Demonstration of several methods of teleoperation and telepresence in a mobile robot

Santiago Morante Cendrero

***Abstract*— In this work, the author tests two methods of teleoperation in a mobile robot. First, a web interface, which has a special version for mobile devices. Second, a wireless joypad version which, combined with a Kinect camera mounted on the robot, improves the final performance. These elements provide feedback to user (color images and xpad vibrations). The remote control is made through WI-Fi connection and is managed by YARP.**

The robot used, named ECRO, is a research platform belonging to the Robotics Society of the Universidad Carlos III de Madrid.

I. Introduction

Teleoperation is an old topic in robotics field, being named first time by Tesla in late 1800s [1]. This term is defined as to operate a machine remotely. The remote control does not include any kind of feedback, so the operator don't have the feeling of being where the machine (in our case, the robot) is.

On the other hand, telepresence has been developed in last 30 years only. Telepresence is the evolution of teleoperation, including elements to allow more natural interaction between the robot environment and the remote user. Those elements usually include, on the user side, actuators to produce reactions.

Those reactions can be related with sensors measures, in the sense of being similar stimulations (tactile-tactile, temperature-temperature), or, in other cases, be completely unrelated but proportional (force-sound, distance-vibration).

II. ECRO Robot

The platform to be controlled in this work is property of the Robotics Society of the Universidad Carlos III de Madrid (figure 1). This robot is called ECRO (figure 2).

Fig. 1. Logo of the Robotics Society. Apart from ECRO research, other lines are being developed in ASROB like: UAV, humanoids, 3D printers and games.

S. Morante is an Automation Engineer from Universidad Carlos III de Madrid and student of the Master in Robotics and Automation `santiago.morante at alumnos.uc3m.es`

A. System Overview

The robot ECRO (acronym for *Earth Civil RObot*) is mainly a group of devices, including laptop and electronics, all mounted on a wheeled base of aluminium.

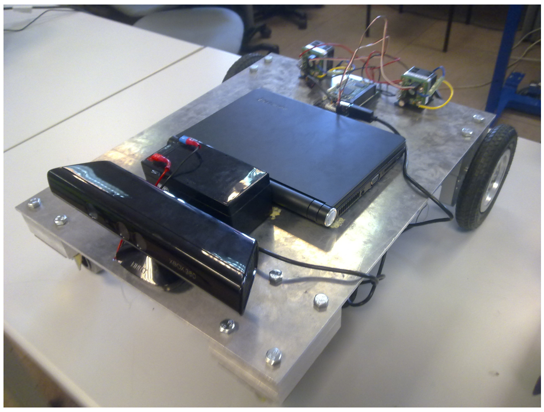

Fig. 2. This robot serves as research platform to ASROB members, specially those related with navigation, autonomy, teleoperation and computer vision.

The ECRO project has many devices to be attached to the robot, and those devices are changed in function of the work being developed. For this work, the robot contains the following devices:

- **Wheels**: Two electric wheels with 12 VDC motors and 15 cm rubber tires.
- **Netbook**: Small Lenovo Laptop with a Linux distribution installed (Ubuntu).
- **Skymega**: Arduino compatible board that serves as meeting point for devices (this is an evolution of open source hardware board Skypic [2]).
- **Xbox Wireless Gaming Receiver**: 9-meters range base for joypad. It is connected via USB to netbook.
- **Kinect**: Motion-sensing device. Used as webcam and range camera.

All these items are mounted on the base, and connected with wires among them (figure 3).

B. Communications Architecture

All the system is interconnected by YARP [3], a very popular open-source software in robotics. For instance, the iCub robot [4], an european open source cognitive humanoid robotic platform, uses it as main software. YARP, is formally called a middleware, which means that YARP only establishes the communication among sensors, actuators, computers, etc.

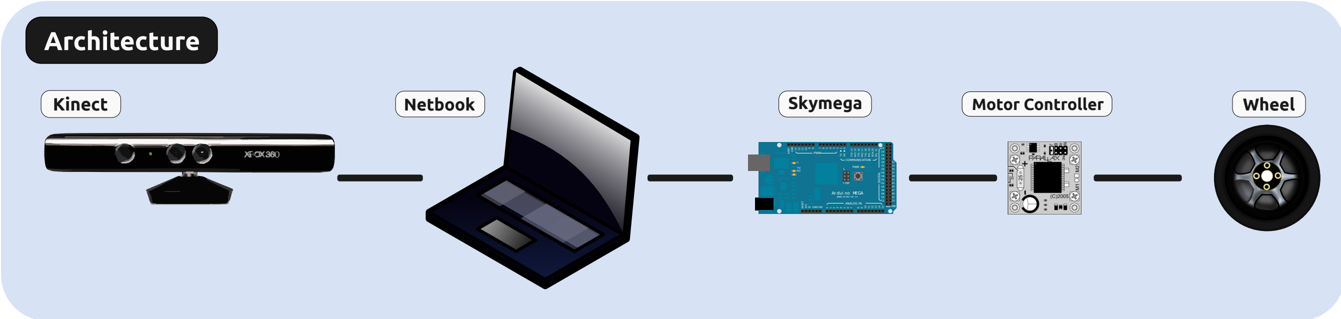

Fig. 3. Those devices are wired on the base. There is no wireless signals between the elements. Netbook provides Wi-Fi connection and the Xbox Receiver is consider part of it. This receiver emits their own wireless signal to Joypad.

In our system, this library allows the publication of information through wireless connection via *carriers*. This corresponds to the kind of transport used to carry data. In Kinect images case, the carrier is **mjpeg** (mjpeg-over-http).

C. PWM Motors

Once we have established communications, the last step is to control the wheels. This task is achieved by Skymega board. This board is directly wired to drivers (HB-25 model). These drivers are controlled in some sense like a servo, but instead of doing position control, they do velocity control. For example, these drivers don't need "refreshing" (resend the signal constantly), because they keep the velocity until another signal arrives.

The control method is PWM (Pulse Width Modulation), which is a very common way of controlling servos. As a summary, intensity of the command is directly proportional to the width of the pulse (with boundaries). The width limits are 0.8 ms (lower) and 2.2 ms (upper). There is a "blank time" of 5 ms where the drivers ignore will ignore incoming pulses.

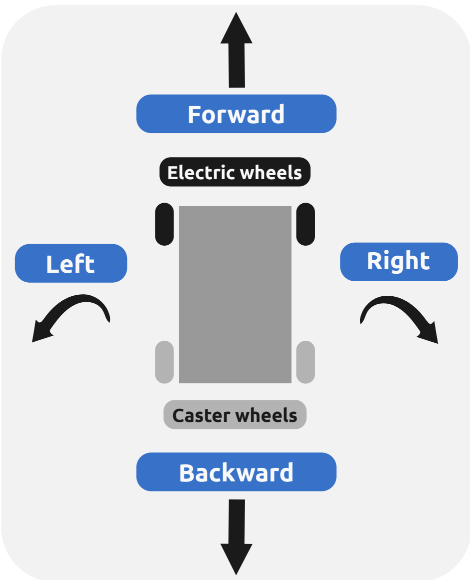

Fig. 4. Movement options of ECRO.

The mechanical configuration of ECRO is seen in figure 4. There are two frontal electric wheels and two posterior caster wheels. The electric wheels can't turn, so movements are based in *Differential Wheels*, applying different velocities to each wheel to be able to rotate.

III. Demonstration

To test teleoperation (and telepresence), in this work, two different ways of remote control has been developed and tested:

- **Web interface**: an HTML web page has been provided with buttons which send orders to wheels motors. This way doesn't include any kind of feedback, so it can be considered as teleoperation.
- **Wireless Joypad**: a wireless controller from a popular platform (Xbox) directly controls the movement of the robot with usual games combination of buttons. In this case, two signals are returned to the operator, so it will be called telepresence.

In next section, those two different ways will be deeply explained. A video of the whole system working with both systems can be found at:

`http://www.youtube.com/watch?v=_9e7dF-dl7I`

A. Method I: Web Interface

In this first method, teleoperation is "blind". The signals go in only one direction, from the user to ECRO, so there is no feeling or interaction between the operator and the environment.

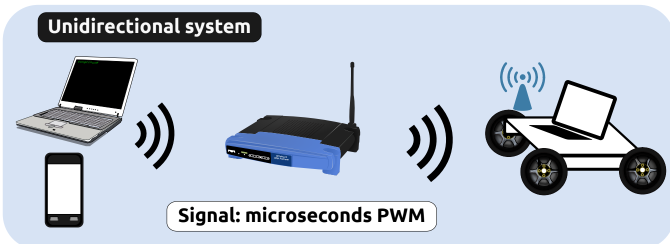

Fig. 5. There is no feedback returning, so the user can't feel ECRO environment.

This is what can be called *Unidirectional system*, as seen on figure 5. The interface to send commands is in this case a web interface.

The web interface allows the possibility of controlling the robot with any electronic device with internet capabilities. A previous version of this interface was developed by the author for a paper [5] published in *Robocity2030 9th Workshop*.

The visual design was programmed in HTML and JavaScript language, and an internal HTML to Python parser (called *CherryPy*) was used to translated the button press into YARP commands.

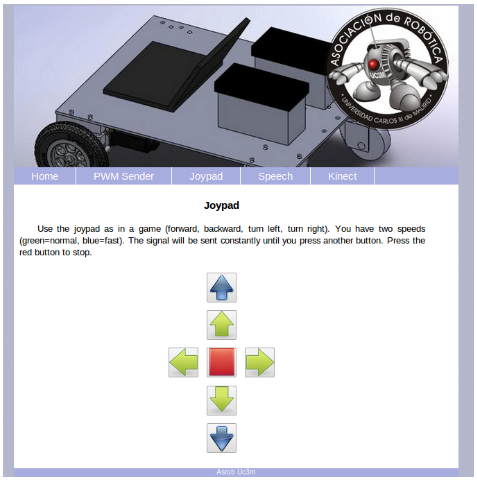

Fig. 6. One of the tool of the interface is a game-like joypad. The page refreshes every time a button is pressed and the signal is maintain until another interaction.

Different tools are organized in tabs, including direct motor commands sender or speech recognition, but in the case of actual teleoperation, only joypad is going to be described.

This tool has several direction buttons (figure 6), one for each direction (forward, backward, left and right) and two extra for maximum velocities in linear directions (blue ones). Another last button to stop the robot is also included.

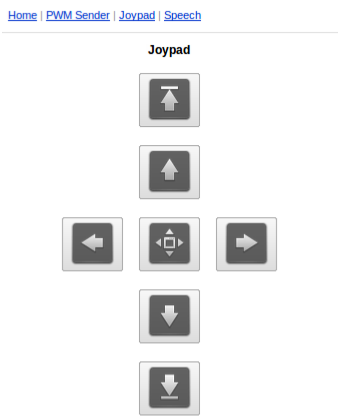

Fig. 7. The mobile version allows small screen devices to easily operate ECRO.

This web interface is size-adapted for medium/high screens, but small devices could have problems with the refresh. The figures will misplaced on every refresh. This is why there is a mobile adapted version (figure 7).

B. Method II: Wireless Joypad

In the second method of teleoperation, we go a step further and apply concepts of telepresence. This concept implies to add returning signals (feedback) to our system. First, and to accomplish this, some elements of the previous system has been replaced, and other have been added.

One change comes from the interface part. We have substitute the interface to send commands for a wireless game controller (xpad), specifically a Xbox 360 wireless controller (figure 9). This device possesses their own receiver, which is connected to the computer.

Fig. 9. The Xbox controller has a lot of available buttons for use.

1) Haptic-like device and feelings substitution: One of the advantages of the xpad is the possibility of rumbling (vibration). Although xpad is not a haptic device, it can be programmed to behave like one. This programmed vibration will serve as an indicator of proximity to ECRO. The closest and object, the stronger the vibration is. This replacement of distance by vibration is an example of *feelings substitution*, a common topic in teleoperation.

Obviously, the Xbox controller is handle with both hands, so this rumbling can be considered like *tactile feedback* (tactile information and virtual reality have been mixed before, e.g. [6], [7]).

2) Accuracy improvement: Another advantage is the analog-like buttons. These are situated in the posterior part of the controller and offer a continuous signal (varying from 0 to 255), which will be used as throttle (pressure in the button rises the velocity) of the mobile robot.

3) Remote vision: Apart from vibration, there is another feedback signal which has not been named yet.

Fig. 10. The range camera create an infrared map to measure distances.

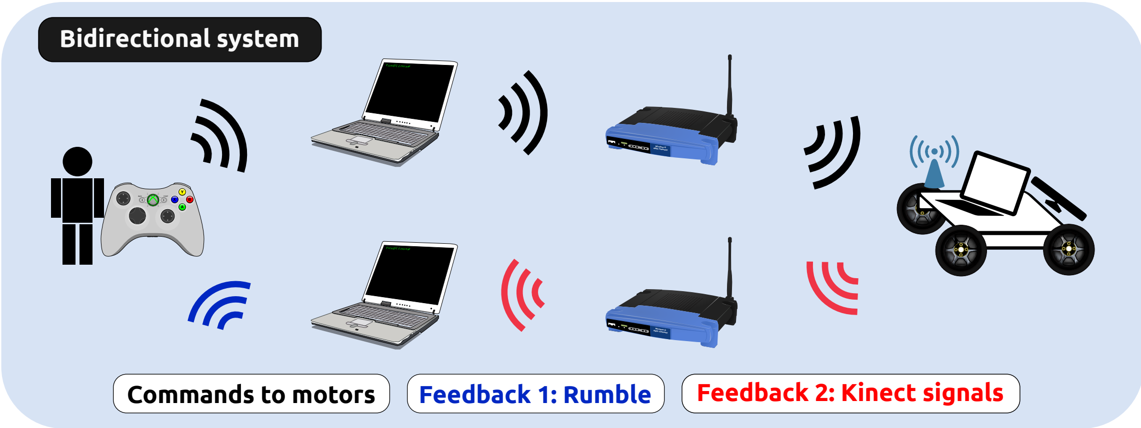

Fig. 8. This architecture performs the *Human In the Loop* paradigm.

In this configuration the user can also see what ECRO sees on the screen thanks to a camera mounted on the platform. This allows to navigate without direct looking to the mobile robot, so real telepresence can be achieved.

Up here, user side news, but there are other changes on mobile robot side. To be able to measure objects distances and provide robot point-of-view images, there is a camera device with depth sensor capabilities. This device is Kinect (figure 10), a very popular element nowadays in robotics. The capabilities of Kinect are being used in teleoperation in the user side [8] and in robot one [9], [10].

As seen on figure 11, there is an area between distances (**d min** and **D max**), where the vibration is enable. The lower limit (d min) is chosen taking into account Kinect hardware limitation (around 50-60 cm), below that limit, the camera cannot obtain distances. The upper limit, on the other hand, is chosen manually. We have decided to fix it on 1500 cm because it looks a reasonable object distance to care about.

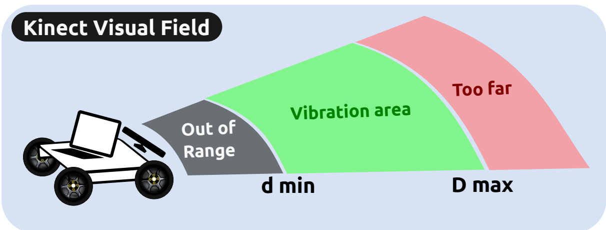

Fig. 11. There is only an area where the vibration is enabled.

Inside these limits, the vibration follows and inversely proportional function, because as the objects short the distance, the xpad rise the vibration level (eq. 1).

$$V_{out} = V_{upper} - \frac{V_{upper}}{D_{max} - d_{min}} * D_{actual} \quad (1)$$

Where V_{out} is the output level of rumbling, V_{upper} is the highest vibration level possible (an integer number in fact), D_{max} and d_{min} are the area limits, and finally, D_{actual} which is the actual distance measured by Kinect depth sensor.

This equation is performed twice, one per image side (left and right), to control distances from both sides at the same time. To measure this proximity, an average of pixel values determines D_{actual}.

4) General framework: All those devices are interconnected (wired or not) and can communicate between them using YARP software (just like before). The whole process can be seen in figure 8 and is defined as follows:

a) User to ECRO direction:

i. The user presses buttons in the **xpad**.
ii. The signal travels to the **Xbox receiver** (through wireless connection).
iii. The orders are translated into motor commands (microseconds that represents width of pulse for PWM).
iv. Those commands are sent to the internet with a **router**. In our case, we have created a local network, but global network can be used too.
v. The signals arrives wirelessly to the **laptop** mounted in the robot (we use Wi-Fi, but 3G network is technically affordable).
vi. Laptop sends them to the **Skymega board** (wired).
vii. The board transmits them to the **drivers**.
viii. And finally the drivers translate the widths of pulse into power energy for the **wheels**.

This is the control commands direction, but now we follow the feedback returning to the user.

b) ECRO to user direction:

i. **Kinect** sensor measures distances of near objects and sent the data via USB to mounted laptop.
ii. Data travels through internet to user's **computer**.
iii. The images are shown on the **screen** and vibration commands are sent to the xpad.
iv. **Xpad** vibrates in function of distances.

Once here, the loop continues from the user again. The whole system considers human as part of the process.

IV. COMPARATIVE

Both methods used has their points, so as a summary, we have list those differences to see them clearly:

A. Web Interface

+ **No specialized devices are required, the only need is internet capability**: Tablets, PDA, laptops, netbooks, in fact, any modern device can be the controller device of the robot by using the web page (even many people at the same time can do it) .
+ **Possibility of use with smartphones:** The specialized page for small screens improves the driving control.
- **Very limited tools:** The only item to control the robot with, are button (or sliders), so there is no big variety and the performance is not astonishing.
- **Refresh problem:** Every time a button is pressed, the whole page refreshes. This slows the answer of the user for reacting events.
- **No feedback:** Robot must be present and visible to be controlled. There is not any returning signal from ECRO.
- **Less accuracy in movement, so less control:** As commented, the absence of feedback make the accuracy worse, so the control is less effective.

B. Wireless Joypad

+ **Several feedback signals:** Kinect images, both RGB and depth, mixing with rumble properties of Xpad provides a comfortable and real experience of control.
+ **Improvement in control:** Xpad configuration (physical buttons) and the small time of answer, have a perceptible impact in terms of control. Smaller time of reaction and higher precision in turns are the most remarkable facts.
- **More complex systems about configuration and programming:** Obviously those improvements commented over are not free. As the system incorporates new devices, the communications get more complex. Not only in programming, but also in wiring. More elements also reduce the batteries faster, reducing power autonomy time.
- **Necessity of specialized, and in some cases expensive, devices:** Despite Kinect is noticeable cheaper than other depth-capable devices, it has a high price for *Do it Yourself* robots. It can be a handicap for some groups interested in the topic. Xpad and its wireless controller must be bought too in order to acquire rumbling signals.

This comparative has shown the lights and shadows of every method. The chosen of one or another must be conditioned on the situation or project to be used in.

V. Conclusions

Some conclusions and thoughts about the system can be made. Let's start with pros and cons of our system election.

Some pros comes from the use of internet as transmission medium. Internet is available worldwide so both, the robot and the user, can be anywhere, no matter the distance between them. Another pro is low time delay because the use of Wi-Fi. Tests performed show small retardation.

The cons are referable to web interface construction. The code behind is programmed in two languages: HTML and JavaScript. That implies that with every new event (button pressed) the whole page refreshes. The problem is specially disturbing in medium screen where the page cannot be fully seen without scrolling. So every time it's necessary to command rapidly two consecutive movements, the scroll action can be an annoyance. One possible solution to the problem could be the use of AJAX (Asynchronous JavaScript And XML), which only refreshes the button pressed.

As last conclusion, highlight the advantages of telepresence over teleoperation. The experiments conducted, about telepresence, has shown a big improvement in driving ECRO because the user can feel the environment and avoid objects more accurately and faster. The only bad point of telepresence is the increase of complexity in communications.

Acknowledgments

I want to thank the Robotics Society of the Universidad Carlos III de Madrid for letting me use the ECRO robot, and all the material, for this project.

Special thanks to Juan G. Víctores, research assistant at Robotics Lab UC3M, for explaining and helping with YARP architecture and the basis of robot communications.

References

[1] N. TESLA, "Tesla," Nov. 8 1898, uS Patent 613,809.
[2] J. González-Gómez and A. Torres, "Hardware libre: la tarjeta skypic, una entrenadora para microcontroladores pic."
[3] G. Metta, P. Fitzpatrick, and L. Natale, "Yarp: Yet another robot platform," *International Journal on Advanced Robotics Systems*, vol. 3, no. 1, pp. 43–48, 2006.
[4] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, "The icub humanoid robot: an open platform for research in embodied cognition," in *Proceedings of the 8th workshop on performance metrics for intelligent systems*. ACM, 2008, pp. 50–56.
[5] J. G. Víctores, A. Jardón, S. Morante, M. F. Stoelen, S. Martínez, and C. Balaguer, *Interacción humano-robot a través de interfaces en la nube*, 2011, pp. 75–92.
[6] Y. Ikei, K. Wakamatsu, and S. Fukuda, "Vibratory tactile display of image-based textures," *Computer Graphics and Applications, IEEE*, vol. 17, no. 6, pp. 53–61, 1997.
[7] D. Kontarinis and R. Howe, "Tactile display of vibratory information in teleoperation and virtual environments," *Presence: Teleoperators and Virtual Environments*, vol. 4, no. 4, pp. 387–402, 1995.
[8] W. Song, X. Guo, F. Jiang, S. Yang, G. Jiang, and Y. Shi, "Teleoperation humanoid robot control system based on kinect sensor," in *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2012 4th International Conference on*, vol. 2. IEEE, 2012, pp. 264–267.
[9] W. WJ IV and D. Bevly, "Using the microsoft kinect for 3d map building and teleoperation," in *Proceedings of IEEE/ION PLANS 2012*, 2001, pp. 1054–1061.
[10] Y. Toda, T. Narita, and N. Kubota, "Information visualization based on 3d modeling for human-friendly teleoperation," in *Evolutionary Computation (CEC), 2012 IEEE Congress on*. IEEE, 2012, pp. 1–7.