

# Electronics proposal for telerobotics operation of P3-DX units

Felipe Espinosa, Marcelo Salazar, Daniel Pizarro and Fernando Valdés  
*University of Alcala. Electronics Department  
Spain*

## 1. Introduction

Telerobotics is the area of robotics concerned with the control of robots from a distance, mainly using wireless connections or the Internet. It is a combination of two major subfields, teleoperation and telepresence. The work presented in this chapter belongs to the field of teleoperated robots, where a remote centre sets commands to the robot and supervises the performed motion by receiving feedback from its sensors. In teleoperated robots the control algorithm can be balanced between the remote host and the local host in the robot, which yields to several kind of possible control schemas.

The key components needed to develop telerobotics applications are the following: control (algorithm and real time implementation), sensors (world sensing and information processing) and wireless communication (generally using standard wireless technologies, i.e. IEEE 802.11) [Angelo, 2003], [Anvari, 2007], [Gumaste, 2007], [Mehani, 2007], [Chumsamutr, 2003], [Hespanha, 2007], [Bambang, 2007].

This chapter is outlined within both educational and research fields in telerobotics, and so its aim is to offer a reliable and low cost architecture to be implemented in research labs. The robotic platform consists of the Pioneer 3DX (P3-DX) from the company MobileRobots (see Figure 1). It is made of an aluminium body (44x38x22cm) with 16.5cm diameter drive wheels. The two DC motors use 38.3:1 gear ratios and contain 500-tick encoders. The differential drive platform is highly holonomic and can rotate in place moving both wheels, or it can swing around a stationery wheel in a circle of 32cm radius. A rear caster is included for balancing the robot. On flat floor, the P3-DX can move at speeds of 1.6 mps. At slower speeds it can carry payloads up to 23 kg. In addition to motor encoders, the P3DX base includes eight ultrasonic transducer (range-finding sonar) sensors arranged to provide 180-degree forward coverage. This robot includes a 32-bit RISC-based controller, with 8 digital inputs and 8 digital outputs plus 1 dedicated A/D port; 4 of the outputs can be reconfigured to PWM outputs [P3-DX, 2009].

The P3-DX can be ordered with a complete electronic hardware [MobileRobots, 2009], which include wide range sensors, an on-board PC and Wireless Ethernet communication device. However, the authors propose to start from a basic structure that allows to be customized depending on the final application. This decision offers the opportunity of working with open platforms which is specially suitable for educational labs in engineering schools. On

the other hand, the final cost of the prototypes is substantially reduced using general purpose hardware and developing ad-hoc software as it is detailed next.



Fig. 1. Basic robotic platform of Pioneer 3-DX.

In the context of telerobotics some questions must be addressed: which are the features that a robot must have to be teleoperated and how to provide a robotic platform with low cost devices so that such features are implemented.

To become a teleoperated robot, three subsystems are needed: control, communications and sensors.

From the control side three levels are proposed in this document:

- Low level control (LLC), which directly controls the active wheels of the robot. The P3-DX includes a PID for each active wheel [P3-DX, 2009].
- Medium level control (MLC), for path following. In this document a linear servo system is proposed. [Ogata, 1994], [Dutton et al., 1997].
- High level control (HLC), where a more complex control is required and extra sensors which give richer information about the environment. As an example, in platooning applications, the HLC determines the path by sensing the distance and relative position of the preceding follower, [Kato et al., 2002], [Espinosa et al., 2006].

From the communications side, a wireless network (short range for indoor applications) is required with a topology depending on the application: Using a star network topology, one or several teleoperated robots behave as wireless nodes whose master node is the remote centre, (see Figure 2.left). In applications where all robots must share the same information a fully connected mesh topology is preferable (see Fig. 2.right).

From the point of view of the sensor included in the robot, both the application and the environment drive the quality specifications and amount of information required for following the commands sent by the remote centre.

If the environment is free from obstacles and the paths are not very large, the odometry information included in the P3-DX can be an option. However, if the paths are large or repetitive, the accumulative error present in the dead-reckoning techniques must be compensated with an absolute localization method (e.g. vision sensors or infrared-beacons)

[Borenstein et al., 1996]. If the application requires the detection of obstacles with a field of view of 180°, 5 meters of depth and a 0.1 feet resolution, the built-in sonar system in the P3-DX platform can be reliable and enough. Contrary, if more accuracy is needed a laser-range sensor, such as the Hokuyo Scanning Laser Range Finder [Hokuyo, 2009] is proposed.

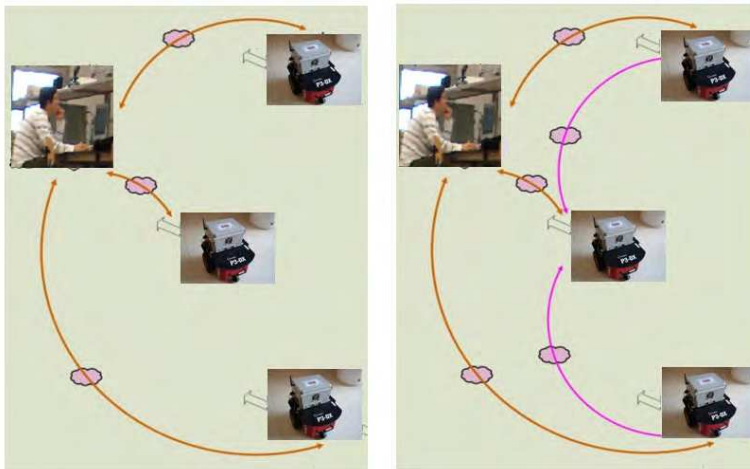


Fig. 2. Example of telerobotics operation: without (left) and with (right) cooperation among robot units.

The basic hardware included in the P3-DX is not enough for supporting the control, communication and sensing requirements in robotic teleoperation. According to the authors' experience, the minimum specifications are the following:

- Embedded PC with native x86 architecture and at least: 2 USB ports, 1 firewire header, on-board LAN, 1 SATA connector, and mouse and keyboard ports.
- SATA Hard Disk of 10 Gb, to save long experimental data.
- Wireless ethernet converter, server and client modules, allowing security system and transmission rate superior to 10 Mbps.
- Additional sensor system to improve the obstacle detection.
- Real Time Operating Systems for control and communications tasks implementation.
- Development tools for low level robotics applications.

## 2. Hardware architecture

In the previous section, the basic hardware of P3-DX has been presented as it is shipped from MobileRobots in its basic configuration (See Figure 1). The more relevant subsystems of this electronic architecture are: Hitachi microcontroller, encoders, sonar ring and the global power electronics from a battery pack. The microcontroller is in charge of, among other functions, executing the LLC loop (PID) of each motor in the active wheels (Left and Right). The LLC obtains feedback from the odometry sensors in the wheels [P3-DX, 2009]. Graphically, the block diagram of this electronic architecture is showed in the Figure 3. (left part).

In the following lines the hardware and software, designed specifically for the teleoperation of the P3-DX robot are described.

## 2.1 Hardware components

Taking into account the aforementioned specifications, the hardware proposed in this chapter consists of the following elements: motherboard VIA EPIA EN Mini-ITX, which incorporates enough ports and slots for supporting new sensors. Ethernet converter WLI-TX4-G54HP, 80 Gb hard disk and DC-DC converter with enough output power for driving the motherboard and the sensors. In Figure 3, it is compared the block diagram of the original hardware provided by MobileRobots and the proposed hardware add-ons proposed by the authors. A more detailed explanation of the hardware is described next.

The VIA EPIA EN15000G Mini-ITX mainboard includes the 1.5 GHz VIA C7 processor, fully compatible with Microsoft® Windows® and Linux operating Systems. This motherboard integrates 1 on-board LAN controller working at 100/1000 Mbps, 1 IEEE 1394 firewire header and 1 PCI expansion slot. Moreover its back panel I/O includes several ports: PS2 mouse and keyboard, VGA, serial, RJ-45, RCA, S-Video, 3 audio jacks and 4 USB 2.0 [Via-Epia, 2009]. The computing capabilities and versatility of the proposed mainboard allows the easy development of robotics and telerobotics applications.

An external hard disk of 80 Gb is connected to one of the free SATA ports, this way enough capacity is kept for debugging software tools and saving information from experimental tests.

The different voltage levels required by the hardware (i.e. +3.3V, +5V y +12V) are provided by the MOREX QT-16045 DC-DC converter.

In order to implement the wireless communication, required in a telerobotics application, see Figure 2, two modules are needed: a router (generally in the remote centre) and a client for each teleoperated robot available in the network. The wireless router chosen for this application is the Buffalo WHR-HP-G54, which is in compliance with standards IEEE802.11b/.11g, offers 11 frequency channels and allows high-speed transmission rate of 125 Mbps. Others characteristics are: wireless security WPA-PSK and 128 bits WPE, 4 LAN ports, high-speed routing. The client chosen to connect to the router is the WLI-TX4-G54HP [Buffalo, 2009].

In Figure 4 it is shown the appearance of the electronic hardware detailed before, whose cost (September 2009) is under 700 \$ USA.

Some details about the P3-DX platform are shown in Figure 5, incorporating the proposed electronics subsystem. The prototype of the picture is part of the research developed by the authors in the COVE project ( Intelligent Transport System for Cooperative Guidance of Electrical Vehicles in Special Environments ) held in the Electronics Department at the University of Alcalá (Spain).

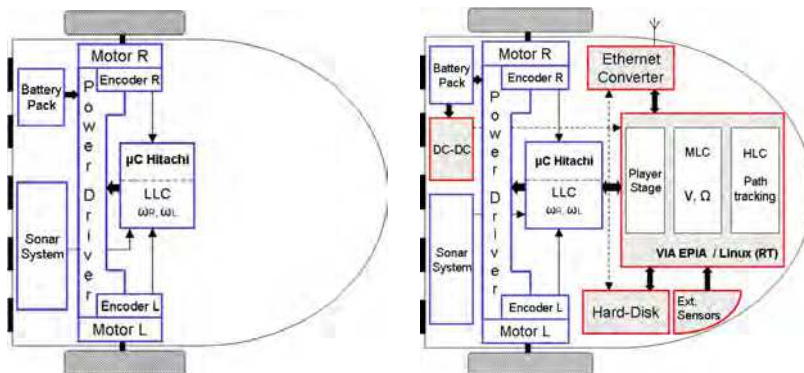


Fig. 3. Block diagram of the P3-DX basic electronic architecture and the add-ons proposed for telerobotics applications.



Fig. 4. Main electronics devices involved in the designed architecture for teleoperation of the P3-DX.

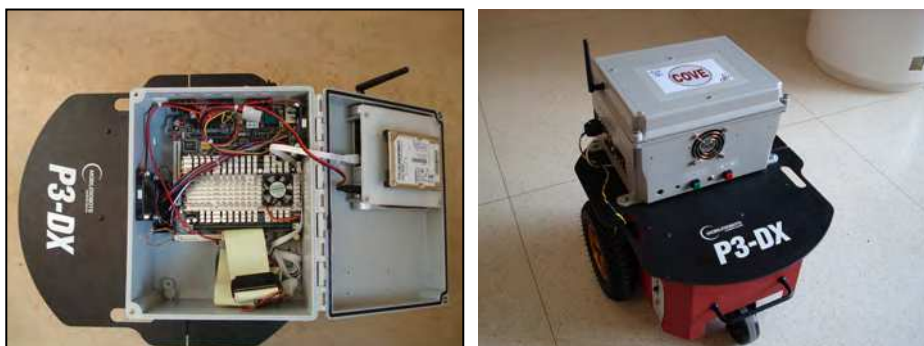


Fig. 5. Photographs of the P3-DX modified version implementing the new hardware architecture hardware for telerobotics applications.

## 2.2 Software components

In order to develop the teleoperation capabilities aforementioned, a multilevel software platform is proposed, which combines high level UML design and control, given by the platform Matlab/SIMULINK [MathWorks, 2009], and the low level client-server software PlayerStage [PlayerStage, 2009], which communicate with P3DX microcontrollers and all sensors in the robot. Both client/server platforms are hosted in Linux/RTAI [RTAI, 2009] soft real time operating system to ensure accurate control over periodic tasks and low latency.

The general diagram of the software platform as well as the input/output topology is shown in Figure 6.

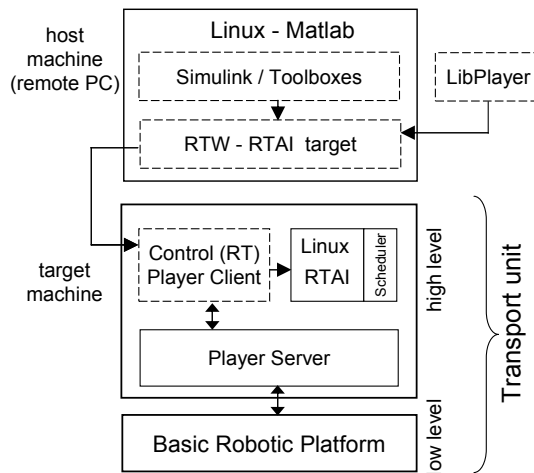


Fig. 6. Software tools integration for control design in telerobotic applications.

Roughly speaking the software architecture can be described in two main parts, the client-server architecture based on Player/Stage and the Matlab/Simulink driver using the Real Time Workshop (RTW) framework [MathWorks, 2009].

### 2.2.1 Client-server architecture based on Player/Stage

The Software Player/Stage (P&S) consists of a complete suite for control, communication and simulation of mobile robots under UNIX operating systems [PlayerStage, 2009].

Player is a network interface for robotic devices: controllers, sensors and actuators. It is a hardware abstraction layer that allows the coordination and distribution of tasks within and between robots. The client-server model of Player easily allows robot control programs to be executed locally or remotely in a connected remote centre. In addition, the open structure of Player allows writing the algorithms in diverse programming languages and executing them in Real Time operating systems (i.e. Linux RTAI).

For the case of the P3-DX platform, Player offers a standard interface for mobile robots that allows sending high-level commands (angular and linear velocity) for motion in a 2D surface as well as dead-reckoning functions for retrieving the 2D position of the mobile

robot [Whitbrook, 2006]. In addition, player offers direct access to sensors onboard the robot, such as laser range or ultrasound rings.

Stage is a simulator of multiple mobile platforms moving in a plane (2D). More precisely, it provides a virtual world for robots and objects that can be detected by sensor systems onboard the robots. This software makes easier the transition between simulation and the physical implementation of the control algorithms.

In order to perform movements with the robotic platform, it is necessary to run at least two programs in coexistence: the P&S server and the client one (see Figure 7).

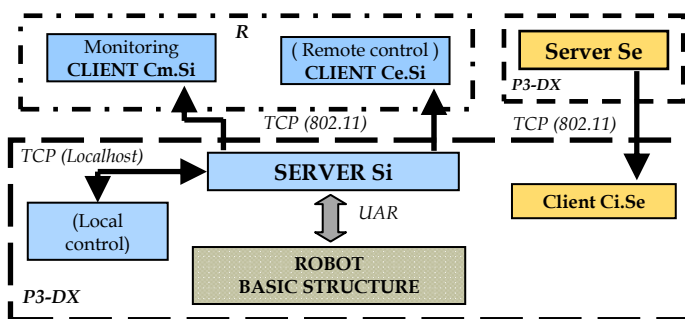


Fig. 7. Proposal of client-server architecture based on Player/Stage for the development of telerobotics application with P3-DX units.

The server (Player Server), namely  $S_i$ , establishes communication with the robot hardware (in this case the basic structure of the P3-DX) using a serial link RS-232 (UART). The server offers information about the state of the robot and accepts orders to control it.

Connected to the  $S_i$  Player Server, there can be clients that are executed either locally (e.g.  $C_{i,S_i}$ ) or externally to the robot hardware (e.g.  $C_{e,S_i}$  and  $C_{m,S_i}$ ). The control program, which is always installed in client mode (Player Client)  $C_{i,S_i}$ , at each sample time  $T_s$  analyzes the state of the robot and decides the movements to be performed. In general there is only one control program for each robot whether it is locally executed  $C_{i,S_i}=C_{i,S_i}$  or run in a remote centre (RC)  $C_{i,S_i} = C_{e,S_i}$ .

In Figure 7, it can be shown two possible clients of the server installed in the P3-DX robot:

- Player client  $C_{m,S_i}$ , is a monitoring program installed for example to detect the location (position and orientation) or changes in motion (linear and angular velocity) of the robot.
- Player client  $C_{e,S_i}$ , performs several different tasks, as for example the control of the robot from the remote centre. In this configuration the RC is in charge of both monitoring and control.

The proposed client-server architecture allows a robot to execute clients which connect to other servers, namely  $C_{i,S_e}$  in Figure 7. This functionality is of great interest in cooperative guiding and robot platoon applications [Espinosa et al., 2008], where the controller of one unit needs to monitor the state of some other robots. To sum up, each server  $S_i$ , executed in a robotic unit can receive connections from the following client processes:

- Client embedded in the robotic unit  $C_{i,S_i}$ , with the tasks described for  $C_{i,S_i}$  if the control is local.

- Monitoring client  $C_m.S_i$  implemented in the RC.
- Clients embedded in other transport units  $C_e.S_i$ , with tasks of  $C_1.S_i$  if the robot control is remote mode.

The physical medium that supports the local client-server communication is transparent to the Player/Stage user. In remote mode this communication can be implemented using TCP/IP protocols over Ethernet (IEEE 802.3) or Wireless (IEEE 802.11). On the other hand, when the client is hosted in the same CPU as the server (e.g. local control configurations), the client-server communication keeps the TCP/IP layer over a loopback local device. This feature allows modifying control configuration with very little modifications in the design. This transparency is shown in Figure 7, where local ( $C_i.S_i$ ) and remote clients ( $C_e.S_i$ ) connect to the same Player server.

### 2.2.2 Matlab/Simulink driver in Real Time Workshop

The Matlab software offers a complete numerical/symbolical engine for scientific simulation. It is widely used among the control community, specially its integrated UML module, called Simulink, designed for simulation and control over continuous and discrete dynamical systems. Thanks to the Simulink environment, the control design is performed and tested in the same language, i.e. using differential equations and transfer functions to describe controllers and models. Simulink allows communicating with real peripherals, such as serial ports, network sockets or any other kind, generating custom blocks or "S-functions" which are integrated in the design. Besides simulation, the combination of Simulink with the Real Time Workshop (RTW) [MathWorks, 2009] tool allows compilation and execution of the designed controller in different targets, such as FPGAs, Digital Signal Processors, Microcontrollers or PCs with real time operating systems as it is the case in this paper.

By using custom blocks generated for Simulink it is possible to generate an S-function block that connects with a P3-DX platform which is running a Player server through a TCP/IP socket. The position of the robot and the state of all its sensors can be sent to the Simulink controller where a block-diagram like algorithm can be easily designed.

In this document, several control configurations are proposed, whereas the control algorithm is executed locally in the robot or externally, using always the framework offered by Simulink and the RTW compiling for Linux RTAI targets. In Figure 8 it is shown the common driver used to perform control of a P3-DX platform using the aforementioned Simulink/RTW schema. The block is composed of two main parts:

- The first part consists of an S-function or father process which is executed at each time sample driven by the inputs from a signal builder in the Simulink environment. The father process communicates using shared memory buffer with a child process that is in charge of network communications.
- The child process, which is executed in parallel, implements socket communication with the Player Server hosted in the robot. It sends angular and linear speed commands and receives position and sensor readings, which are sent back to the father process. The main justification of this father-child methodology is to separate blocking connections in the TCP/IP socket from the S-function main loop.



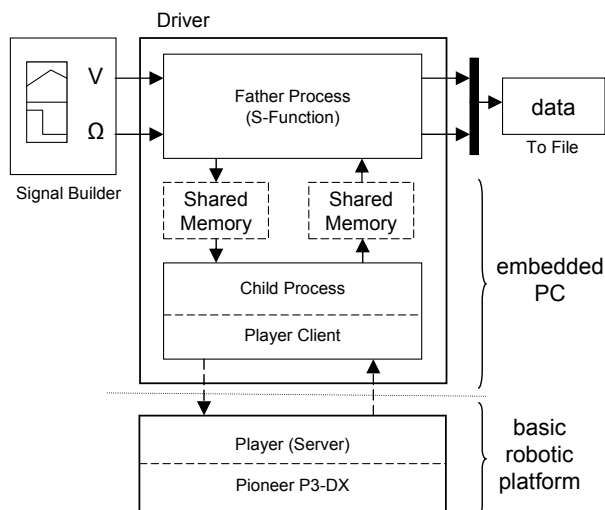


Fig. 8. Involved process in the design of the Simulink-driver for the low level access of the P3-DX platform.

Using the driver presented before, several control schemes are defined, which can be executed locally and remotely by only changing few settings in the target obtained by RTW.

### 3. Teleoperation control schemes

In this section, several teleoperation control schemes are described with the aim of showing the capabilities of the proposed architecture for such kind of applications. Therefore, a simple controller is suggested for tracking the linear and angular speeds of the robot, which was mentioned in the Introduction section as the middle level control (MLC). Then, two modes of operation are presented, where the controller is hosted externally to the robot's hardware (remote servocontroller) and directly executed inside (local servocontroller). Using the stated software/hardware architecture, the different teleoperation modes only require minor design modifications.

#### 3.1. Middle level control of the robot

The independent LLC for each wheel (PID) is in practice not enough to correctly control the angular and linear speeds of the robotic platform. The authors propose to use a servosystem controller as the Middle Level Controller that guarantees null error in steady state for constant references and low constant error for ramp references of angular and linear speed. First, a parametric identification of the robot is carried out by means of a linear and time invariant state-variable model  $G, H, C, D$ , where the state variables are related to both wheels speeds at different sample times, and the inputs are obtained from the desired linear and angular commands sent to the robot. The state-variable model is dependent of the LLC and eventually can replace it if the local PIDs of the wheels are disconnected.

As it is shown in Figure 9 the servosystem is designed to follow angular and linear speed commands in the robot. The controller has several degrees of freedom in its design, affecting its performance. The gains  $K_r$  and  $K_i$  are set using a Linear Quadratic Regulator (LQR) [Ogata, 1994], [Dutton, 1997] approach.

As was commented before, the P&S architecture makes possible the implementation of the MLC in two versions (local and remote) for telerobotics applications. In both cases, the motion reference or motion set-points are imposed by the remote centre.

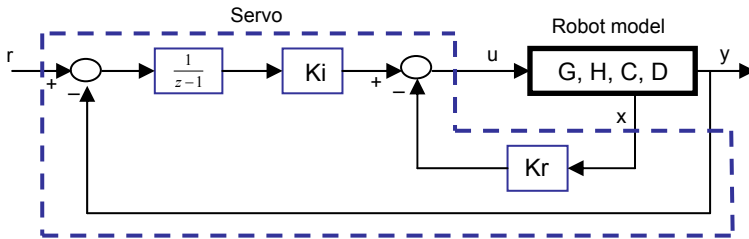


Fig. 9. Block diagram of the designed servosystem for robot velocities tracking.

**3.2. Telerobotics application of P3-DX through remote servocontroller**

In this alternative, the middle control level tasks (global velocities tracking of the mobile unit) are carried out in an external computer that serves as remote controller, see Figure 2. The commands given to the LLC to follow a given trajectory are established in the RC, the instantaneous error is calculated from the information received by the robot on-board sensors using the wireless communication channel. In the same way the resulting control outputs are sent to the mobile platform using the wireless channel. This control strategy is compatible with the Player/Stage architecture: the control algorithm in the RC is defined as a client  $C_{RC.S_R}$  of the server  $S_R$  in the P3-DX robot. A general diagram of the elements involved in this solution for control and communication can be seen in Figure 10.

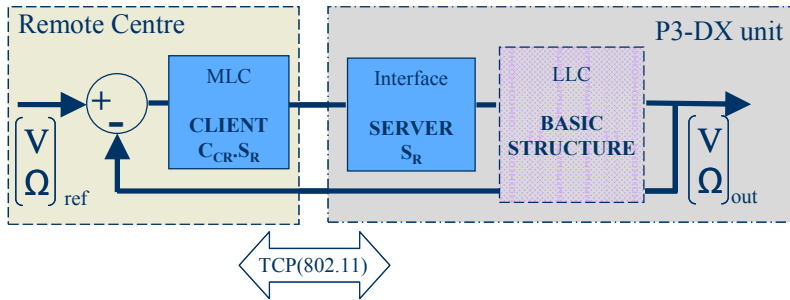


Fig. 10. Client-server structure based on Player/Stage for remote control for tele-operation of P3-DX units.

This is a clear example of how an external client links to the on-board server, no matter if they are implemented on different hardware platforms. Player/Stage satisfactorily handles these kinds of information transfer.

The remote servocontroller allows to minimize the hardware required in the robotic unit, however the system stability can be compromised due to well known problems derived from the wireless channel (delays, packet dropout, limited bandwidth,...).

**3.3. Telerobotics application of P3-DX through local servocontroller**

This alternative considers the implementation of both control levels LLC and MLC onboard the robot, see Figure 11.

The RC is only in charge of sending the commands for the desired movement. This task requires a non periodical updating time that generally is higher than the control sample time  $T_s$ . This is why a complementary client-server service based on sockets is implemented making easier the P3-DX teleoperation. In this way the client  $C_R.SS_{RC}$  in the robot unit is updated with the reference locations from the server  $SS_{RC}$  for local control set-points.

The primary advantage of this proposal is that the critical control tasks are executed in the robot client  $C_1.S_R$ , which means higher immunity to wireless communication channel. On the other hand, the  $C_R.SS_{RC}$  update, which is directly affected by the channel, can be asynchronous and longer.

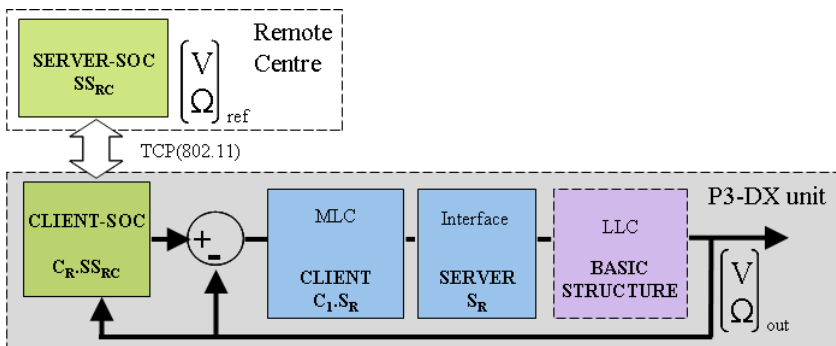


Fig. 11. Client-server structure, based on Player/Stage and sockets, for local control of teleoperated P3-DX.

**4. Experimental results**

Both hardware and software architectures described in this chapter are being used in different robotic applications inside the aforementioned COVE research project. In this section some experimental results are presented which support the pros and cons of the proposed architecture.

In Figure 12, a typical laboratory work session is shown, in which the human operator uses a PC as a remote center of the teleoperated P3-DX robot, which includes the designed hardware and software add-ons. As an example of the system capabilities, a path involving changes in angular and linear speeds is sent to a single robot. The teleoperated unit is

running the servo controller designed in section 3.1, and the sensor systems consists of odometry readings.

In this experiment the communication protocol consists of the UDP protocol (instead of TCP) allowing thus packet loss between the client and server. In the case that a command is lost, it is replaced with the one from previous instant. Such way of managing the packet loss is done in two directions: RC-robot (command sent), and robot-RC (odometry readings).

In a first experiment, the speed control for path following is implemented in local mode as it is stated in section 3.3. In this case, the remote center is only in charge of sending commands and supervising the result, but the MLC is hosted in the robot's CPU.

A second experiment is proposed, whose objective is to track the same commands as in the local mode, but the MLC is implemented in the remote center, following the idea presented in section 3.2. Therefore, only the LLC (PID) is executed in the robot, whilst the command generation, control of angular and linear speeds (MLC) and path monitoring are executed in the remote center.



Fig. 12. View of the laboratory set-up of telerobotics applications with P3-DX units.

In Figure 13 it is shown a temporal evolution of the linear speeds in the following cases:

- The linear speed performed using the local mode is shown in blue. As it can be observed, the speed (m/s) is following a trapezoid shape as it is expected in the experiment.
- The linear speed performed using the control in remote mode, and supposing that there is no packet loss is displayed in red. In the inferior part of the figure, where a zoom region of interest is presented, the channel communications includes two delays (two zero samples at the beginning), one for each direction in the channel.
- In yellow it is presented the linear speed of the robot in remote mode but including some percentage of packet loss and keeping, as was commented before, the previous command in that case. This experiment is performed with an approximate rate of 5% of packet dropout.

- The same experiment than the previous one but supposing a packet loss percentage around 50% is displayed in black. In this case it is expected that the trajectory performed by the robot derives away from the desired one.

In order to stress the risks of including a wireless channel inside a teleoperation process in remote mode, Figures 14 and 15 are shown. Figure 14 shows six realizations of the same trajectory followed by the robot in different control configurations. The three trajectories displayed in green belong to the MLC in local mode and, as it is observed, the differences between the three are negligible. The three paths displayed in red consists of the MLC in remote mode assuming a 5% of packet dropout, which as it is expected increase the differences between the paths.

The Figure 15 shows the same experiment but including a 50 % of packet dropout. In this case the channel is highly corrupted, which can invalidate the teleoperation capabilities, especially in the remote mode. The main problem of this level of packet loss is not that the target is not reached exactly as it is expected, but that the effects on the trajectory are highly random.

It must be remarked that the experiment has been performed using only the odometry readings from the robot. If more sensors are included, which allow an absolute positioning device (e.g. sonar, laser, vision...), the deviations from the desired path can be minimized.

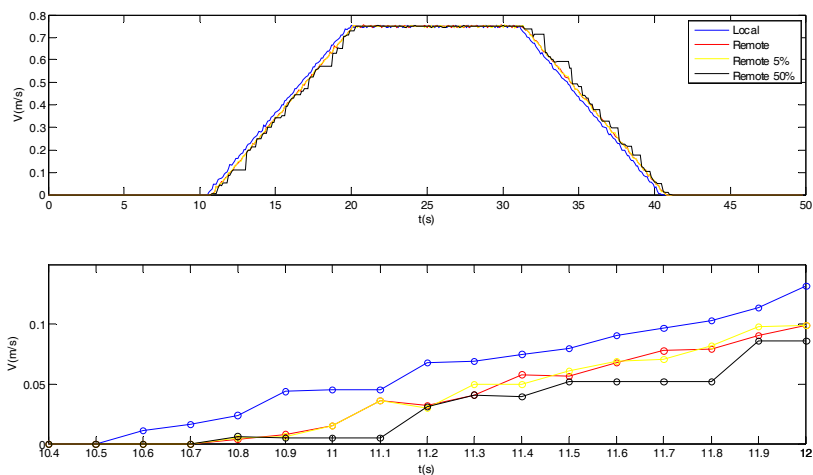


Fig. 13. Comparison of linear velocities from different tests of local control and remote control (UDP with packet dropout).

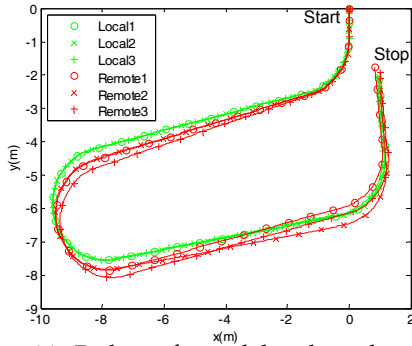


Fig. 14. Path performed by the robot in both local and remote mode (UDP with 5% packet dropout).

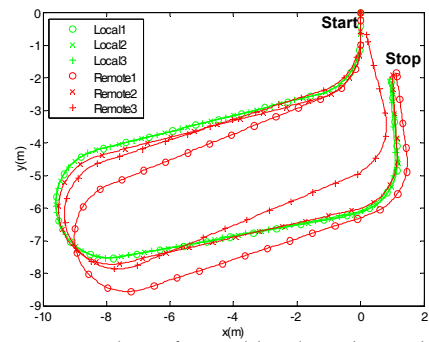


Fig. 15. Path performed by the robot in both local and remote mode (UDP with 5% packet dropout).

## 5. Conclusion

In this chapter the authors describe a solution for providing a basic commercial robot with the capabilities of being teleoperated. The proposed electronics architecture fits the requirements in terms of sensor processing, control and wireless communications needed in a telerobotic application.

From the hardware point of view, the key elements are the motherboard Via-Epia miniITX, and the wireless hardware included in the remote client WHR-HP-G54 and the one mounted on the robot WLI-TX4-G54HP.

From the software side, the more relevant feature is the high flexibility which is provided from the synergy between Matlab/Simulink/RTW with the client-server structure of the Player/Stage open software for the implementation of control applications.

Together with theoretical descriptions, examples of real telerobotics applications are shown. For that, the ad-hoc configured robotic platform is tested, working in both control modes: local and remote. Properties and drawbacks of using a wireless channel inside a control loop are remarked.

In practice, the prototype carried out by authors is being used as a real test bench for telerobotics and control applications, including robot cooperation problems in both educational and research fields.

## 6. References

- Angelo, J.A. *Robotics: a reference guide to the new technology*. IEEE Communications Magazine. Volume: 41, Issue: 12, page(s): 60-67. Dec. 2003 ISSN: 0163-6804. INSPEC Accession Number: 7950580
- Anvari, M. *Telemoring and remote telepresence surgery*. Chapter of the book "Robotics in Surgery: history, current and future applications". ISBN: 1-60021-386-3. Nova Science Publisher, Inc 2007

- Bambang, R. *Development of architectures for internet telerobotics systems*. Proceedings of the International Conference on Intelligent Unmanned System (ICIUS 2007), Bali, Indonesia, October 24-25, 2007, Paper No. ICIUS2007-B004
- Borenstein, J.; Everett, H.R.; Feng, L. (1996). *Where am I?. Systems and Methods for Mobile Robot Positioning*. Edited and compiled by J. Borenstein. March 1996. Electronic copies of this report in its entirety can be downloaded from <http://www-personal.umich.edu/~johannb/shared/pos96rep.pdf>
- Buffalo (2009). WLI-TX4-G54HP Datasheet available:  
[http://www.buffalotech.com/files/downloads/WLI-TX4-G54HP\\_DS.pdf](http://www.buffalotech.com/files/downloads/WLI-TX4-G54HP_DS.pdf)
- Chumsamutr, R. and Fujioka, T. *Development of Car-Following Model with Parameters Identified by Genetic Algorithm*. JSME Int Journal. Ser C. Mech Systems, Mach Elem Manuf. Journal Code: X0995A. ISSN: 1344-7653. VOL.46;NO.1;PAGE.188-196 (2003)
- Dutton, K; Thompson, S.; Barraclough, B. (1997). *The art of control engineering*. Addison-Wesley, 1997. ISBN- 0-201-17545-2
- Espinosa, F.; Awawdeh, A.M.H. (2006). *Focus on Robotics Research Book. Chapter: New strategies and control algorithms to reduce platoon oscillations in linear as well as non-linear trajectory*. Editor: John X. Liu, Nova Science Publishers, Inc. Hauppauge, New York. 2006
- Espinosa, F.; Salazar, M.; Bocos, A.; Valdés, F.; Iglesias, R. (2008). *Design and Implementation of a Communication Architecture based on Player/Stage for Telerobotics Operation of P3-DX units*. International Conference on Robotics and Automation -ICRA 2008-. Workshop: New Vistas and Challenges in Telerobotics. IEEE Catalog Number: CFP08RAA-CDR, ISBN: 978-1-4244-1647-9, ISSN: 1050-4729. Pages: 65-70. Pasadena, California, USA. May 19-23, 2008
- Gumaste, A.; Singhai, R. and Sahoo, A. *Intelligent Vehicular Transportation System (InVeTraS)*. Telecommunication Networks and Applications Conference, 2007. ATNAC 2007. Australasian. 2-5 Dec. 2007 Page(s):58 - 63. Digital Object Identifier 10.1109/ATNAC.2007.4665283
- Hespanha, J.P.; Naghshtabrizi, P. and Xu, Y. *A survey of recent results in Networked Control Systems*. Proceedings of the IEEE, vol. 95, no 1, pp. 138-162, January 2007
- Hokuyo (2009). <http://www.active-robots.com/products/sensors/hokuyo.shtml>
- Kato, S.; Tsugawa, S.; Tokuda, K.; Matsui, T. and Fujii, H. (2002). *Vehicle Control Algorithm for Cooperative Driving with Automated Vehicles and Intervehicle Communications*. IEEE Trans. on Intelligent Transportation Systems, Vol. 3, n° 3, pp 155-161
- MathWorks (2009). <http://www.mathworks.com/products/rtw/>
- Mehani, O.; Benenson, R.; Lemaignan, S. and Ernst, T. *Networking needs and solutions for road vehicles at Irnara*. ITST '07. 7th International Conference on ITS. ISBN: 1-4244-1178-5. Digital Object Identifier: 10.1109/ITST.2007.4295894
- MobileRobots (2009). <http://www.mobilerobots.com/>
- Ogata, K (1994). *Discrete-time control systems*. Prentice Hall, 2 edition. December, 1994. ISBN-10: 0130342815
- P3-DX (2009). <http://www.activrobots.com/ROBOTS/p2dx.html>
- PlayerStage (2009). The Player Project: <http://playerstage.sourceforge.net>
- RTAI (2009). RTAI-Linux Target HowTo.  
<http://www.mathworks.com/matlabcentral/files/10742/RTAI-TARGET-HOWTO.txt>

Via-Epia mini-ITX (2009). Datasheet available:

<http://www.via.com.tw/en/products/embedded/ProductSeries.jsp?serialNo=2>

Whitbrook, A. *Controlling the Pioneer P3-DX robots at CSiT*. University of Nottingham. April 2006





## **Remote and Telerobotics**

Edited by Nicolas Mollet

ISBN 978-953-307-081-0

Hard cover, 220 pages

**Publisher** InTech

**Published online** 01, March, 2010

**Published in print edition** March, 2010

Any book which presents works about controlling distant robotics entities, namely the field of telerobotics, will propose advanced technics concerning time delay compensation, error handling, autonomous systems, secured and complex distant manipulations, etc. So does this new book, Remote and Telerobotics, which presents such state-of-the-art advanced solutions, allowing for instance to develop an open low-cost Robotics platform or to use very efficient prediction models to compensate latency. This edition is organized around eleven high-level chapters, presenting international research works coming from Japan, Korea, France, Italy, Spain, Greece and Netherlands.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Felipe Espinosa, Marcelo Salazar, Daniel Pizarro and Fernando Valdes (2010). Electronics Proposal for Telerobotics Operation of P3-DX Units, Remote and Telerobotics, Nicolas Mollet (Ed.), ISBN: 978-953-307-081-0, InTech, Available from: <http://www.intechopen.com/books/remote-and-telerobotics/electronics-proposal-for-telerobotics-operation-of-p3-dx-units>

# **INTECH**

open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.