

Robots Mobile Interactive Control Model For Teleoperation

Mohamed Emharraf*, Mohammed Rahmoun,
Mohammed Saber

Lab.Electronic and Telecommunications, ENSAO
Mohammed First University
Oujda, Morocco

Mostafa Azizi
Lab.MATS,ESTO
Mohammed First University
Oujda, Morocco

Abstract— Nowadays, remote control of robots (telerobots) is mainly based on two models; the first one is used for controlling robot that work in well-defined environments; The second model concerns the monitoring and surveillance robots acting in an unknown and dynamic workspace. The disadvantages of these models are the lack in real human control of the robot tasks, and the limited available functionally interfaces. In this paper, we propose an implementation of an interactive control model; this model allows the robot not only to perform the operator commands, but also to run local functions, such as dealing with unexpected events. We develop a specific interactive control with two control interfaces respectively for P2P and HTTP protocols. The experimental tests that we performed to evaluate the reliability and effectiveness of the interactive control model are promising.

Keywords: *telerobot, remote control, interactive control, communication protocols.*

I. INTRODUCTION

Currently, robots are not only able to perform basic motions [12], they are also capable of interacting with human operators [1-11]. Telerobots are present in different areas of remote control with several potential applications such as telemedicine, distance learning, industrial automation and military. The main difficulties and limitations of remote telecontrol include the problems related to the control network such as bandwidth scarcity, transmission delays, as well as lost packets. All these limitations affect remote control telerobotics performances [13]. To find new solutions to these problems, some issues of telerobotics via network have been explored in recent years.

The Mercury project1994 [2] and Australia Tele-robot [1] are pioneer groups in the implementation of remote control of mobile robot (telerobotics). Currently the implementation of telerobotics are developed by several research teams (some of them are available online). The first generation of telerobots was mainly based on simple robot manipulators or mobile robots directly controlled by human operators [1, 2, 5]. These telerobots work in a structured environment with little uncertainty, and have no local intelligence. However, current research focuses on autonomous mobile robots navigating in dynamic and uncertain environments [3,4,20]. This generation of telerobots is founded on autonomous and interactive robots; these robots can navigate in the real world and deal with

uncertainties. The main goal of this generation is the remote monitoring and control [13].

The existing telerobots can be classified into two categories: robot manipulators [1, 2, 24], and mobile robots used for navigation [3-7, 21, 23]. For both categories, there is a multitude of control methods; the robot manipulators are often located in a limited work area and the direct control is the most popular control model. To address the problem of delay in these systems, three approaches are used [14]: the predictive approach, the simulating and planning display approach and the event-based approach [8, 11].

Some robots use direct control [5,7,22] which is not really suitable for performing remote operations by mobile robots. This is due to its high latency and other problems inherent to the use of communication networks, such as scarce bandwidth and the loss of packets.

Most telerobots use the supervisory control [3,4,20] with local intelligence to solve any problem arising from network communications. Unfortunately, most of these systems [3,4] lack interaction between the operator and the robot, this type of model is labeled as a passive surveillance control [15]. Researchers are trying to add more interaction between humans and robots [19,24], such as supervised autonomy, shared control, cooperative and collaborative control. These modes of control are designed by active control of surveillance or interactive control [15], the main drawbacks of these control modes are: (1) they lack a complete architecture to process with orders that can be sent continuously by the operator, (2) it is difficult to evaluate their performance in running mode and provide the appropriate intervention measures. (3) Their components are interdependent, thus, the breakdown of one component can cause multiple failures, (4) the interfaces are not sufficiently clear for human operator.

In this paper, we attempt to solve the problems related to the passive surveillance controls and interactive methods of control, by implementing an interactive control model (telecontrol) for mobile robot teleoperation.

This paper is organized as follows: Section 2 presents the overall framework for telecontrol ; Section 3 discusses our telecontrol model ; in Section 4, we present the obtained results.

II. THE MODEL OF INTERACTIVE TELEOPERATION:

A. The telecontrol system architecture:

For a navigation system guided by an operator, the operator has three options to guide the system. The first method is to guide directly the system step by step. For the second method, the guide just gives notes about the direction (e.g: turn right and go forward 20 meters, then turn right ...). The third method is to use an environment map and a number of geographic points for setting the route on the map. We note BT the basic methods of telecontrol, and AT the advanced telecontrol methods. Figure 1 shows our telecontrol architecture, which is a basic architecture of interactive telecontrol, the operator orders issued by interactive commands with the application running on the telecontrol robot system, the control interfaces allow the operator to view the video stream captured by the robot, the information about

the obstacle within a radius of 2 m and the actual speed of robot.

The Command Processor module (CPM) processes and executes user commands; the transfer sensing module collects data from different robot sensors and prepares this data for processing; The command executor module creates a feedback loop between the control system, actuators, and the results of sensors. This feedback allows the robot to react quickly to unexpected events derived from the dynamic of real world. The update block module is responsible for transforming sensor data to high-level data (eg, the distance between robot and obstacle in a precise angle). The event detected by the robot allows CPM to make deliberation plan and respond to the encountered situation.

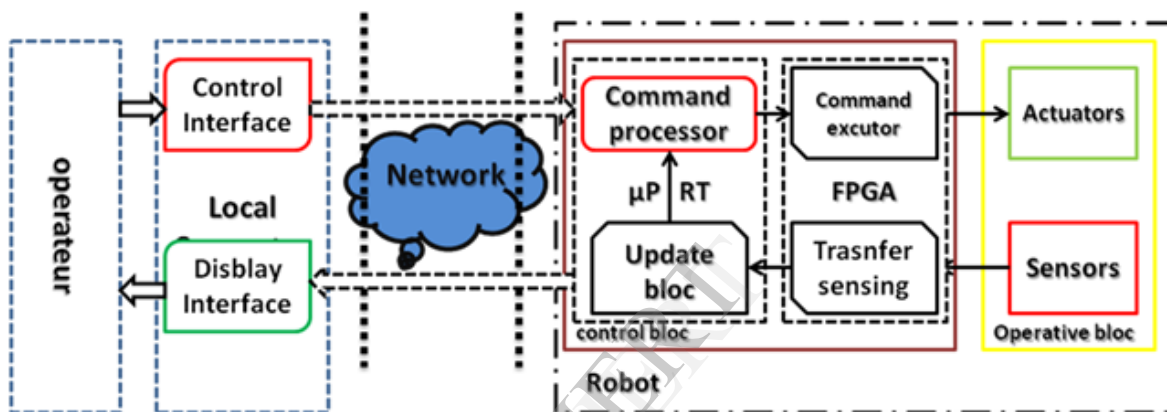


Figure 1. Telecontrol Architecture

B. The telecontrol system:

Our telecontrol system is similar to the direct control used for vehicles driven by a human being. The operator uses the Direction command (change the direction left or right) to turn the robot, Up and down commands to control the speed of robot, to stop or even reverse the direction of robot walking.

In fact, our telecontrol system differs from the way the human operator driving a car. This is a classic example of direct control, where the process has three important characteristics: (1) the driver can receive environmental information in real time, thanks to human vision and using equipment of the car, from this information, it is possible simultaneously to build a model of the real world. (2) the driver can immediately respond to any eventuality and its actions are effective immediately. (3) Most cars lack of autonomous intelligence and rely on the driver for managing the contingencies. This is different from the situation of a human driver, who must constantly make comments on the direction or acceleration. In addition, the robot has a certain level of autonomy to respond quickly to events in such a way that the human operator does not need to deal with command details.

The operator sends commands only if necessary (changing speed, direction, etc.), this greatly reducing the number of operator commands (compared to classic case). The

robot decides autonomously about current situations; for example in the case of danger detection (obstacles), the robot autonomously reduces the speed to a reasonable level and sends a warning message to the operator control panel. If the danger is immediate, the robot goes in standalone mode to move away from danger. In some situations, the robot gets in the autonomous behavior mode that dominates potentially the privilege control and it remains there until it detects that the danger disappeared. Here, the robot can't respond any operator command.

The UP and DOWN commands affect the robot speed ; the direction control affects the steering angle of the robot, ; we define the UP function in a way to allow the robot autonomously reducing its speed to a reasonable value, in the case of rotation or obstacle detection.

III. EXPERIMENTATION :

Our research is realized on a unicycle mobile robot with four wheels and two motors and an ultrasonic sensor mounted on a DC motor. We equipped the robot both with a WIFI router configured as a client bridge of the main router, and an IP camera connected to the control system.

A. Control architecture :

Our control platform is based on an embedded system that comprises essentially a real-time microcontroller in charge of communication and control, FPGA hardware for the management of robot inputs/outputs (Figure 4). For the implementation of our control system, we used the LABVIEW tool [18] that offers many specific modules for data communication, data processing, control, etc.

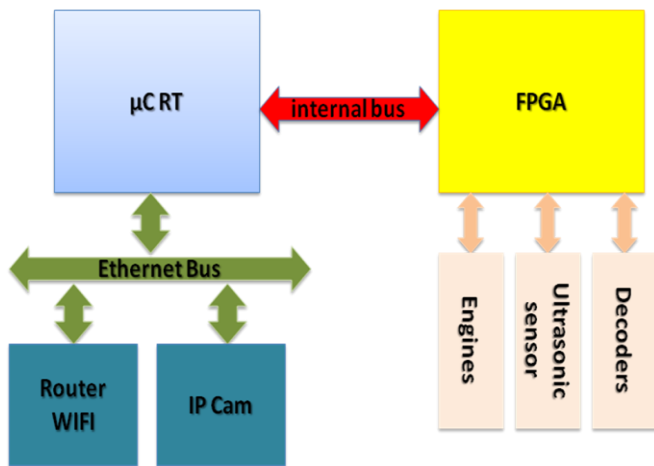


Figure 2. A hardware architector of control system

B. The implementation of some functions of telecontrol platform:

1) The obstacle avoidance function :

The functions of obstacle avoidance [16,17], having a local reactive behavior, that must be effective against all obstacles, Our robot mounted with a proximity sensor (ultrasonic sensor), the sensor is mounted on a DC motor on the front side of robot, which allows a rotation of 130° as shown in Figure 2, This rotation allows the robot to detect obstacles in front of the robot in a distance of 2m. Each measurement is represented by two variables, the sensor rotation degree who increment of 2° for each measure, the distance between the robot and the obstacle. The angle and distance curves are used as the membership functions. We use for decision a fuzzy logic module.

The control system calls the obstacle avoidance function in the case where the robot detects a near obstacle (10 cm), autonomously the execution of this function gets away the robot from the obstacle. Once the robot detects a distance of trust (30cm) from the obstacle, the control system returns to handle the operator commands.

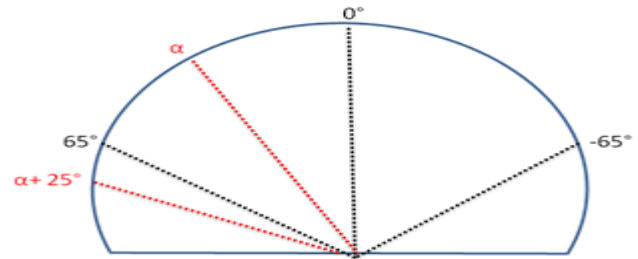


Figure 3. coverage angle of distance sensor

2) The motion control function :

The engine control system is implemented in two ways for the sake of comparison; the first on the FPGA and the second on the real time μC .

On the FPGA, we used a nonlinear control system based on a PID controller to stabilize the setpoint and to avoid the problem of engines pumping, over the control pulses management system which adjusts the motor speed according to Table 1.

TABLE I. ROTATION ANGLE IN TERMS OF FREQUENCY SETPOINT

Pulse Period (us)	Angle of rotation (rad)
600	$-\pi/2$
1050	$-\pi/4$
1500	0
1950	$\pi/4$
2400	$\pi/2$

To calculate the speed of the motors we used a module that is based on the pulses arriving from the encoder to deduce the actual speed of motor.

The engine control module implemented on the microcontroller allows:

- An increase / decrease of velocity controlled by the operator with a maximum value
- Display actual speed of each motor.
- The option to turn robot either left or right.

Because of network control problems already mentioned, we used an averaging filter to stabilize the speed setpoint, to turn the robot, this control system must decrease the motor rotation speed using the following equation:

$$\omega_1 = \omega_2(\lambda - \chi) / \lambda \quad (1)$$

Where

ω_1 : wheel speed side rotation (inside wheel)

ω_2 : external wheel speed.

χ : variable determines the degrees of rotation required (in our case 12 degrees).

λ : constant ($\lambda \in [-\chi, \chi]$).

C. Navigation System :

Several algorithms for mobile robot navigation were developed [25], The navigation control system is in charge of acquiring images from the IP camera, and does some basic image processing ,then compresses the images and transfers them to the operator control panel using the UDP protocol for a higher speed.

D. P2P interface :

This interface is a control panel (Figure 4) created under LABVIEW, which opens a peer to peer communication with robot control system by specifying an address and a port for communication, the interface gives the user a full control of the robot, displays its state and video streaming, the advantage of this interface is the simplicity of operation and data security, because the communication data are encoded by a LABVIEW cluster which are interpreted just by the command panel and robot control system, but the major disadvantage of this interface is the need of the LABVIEW Platform and the control interface in the command station to guide the robot.

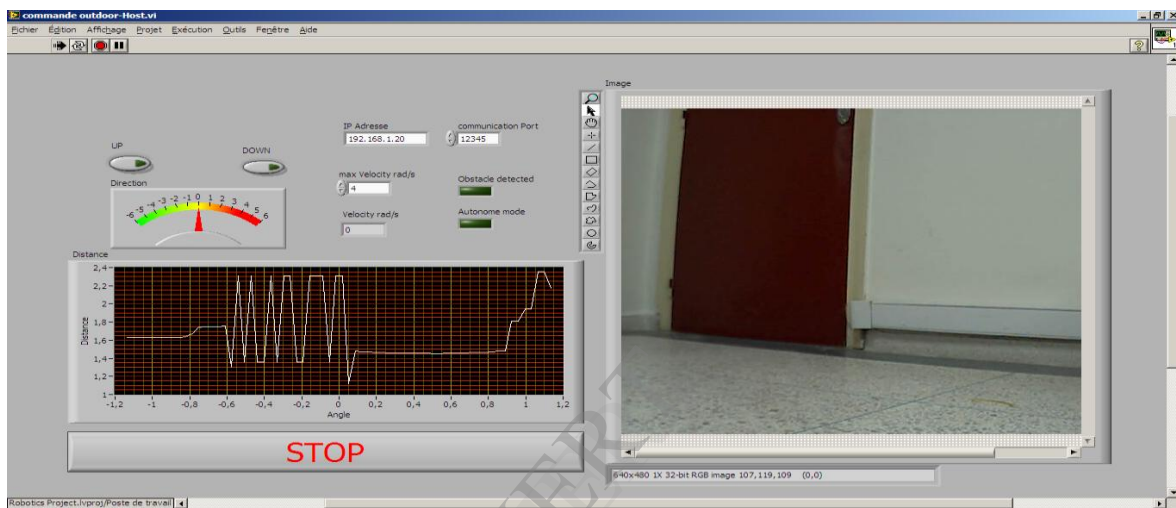


Figure 4. P2P interface

E. Web interface :

This solution allows the user to access the robot control interface (Figure 5) by using a web browser. We implemented a web client on the robot control system that allows the system sharing the command interface via the network using HTTPS protocol. This interface gives the user a simple access to the robot through a simple web browser; but on the other side, this solution presents some disadvantages such as a lack of system security and an overloading of the control system by adding a web client.

around the points B, C, D. This is normal because the user changes the direction in these points which imply an automatic decrease of speed.

IV. RESULTS AND DISCUSSIONS :

We test our telecontrol system using the path shown in Figure 6. The human operator uses the commands UP, DOWN and DIRECTION to control the robot to move from A to D (Figure 6). We notice that the robot moves more slowly

The robot starts to increase the speed every time it receives an UP command from the operator, and the speed becomes stable once it gets the max value defined by the operator. When the robot becomes closer to an obstacle, the control system informs the operator about the obstacle and automatically reduces the speed to a reasonable value. When the robot is more close to the obstacles (~ 10 cm), the speed jump to 0 rad/s and the control system uses the function of obstacle avoidance autonomously to get the robot out of danger ; once the robot is brought out of danger, the control system comes back to execute the operator commands.

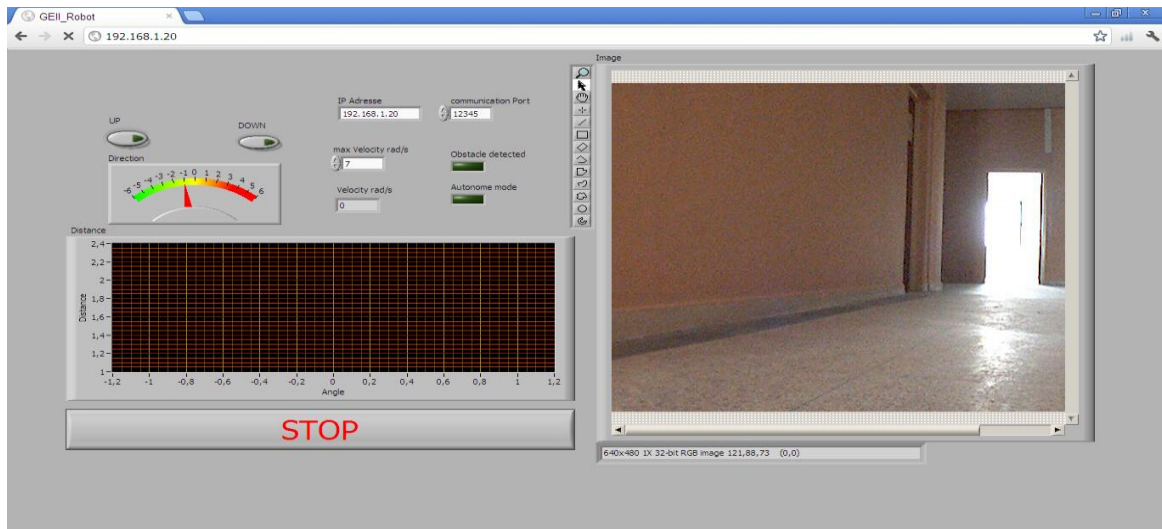


Figure 5. Web control interface



Figure 6. Robot test set in the path from A to D

V. CONCLUSION :

In this paper, we propose an interactive remote control model (telecontrol), accessible remotely with two interfaces: telecontrol using P2P applications to communicate with the robot, or a web-based telecontrol using a web client run in the robot system for sharing control interface via the Internet.

The control system is designed to perform various tasks independently and to react to expected events while the robot is dealing with unexpected events, for navigating in an unknown and dynamic world. The control interfaces allow the operator in the normal situation to control robot movements, display information about the robot and the video-streaming, once the robot detects a danger situation the control system moves to standalone mode to get away the robot from the risk, based on local intelligence, and return to execution of the operator commands otherwise.

The experiments demonstrate that the results are promising. As perspective, we plan to go deep in this work and to add some extensions about quality and fulfillment of real-time properties.

REFERENCES

- [1] K. Taylor, J. Trevelyan, Australia's telerobot on the web, in: International Symposium on Industrial Robots, 1995, pp. 39–44.
- [2] K. Goldberg, S. Gentner, et al. The Mercury project: a feasibility study for Internet robots, *IEEE Rob. Autom. Mag.* 7 (March(1)) (2000) 35–40.
- [3] R. Simmons, Fernandez, et al., Lessons learned from Xavier, *IEEE Rob. Autom. Mag.* 7 (2) (2000) 33–39.
- [4] S. Thrun, M. Bennewitz, et al., MINERVA: a second-generation museum tour-guide robot, *IEEE Int. Conf. Rob. Autom.* 3 (1999) 1999–2005.
- [5] P. Saucy, F. Mondada, KhePonTheWeb: open access to a mobile robot on the Internet, *IEEE Rob. Autom. Mag.* 7 (March (1)) (2000) 41–47.
- [6] R. Siegwart, P. Saucy, Interacting Mobile Robots on the web, in: *ICRA'99*, Detroit, MI, USA, May 1999.
- [7] H. Huosheng, Y. Lixiang, et al., Internet-based robotic systems for teleoperation, *Int. J. Assembly Autom.* 21 (2) (2001) 1–10.
- [8] P. Li, W. Lu, Implementation of an event-based Internet robot teleoperation system, in: *Fourth World Congress on Intelligent Control and Automation*, vol. 2, 2002, pp. 1296–1300.
- [9] M.R. Stein, The PumaPaint project, *Autonom. Rob.* 15 (2003) 255–265.
- [10] J. Cui, Z. Sun, P. Li, Visual technologies in shared control mode of robot teleoperation system, in: *Fourth World Congress on Intelligent Control and Automation*, vol. 4, 2002, pp. 3088–3092.

- [11] P. Li, W. Lu, Implementation of an event-based Internet robot teleoperation system, in: Fourth World Congress on Intelligent Control and Automation, vol. 2, 2002, pp. 1296–1300.
- [12] U. Nehmzow, Mobile robotics: a practical introduction, Springer-Verlag, London, 2000.
- [13] T.B. Sheridan, Telerobotics, automation and human supervisory control, The MIT Press, London, England, 1992.
- [14] R.C. Luo, K.L. Su, Networked intelligent robots through the Internet: issues and opportunities, IEEE Proc. 91 (3) (2003) 371–382.
- [15] Meng Wang, James N.K. Liu, Interactive control for Internet-based mobile robot teleoperation Robotics and Autonomous Systems, Volume 52, Issues 2–3, 31 August 2005, Pages 160-179
- [16] S. Thongchai, S. Suksakulchai, D.M. Wilkes, N. Sarkar, Sonar behavior-based fuzzy control for a mobile robot, in: IEEE Conference on Systems, Man, and Cybernetics, vol. 5, 2000, pp.3532–3537.
- [17] A. Saffiotti, Fuzzy logic in autonomous navigation, in: D.Driankov, A. Saffiotti (Eds.), Fuzzy Logic Techniques for Autonomous Vehicle Navigation, Physica-Verlag, Heidelberg, New York, 2000, pp. 3–22.
- [18] <http://www.ni.com/labview/>
- [19] J. Chung, B.S. Ryu, H.S. Yang, Integrated control architecture based on behavior and plan for mobile robot navigation, Robotica 16 (1998) 387–399.
- [20] R.C. Luo, T.M. Chen, Development of a multi-behavior based mobile robot for remote supervisory control through the Internet, IEEE/ASME Trans. Mechatronics 5 (December (4)) (2000) 376–385.
- [21] G. Cheng, A. Zelinsky, Supervised autonomy: a framework for human–robot systems development, Autonom. Rob. 10 (2001) 251–266.
- [22] V. Mut, J. Postigo, E. Slawinski, B. Kuchen, Bilateral teleoperation of mobile robots, Robotica 20 (2002) 213–221.
- [23] K.H. Han, S. Kim, Y.J. Kim, J.H. Kim, Internet control architecture for Internet-based personal robot, Autonom. Rob. 10 (2001) 135–147.
- [24] A. Halme, J. Suomela, M. Savela, Applying telepresence and augmented reality to teleoperate field robots, Rob. Autonom. Syst. 26 (1999) 117–125.
- [25] G.N. DeSouza&A.C. Kak. Vision for Mobile Robot Navigation : A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 2, pages 237–267, 2002.

IJERT