidance, two sets of infrared emitters and sensors are placed on the front (left and right corner), one set on the left side and one on the right side. The entire unit is powered by a 6V battery.

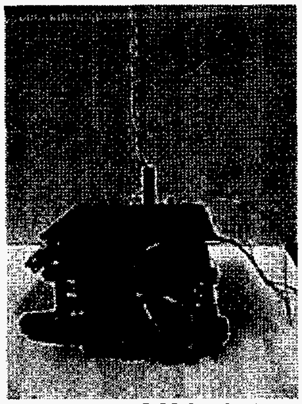

Figure 3: UBROBO side view

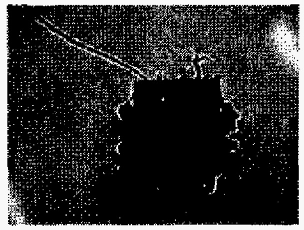

Figure 4: UBROBO top view

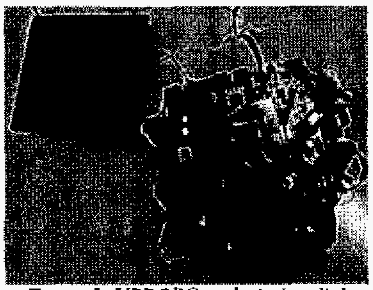

Figure 5: UBROBO and wireless link connectivity

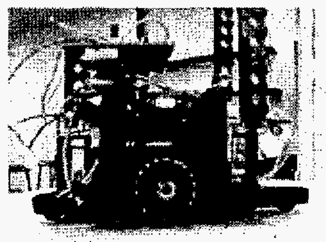

Figure 6: Encoders of the wheel

3.2 Path planning

The path planning of robot is done through the web server. The server processes images obtained from the network camera. After calculation, it commands the robot proper actions based on current obstacle status. As soon as the user submits a request, the shortest path is computed based on the robot's present position and goal position. Path is recalculated if sensors on the robot detect an obstacle. At the end, the server checks if the robot has reached the exact position. This is done in order to reduce errors due to odometry.

3.3 Embedded Software in MCU

ATMEL's ATM 103 is used as the central control unit of UBROBO. The embedded software in the MCU can be divided into the modules shown in Figure 7.

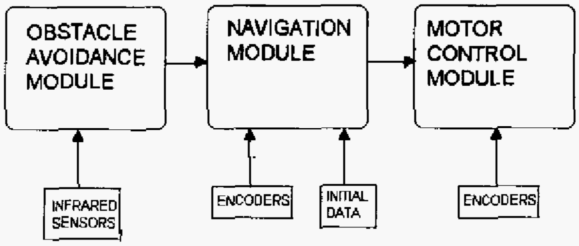

Figure 7: System Architecture of Embedded Software in MCU

3.3.1 Obstacle Avoidance Module

Four infrared emitters and sensors are placed in front side of the robot for avoiding obstacles. The obstacle avoidance module signals to the PC whenever an obstacle is detected.

3.3.2 Navigation Module

The Navigation module operates on data obtained from encoders, obstacle avoidance module, and the desired path sent from the web server. To make inverse kinematics as simple as possible, the robot always tries to maintain a straight-line path. The path coordinates are converted into the following parameters:

1) Angle: the angle that the robot has to rotate
2) Distance: the direct distance between the initial position and final position

Navigation is carried out in a straight-line path until an obstacle is detected.

3.3.3 Motor Control Module

To maintain a straight path, both wheels have to be driven at the same speed. A PID algorithm is used to maintain straight line path. The PID algorithm operates on the difference of data between the two wheel encoders.

4. WIRELESS LINK INTERFACE

Radio communication is based on Linx RF receiver module RXM 433 LCS installed on the robot (figure 8, 9) and transmitter module TXM 433 LCR installed on a breadboard (figure 10, 11). It is interfaced with the MAX 232 level converter. The data is transmitted serially from the PC to the robot at a baud rate of 2400 bps.

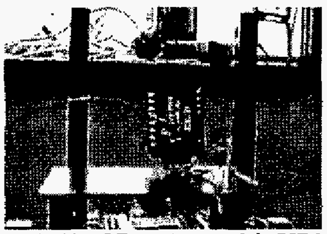

Figure 8: Linx RF receiver module RXM 433 LCS installed on the robot

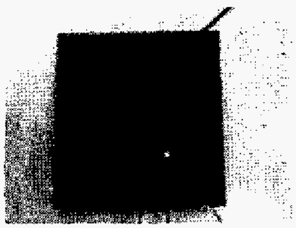

Figure 9: RF receiver top view

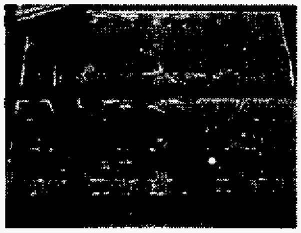

Figure 10: Interface overview between Max232 and TXM433 LCR based on breadboard

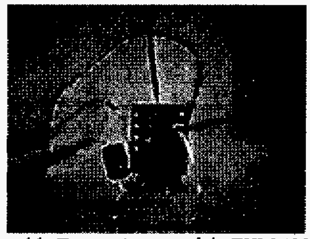

Figure 11: Transmitter module TXM 433 LCR installed on a breadboard

5. WEB INTERFACE

The web interface has been developed with Java Server Page Technology, Servlet and Java(tm) Communications API. Java Communications API contains support for the RS232 serial ports and performs asynchronous I/O with its updated functionality. The web interface is the frond end that interacts with the user, and enables the user to control the wireless robot. Through Servlet, the Java Communications API communicates with the serial port, which connects with the Wireless Link Interface, as shown in Figure 12.

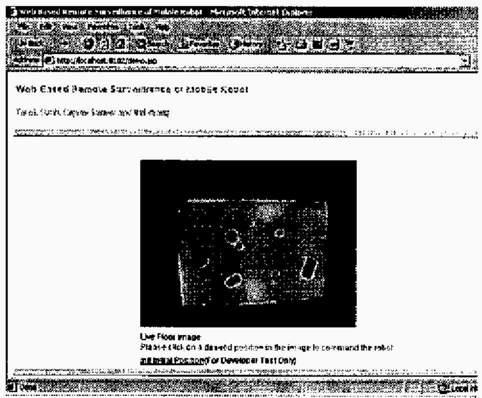

Figure 12: Web interface to command robot. Robot is shown avoiding obstacles along the path.

6. RESULTS

This section includes results from a robot web control sample. After logs on to our web server, the user clicks on the image with the robot and watch the robot moves to the desired clicking point with live video. Updates on the web services and server availability information will be posted under www.bridgeport.edu/~beiwang/paper2/. A new version of our robot is under development currently. There are several video clips available on the website which shows the robot in action, while avoiding obstacles along the path. There is also a video clip available which shows the actual control process on the web interface.

Positional errors have been observed during the operation of the robot. These are primarily due to the low pulse rate of the encoders (32 per revolution). To minimize the positional errors being accumulated, the robot has to be manually reset to the default position and orientation after

10 operations. Positional errors are also due to jerky motion during running. By reducing the running speed of the motors, these errors can be minimized.

7. FUTURE WORK

Future work to be performed on the robot includes:

1) Adding a suitable RF transceiver for the LEGO CAM.
2) Installing better encoders to reduce positional errors.
3) Adding a module on the embedded software in the MCU, this will manage smooth starting and stopping of the motors.
4) Enabling multiple web users.

8. CONLUSION

We have developed an experimental architecture for web-based control of an autonomous robot. The autonomous robot provides further proof that "autonomy mitigates the effects of low bandwidth and unreliable communication" [1]. We conclude that the web interface has made the human interaction with the robot more convenient and flexible. We see a significant potential in expanding this research to similar remote control applications.

9. REFERENCES

[1] R. Simmons, J. Fernandez, R. Goodwin, S. Koenig and J. O'Sullivan "Xavier: An Autonomous Mobile Robot on the Web," *IEEE Robotics and Automation Magazine*, 1999.

[2] S. Thrun, M. Bennewitz, M. Burgard, F. Dellaert, D. Fox, D. Haehnel, C. Rosenberg, N. Roy, J. Schulte and D. Schulz "MINERVA: A Second generation Mobile Tour guide Robot," *Proc. of the IEEE International Conference on Robotics and Automation (ICRA'99)*, 1999.

[3] W. Burgard, A. B. Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Schulz, W. Steiner, S. Thrun, "Experiences with an interactive museum tour guide robot, " *Artificial Intelligence*, 114(1-2), 2000, pp. 1-53.