J. Automation & Systems Engineering 3-4 (2009): 208-225

Regular paper

Multi-Agent Architecture for Telelerobotics of Mobile Robot Manipulators

A. Hentout, B. Bouzouia, R. Toumi and Z. Toukal

This paper describes a multi-agent architecture for telerobotics of mobile robot manipulators. The architecture consists of three agents: Supervisory Agent, Mobile Robot Agent and Manipulator Robot Agent. The Supervisory Agent is composed of one Piloting Layer. The two other agents are composed, each one, of two layers (Piloting Layer and Operative Layer). The Piloting Layers are installed on an off-board PC. The Operative Layers are installed on the on-board PC of the robot. The autonomy of decision-making while performing a cooperative mission is employed at the local site (Piloting Layers of the agents) and, then, transmitted to the remote site (Operative Layers of the agents) for execution. The implementation of the architecture is done on the RobuTER/ULM mobile robot manipulator.

Keywords: Multi-agent architecture, Telerobotics, Mobile robot manipulators, *RobuTER/ULM*.

1. INTRODUCTION

Teleoperation refers to operation of a system over a distance [1]. The *operator* is the controlling entity, whereas the *teleoperator* refers to the system being controlled [2]. The main functions of a *teleoperation* system are the assistance of the operator [3] and the improvement of its safety [4] while performing complex tasks in hazardous environments.

Telerobotics is teleoperation applied to robotics, i.e., a robot controlled at a distance by an operator [5]. It is an active research field in robotics. A survey can be found in [6] and [7]. Many important telerobotic applications are in use today as space exploration [8], outer space [9], underwater [10], military applications [11], mining robots [4], inspection and maintenance of nuclear reactors [12], surgery [13], assistance to disabled [14] and elderly persons [15], Urban Search and Rescue (USAR) [16], etc. Most of works dealing with telerobotics investigates manipulators, or discusses teleoperation of mobile or a team of mobile robots. Little work is carried out in telerobotics of mobile robot manipulators.

In general, manipulators are remotely controlled to repair or replace some machine units [17]. Among first works is that developed by Goertz [18] to handle radioactive elements placed in a protected zone. Schaude [19] developed *Teleoperation System* (*TELOS*) allowing the operator to develop telerobot activities, done in three phases, with a graphical interface. In the first phase, all activities are prepared, verified in the next phase within a simulation, before the execution on a telemanipulator in the third phase. A system integrating sensory feedback from end-effector into the human-machine control loop was developed in [20]. The manipulator was called to follow contours of undefined object geometry which demands continuous adaptation of the robot path. Chatila [21] discusses a telerobotic mobile robot which accepts task commands from the ground operator and moves autonomously on Mars. The operator acts as a mission planner and supervisor. Lin

Division of Computer-Integrated Manufacturing and Robotics (DPR), Advanced Technologies Development Centre (CDTA), BP 17, Baba Hassen, Algiers 16[273, Algeria. {ahentout, bbouzouia, ztoukal}@cdta.dz
Laboratory of Robotics, Parallelism and Electro-energy (LRPE), University of Sciences and Technology Houari Boumediene (USTHB), BP [29, El Alia, Bab Ezzouar, Algiers 16111, Algeria. rtoumi@usthb.dz

[22], Bruemmer [23], Fong [24] and Ong [25] described telerobotic systems of mobile robots with multi-sensor feedback. For these systems, different degrees of cooperation are considered between the operator and the mobile robot.

Few papers only deal with telerobotics of mobile robot manipulators. Barnes [26] developed a telerobotic interface allowing the operator to communicate motions commands to their *K2A* mobile base attached to *PUMA560* manipulator. The system allows the robot to communicate to the operator its motion when performing autonomous collision avoidance. A system implementing a virtual reality interface, having client/server architecture, is developed in [27]. The client side contains the operator interface. The server side is a simulated predictive environment showing the navigation of the mobile base and the real time teleoperated motion of the manipulator. Howard and Park [28] developed an approach incorporating haptic feedback to guide human behavior in performing manipulation task using a 5 degrees of freedom (dof) manipulator attached to a *Pionner3AT* mobile base.

This paper describes the work carried out in the development of a multi-agent architecture for telerobotics of mobile robot manipulators. The next section gives an overview of previous works done in multi-agent telerobotics. Section three presents control modes for telerobotics. Section four presents the multi-agent architecture for telerobotics of mobile robot manipulators. The architecture and the kinematic analysis of the experimental mobile robot manipulator are described in section five. After that, the implementation of the architecture is explained in section six. Section seven studies the autonomy of decision-making while performing a cooperative mission. Finally, conclusions and future works are presented.

2. MULTI-AGENT TELEROBOTICS

Multi-agent telerobotics is a general discipline that has *Telerobotics* as a special case when the system consists of one remote robot (mobile, manipulator or mobile robot manipulator) [29]. Here, cooperation is provided by a multi-agent system. In addition, the decision-making is employed at the local site (multi-agent system) and, then, transmitted to the remote robot for execution.

Few architectures have only been proposed in the literature in the combined area of multi-agent telerobotics [30]. The *Multiple Agent Supervisory Control* (*MASC*) [31] is a supervisory control system for multiple robots that permits interaction at various levels in the perceptual processing. The robots work autonomously and when they are not able to carry out their task, they ask for assistance from the operator.

The *ACTor-based Robots and Equipments Synthesis System* (*ACTRESS*) [32] is a multi-agent system developed for heterogeneous agents and focuses on communication issues. The operator gives task commands to the robots which themselves coordinate how tasks are carried out. The operator can direct one robot or a group of robots at a time. Rybski [33] presented a control architecture of a team of miniature robots, called *Scouts*, that must use very low capacity *RF* communications due to their small size. The architecture is constrained by the limited computational capabilities of the robots, leading to a proxy-processing scheme enabling robots to use remote computers for their computing needs. Wegner and Anderson [34] developed an approach to blending teleoperation and autonomous control in behavior-based mobile robots for *USAR*, to explore damaged or collapsed urban structures in search of disaster victims. The central part of this approach involves the use of two agents running on each robot to effectively balance teleoperated and autonomous components. A mediation agent is used to appropriately blend the commands from a teleoperator with a robot autonomous processing, while an intervention recognition agent recognizes situations in which an operator should be informed that his intervention is required. Trivedi [35] designed a system intended to allow robotic units to recognize traffic collisions. This system uses teleautonomous robots that can form a perimeter around a collision. The remote operation provides very basic instructions to guide the robots to form perimeters around specific areas.

AUtonomous robot-Team Object MANipulation (*AUTOMAN*) [36] is a control hierarchy for object-based task-level control of a team of robots. It is used for a team of two free-flying robots for the task of manipulating a free-flying object. The operator does not need to focus on the actions of each robot what makes the robot group easier to control.

The above multi-agent telerobotic architectures range from *MASC*, which is almost autonomous, to *AUTOMAN*, which allows the user to specify the desired locations and orientations for manipulating an object, but not how to move between these locations. In addition, *MASC* allows the user to control only one robot at a time, while *AUTOMAN* requires the user to instruct all the robots at once. *ACTRESS* allows the operator to instruct one or more robots at a time [30]. The other systems provide various control modes, ranging from *direct control mode* to *autonomous control mode* (see next section).

3. CONTROL MODES FOR TELEROBOTICS

A human/robot cooperation allows human and robot to work together by letting them contributing according to their degree of expertise in different situations of the task. This concept not only considers how a robot might assist the operator but also how the operator might assist the robot. *Control modes for telerobotics* can be separated into four types ranging from "*no assistance provided to the operator by the robot*" to "*no assistance provided to the robot by the operator*" [25].

The choice of an adequate control mode depends on the task to be carried out. One of the most common reasons for using any control mode is to deal with time delay in teleoperation [37]. Other common reasons include safety and ease-of-use [38]. However, the problem remains of how to communicate to the operator exactly what the robot is doing so that its behavior can be overridden, as and when required [26].

3.1 Direct (Manual) mode

The operator specifies robot motion by continuous input by using a suitable input device to control the robot [39]. The robot takes no initiative except to stop when it recognizes that communication is broken-down [40].

3.2 Traded mode

This mode provides alternative control of the robot. The robot competence includes capabilities to choose its own path, to respond intelligently to the environment, and to accomplish local goals using a sequence of behaviors [25]. The operator assumes direct control in case of critical situations [22].

3.3 Shared mode

The operator and the robot can control different aspects of the system concurrently. This mode relieves the operator from controlling details and lets him concentrate on the goals of teleoperation [22].

3.4 Supervisory (Autonomous) mode

The operator performs high-level planning and monitors the robot execution. He may have to interrupt its execution in dangerous situation or help the robot executing its task. In this mode, the robot has the highest degree of autonomy and the entire robot motion is specified by computer input [22].

4. MULTI-AGENT ARCHITECTURE FOR TELEROBOTICS OF MOBILE ROBOT MANIPULATORS

The architecture (Fig. 1) consists of three agents: *Supervisory Agent*, *Mobile Robot Agent* and *Manipulator Robot Agent*. Each agent models a principal function of the robot and manages a different sub-system. *SA* is composed of one *Piloting* layer only, which is independent from the nature of the robot to be controlled. The two other agents, *MRA* and *ARA*, are composed, each one, of two layers: the *Piloting Layer* (*PLMRA*, *PLARA*) which is also independent from the nature of the robot to be controlled and, the *Operative Layer* (*OLMRA*, *OLARA*) which must be adapted to the software and hardware architecture of the robot to be controlled. For each layer corresponds a mechanism connecting the three capacities *Perception*, *Decision* and *Action* explained in more details in [41]. In addition, the *Supervision* capacity selects modules which result in the necessary behavior facing a given situation. The following are the basic functions of the different layers of the architecture:

- *Supervisory Agent, SA*: receives the mission to be carried out and, decides on its feasibility according to the availability and the state of the required equipments and resources of the robot (*Perception+Decision*). If the mission is accepted, *SA* distributes it on the *piloting layers* of the other agents for execution (*Action*).
- *Piloting Layer of the Mobile Robot Agent, PLMRA*: receives the remote environment information of the mobile base and obtains feedback (reports) from *OLMRA* on the execution of operations (*Perception*). Furthermore, it cooperates with the *piloting layer* of the other agent (*PLARA*) for decision-making according to the received information and the status of the other agents (*Decision*). At the end, *PLMRA* sends requests to *OLMRA* for execution (*Action*).
- *Piloting Layer of the Manipulator Robot Agent, PLARA*: builds a correct image on the environment of the manipulator, observes the environment by the vision system and receives information sensors and reports on the execution of the operations sent to *OLARA* (*Perception*). In addition, this layer cooperates with the piloting layer of the other agent (*PLMRA*) to build a cooperative operations plan for the execution of the assigned mission and, extracts useful and required information for the execution of the mission from captured images (images processing, extraction of 3D coordinates of objects, etc.) (*Decision*). Finally, it sends request to *OLARA* for execution and supervises its status (*Action*).
- *Operative Layer of the Mobile Robot Agent, OLMRA*: scans proprioceptif and exteroceptif sensors equipping the mobile base (*Perception*). In addition, it sends useful information to *PLMRA* to maintain a correct representation (up-to-date) of the environment. Moreover, it insures the local control of the mobile base by sending instructions to its actuators and executing the multiple control strategies (motion of the mobile base) offered by *PLMRA* (*Decision+Action*).
- *Operative Layer of the Manipulator Robot Agent, OLARA*: collects information of sensors equipping the manipulator and sends it to *PLARA* for processing (*Perception*). In

addition, this layer insures the local control of the manipulator by sending instructions to its actuators and executing the movements (motion of the manipulator) offered by *PLARA* (*Decision+Action*).

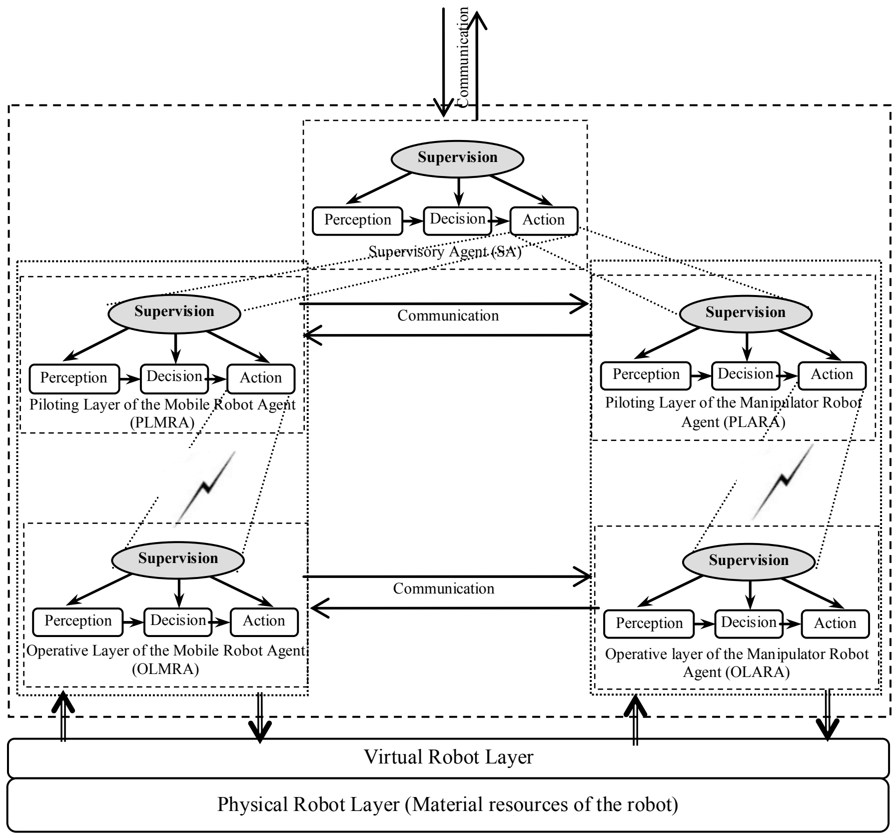

Fig. 1. Multi-agent architecture for telerobotics of mobile robot manipulators

The *Operative Layers* are directly connected to the *Virtual Robot Layer*. In its turn, this last layer is built on the material capacities of the *Physical Robot Layer*.

5. ARCHITECTURE OF THE EXPERIMENTAL ROBOTIC SYSTEM

The experimental robotic system, given by Fig. 2, consists of a *Local* (*Operator*) *site* and a *Remote site*, connected by wireless communication systems:

- *Local site*: includes an off-board *PC* under *Windows XP*, a wireless *TCP/IP* communication media, a wireless video reception system and input devices.
- *Remote site*: includes the *RobuTER/ULM* mobile robot manipulator, a wireless *TCP/IP* communication media and a wireless video transmission system.

5.1 Architecture of *RobuTER/ULM*

RobuTER/ULM is composed of a rectangular differentially driven mobile base on which is mounted a manipulator. The robot is controlled by an on-board *MMX* industrial *PC* and by four *MPC555* microcontroller cards communicating via a *CAN* bus. One *MPC555* card controls the mobile base. Two other cards control the manipulator. The last *MPC555* controls the effort sensor. The on-board *PC* is running under *Linux* 6.2 with *RTAI* layer 1.3. This layer interfaces *C/C++* application with that developed in *SynDEx* [42].

The mobile base has two driven wheels insuring its mobility and two free wheels to maintain its stability. The mobile base is equipped with a belt of 24 ultrasonic sensors, a laser measurement system at the front and an odometer sensor on each driven wheel. The manipulator is a six dof ultra-light manipulator (*ULM*) with two fingers electrical gripper. All of the joints are rotatable. The manipulator is equipped with incremental position sensor for each articulation and with a six dof effort sensor integrated on the gripper. The robot is also equipped with a monochrome *CCD* camera on the gripper (eye-in-hand camera) with an acquisition card. Images are directly transmitted to the off-board *PC* via a wireless video transmission system.

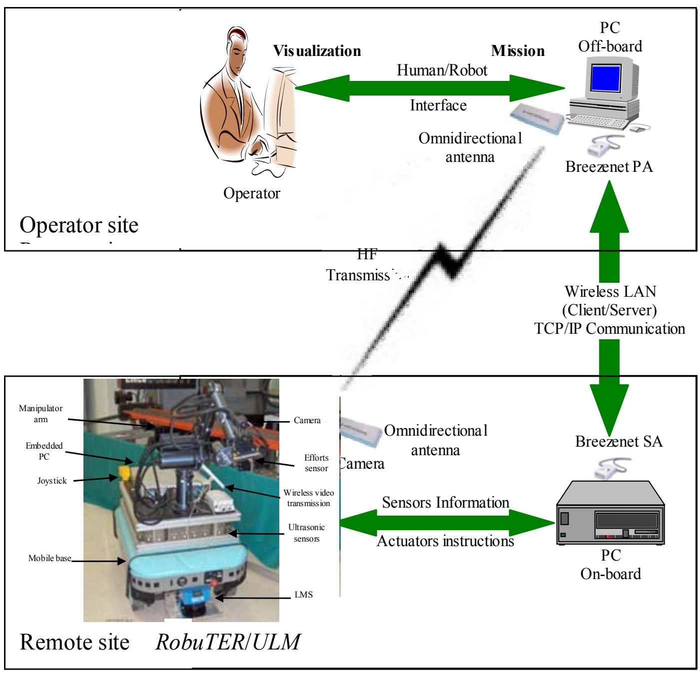

Fig. 2. Architecture of the experimental robotic system

5.2 Kinematic analysis of *RobuTER/ULM*

The kinematic analysis of *RobuTER/ULM* needs to focus on the following frames and transformation matrices (Fig. 3) [43].

- *Absolute frame* $R_A = (O_A, \vec{x}_A, \vec{y}_A, \vec{z}_A)$: fixed in plane of motion. Position coordinates and orientation of the robot are expressed according to $\vec{x}_A$, $\vec{y}_A$ and $\vec{z}_A$ axis.
- *Mobile base frame* $R_B = (O_B, \vec{x}_B, \vec{y}_B, \vec{z}_B)$: $\vec{y}_B$ is along the coaxial of the two driven wheels. $\vec{x}_B$ is the vertical with $\vec{y}_B$ and passes through the mid-point of the line connecting the centers of the two driven wheel. R_B is represented in R_A by the position of $O_B(X_B, Y_B, Z_B)$.

- *Manipulator frame* $R_M = (O_M, \vec{x}_M, \vec{y}_M, \vec{z}_M)$: assigned to the manipulator basis where are calculated the end-effector position coordinates and orientation angles. R_M is represented in R_B by the position of $O_M(X_M, Y_M, Z_M)$.
- $R_E = (O_E, \vec{x}_E, \vec{y}_E, \vec{z}_E)$: The end-effector frame.
- $R_k = (O_k, \vec{x}_k, \vec{y}_k, \vec{z}_k)$: The frame attached to the joint *k*.
- ${}^{M}T_E$: The transformation matrix defining R_E in R_M.
- ${}^{B}T_M$: This matrix defines R_M in R_B.
- ${}^{A}T_B$: This matrix defines R_B in R_A.
- ${}^{A}T_E$: The matrix defining R_E in R_A.
- ${}^{k-1}T_k$: The transformation matrix defining R_k in R_{k-1}.

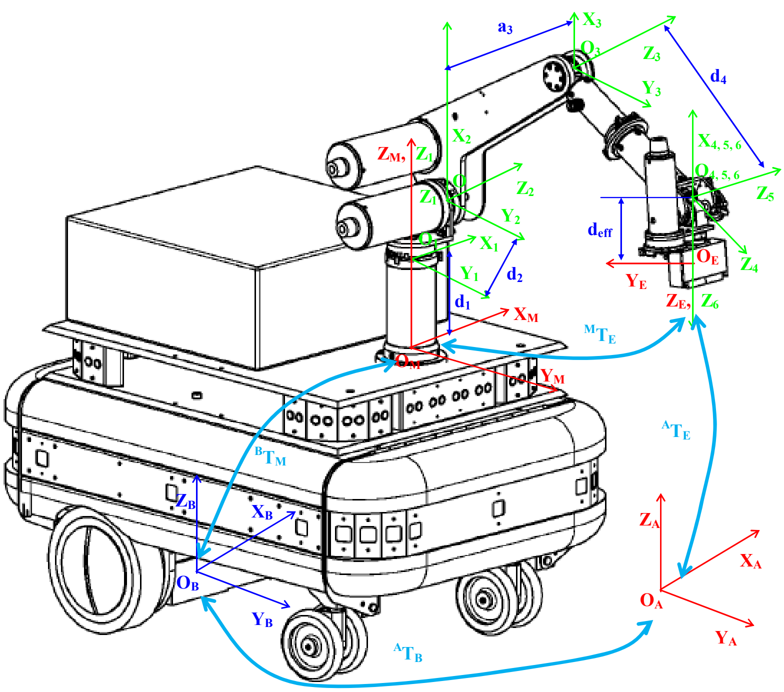

Fig. 3. Main reference frames of *RobuTER/ULM* and the corresponding transformation matrices

5.3 Kinematic analysis of the *ULM* manipulator

Position coordinates and orientation angles of the end-effector are calculated following the *Modified Denavit-Hartenberg* (*MDH*) representation [44]. The location of the end-effector is calculated in R_M by (1) where ${}^{M}T_2$ defines R_2 in R_M and ${}^{6}T_E$ defines R_E in R_6.

$$ {}^{M}T_E = {}^{M}T_1 * {}^{1}T_2 * {}^{2}T_3 * {}^{3}T_4 * {}^{4}T_5 * {}^{5}T_6 * {}^{6}T_E \quad (1) $$

${}^{M}T_1$, ${}^{6}T_E$ and ${}^{k-1}T_k$ (k=2… 6) are given by (2) [45]:

$$ {}^{k-1}T_k = \begin{bmatrix} \cos\theta_k & -\sin\theta_k & 0 & a_k \\ \cos\alpha_k.\sin\theta_k & \cos\alpha_k.\cos\theta_k & -\sin\alpha_k & -d_k.\sin\alpha_k \\ \sin\alpha_k.\sin\theta_k & \sin\alpha_k.\cos\theta_k & \cos\alpha_k & d_k.\cos\alpha_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2} $$

The *MDH* parameters α_k, d_k, θ_k, a_k and the joints limits of the manipulator are given in Table 1 [43].

TABLE 1

k	α_k (rad)	d_k (mm)	θ_k	a_k(mm)	Q_{Min}(°)	Q_{Max}(°)
1	0	d_1=290	θ_1	0	-95	96
2	π/2	d_2=108.49	θ_2	0	-24	88
3	-π/2	d_3=113	0	a_3=402	–	–
4	π/2	0	θ_3	0	-2	160
5	π/2	d_4=389	θ_4	0	-50	107
6	-π/2	0	θ_5	0	-73	40
7	π/2	d_{eff}=220	θ_6	0	-91	91

5.4 Kinematic analysis of the mobile base

Assuming that the robot moves on the plane, the kinematic model of the mobile base can be decided by three parameters: (X_B, Y_B, θ_B) the Cartesian coordinates and the orientation angle. During its motion, the mobile base calculates, by odometry, its position coordinates and orientation angle in real time (see Fig. 4) [43]:

- P_k, P_{k+1}: Current and next position of the mobile base.
- ω_{Lk}, ω_{Rk}: Left and right wheel velocity at position P_k.
- ϕ_k: Yaw angle at position P_k.
- R_k: Steering radius of the mobile base at position P_k.
- L: Half distance between the two driven wheels.
- r: The diameter of the driven wheels.

The Left and right elementary advances of the robot ΔD_{Lk}, ΔD_{Rk} are given by (3):

$$ \begin{pmatrix} \Delta D_{Lk} \\ \Delta D_{Rk} \end{pmatrix} = \begin{pmatrix} R_k + L \\ R_k - L \end{pmatrix} . \phi_k \tag{3} $$

The elementary rotation of the mobile base $\Delta\theta_k$ is given by (4):

$$ \Delta\theta_k = \frac{\Delta D_{RK} - \Delta D_{LK}}{2L} \tag{4} $$

The elementary displacement of the mobile base ΔD_k is given by (5):

$$ \Delta D_k = \frac{\Delta D_{Rk} + \Delta D_{Lk}}{2} \tag{5} $$

So, the next orientation angle is given by (6):

$$ \theta_{k+1} = \theta_k + \Delta\theta_k \tag{6} $$

Finally, the next position of the mobile base is given by (7):

$$\begin{pmatrix} Xb_{k+1} \\ Yb_{k+1} \end{pmatrix} = \begin{pmatrix} Xb_k \\ Yb_k \end{pmatrix} + \Delta D_k \cdot \begin{pmatrix} \cos(\theta_{k+1}) \\ \sin(\theta_{k+1}) \end{pmatrix} \quad (7)$$

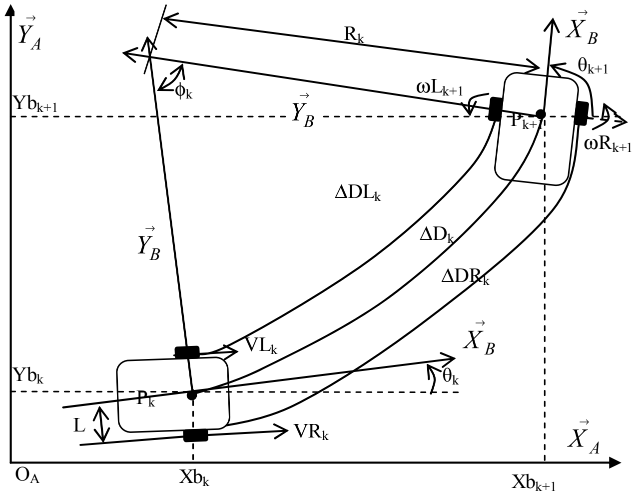

Fig. 4. Different parameters of the motion of the mobile base

5.5 Kinematic analysis of the mobile robot manipulator

The kinematic analysis of the mobile robot manipulator involves the interaction between its mobile base and its manipulator. The location of the end-effector is given in R_A by (8).

$$^{A}T_E = {^{A}T_B} * {^{B}T_M} * {^{M}T_E} \quad (8)$$

$^{A}T_B$ and $^{B}T_M$ are given by (9) and (10) respectively:

$$^{A}T_B = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & X_B \\ \sin\theta & \cos\theta & 0 & Y_B \\ 0 & 0 & 1 & Z_B \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

$$^{B}T_M = \begin{bmatrix} 1 & 0 & 0 & X_M \\ 0 & 1 & 0 & Y_M \\ 0 & 0 & 1 & Z_M \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

In case of *RobuTER/ULM*, Z_B=120mm, X_M=[27mm, Y_M=00mm and Z_M=520mm (Fig. 5) [43].

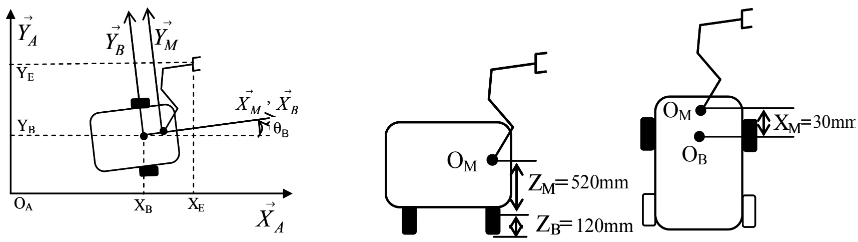

Fig. 5. Kinematic parameters of *RobuTER/ULM*

6. IMPLEMENTATION OF THE ARCHITECTURE

PLMRA, *PLARA* and *SA* are developed in *Borland C++ Builder 6.0*. They are installed on the off-board *PC*. *OLMRA* and *OLARA* are developed in *C/C++* and *SynDEx*. They are installed on the off-board *PC* of the robot. Each agent is implemented as a set of concurrent threads communicating via sockets using *TCP/IP* protocol. Furthermore, each agent has a *Knowledge Base* that describes its configuration.

6.1 Operative Layer of the Mobile Robot Agent (OLMRA)

This layer is composed of two parts:

1. The first part is developed in *SynDEx* and being carried out in the corresponding *MPC555* card. This part sends instructions to the actuators of the mobile base and reads odometer sensors.
2. The second one is developed in *C/C++* and being carried out on the on-board PC. It consists of seven threads:
   - *Configuration:* allows the operator to configure the agent (*ID*, *IP*, competences, etc.) and, to introduce information on agent knowledge and partial knowledge about its acquaintances and environment.
   - *Communication*: interfaces the agent with its acquaintances. It contains traditional communication functions (*Connect*, *Send*, *Receive*, etc.).
   - *Supervision*: activates threads that result in the necessary behavior facing a given situation and deactivates the others.
   - *Reading LMS Sensor*: scans continuously the serial port of the on-board *PC* (of the *LMS sensor*).
   - *Reading US Sensors*: scans the *US sensors*.
   - *Odometry*: reads the incremental encoders (ΔX, ΔY) installed on the two driven wheels of the mobile base and, calculates its current position and orientation angle (*X_Real*, *Y_Real*, *θ_Real*).
   - *Navigation*: The mobile base has to move to a *Target* (X_{Target}, Y_{Target}, θ_{Target}) while avoiding obstacles. This thread calculates velocities (V_R, V_L) according to the principle shown in Fig. 6 [46].

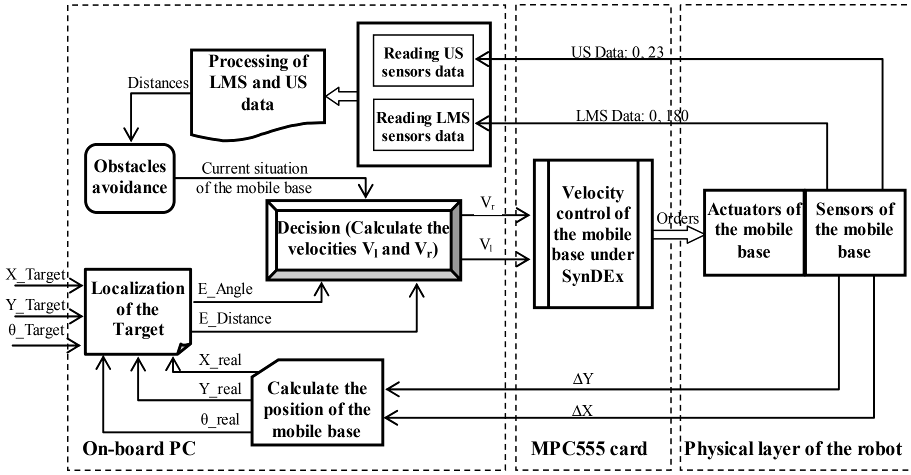

Fig. 6. Principle of motion of the mobile base

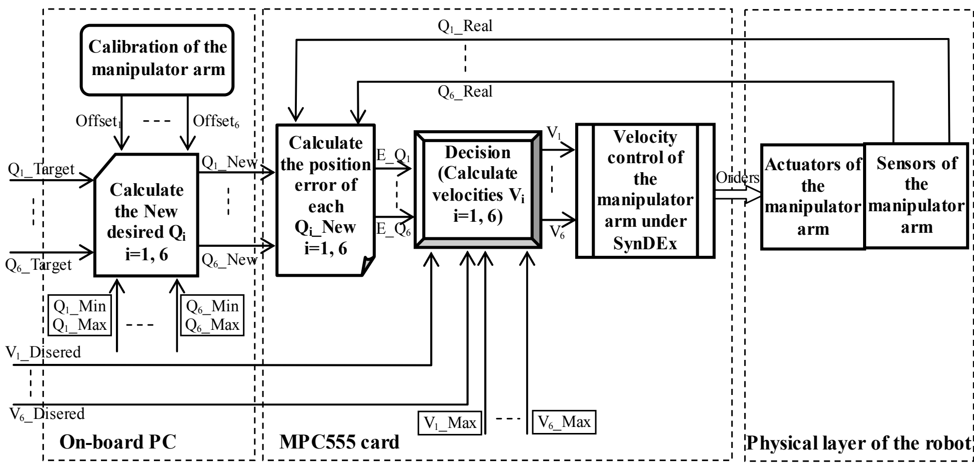

Fig. 7. Principle of motion of the manipulator

6.2 Operative Layer of the Manipulator Robot Agent (OLARA)

This layer is also composed of two parts:

1. The first one is developed in *SynDEx* and being carried out in the two corresponding *MPC555* cards. The role of this part is to send instructions to the actuators of the manipulator and to read positions and effort sensors.
2. The second part is developed in *C/C++* and being carried out in the on-board *PC*. This part is composed, in its turn, of seven threads:
   - *Configuration*.
   - *Communication*.
   - *Supervision*.
   - *Reading Effort Sensors/State Gripper*: reads the effort sensor and the state of the gripper (Opened or closed).

- *Reading Positions Sensors*: reads incremental sensors installed on the articulations of the manipulator.
- *Open/Close gripper*: opens or closes the gripper.
- *Movement*: calculates orders to be sent to the actuators of the manipulator to move to a *Target*(Q_1, …, Q_6) with velocities (V_1,…, V_6) according to the principle shown in Fig. 7.

6.3 Piloting Layer of the Mobile Robot Agent (PLMRA)

PLMRA consists of six threads:

- *Configuration*.
- *Supervision*.
- *Communication High-level*: communication with the other piloting layers.
- *Communication Low-Level*: communication with *OLMRA*.
- *Sensors Processing*: receives and processes *LMS*, *US* and *Odometer* sensors from *OLMRA*.
- *Position Calculation*: calculates the position of the mobile base on a plan relatively to any frame.

6.4 Piloting Layer of the Manipulator Robot Agent (PLARA)

PLARA consists of ten threads:

- *Configuration.*
- *Supervision.*
- *Communication High-Level*: communication with the other piloting layers.
- *Communication Low-Level*: communication with *OLARA*.
- *Sensors Processing*: receives *Positions* sensors, *Effort* sensors and the *State* of the gripper from *OLARA*.
- *Position Calculation*: tests either a given Cartesian position (x, y, z) belongs of the current workspace of the manipulator or not. In addition, it calculates the *Direct Kinematic Model* (*DKM*) and the *Inverse Kinematic Model* (*IKM*) of the manipulator.
- *Image Capturing*: displays the continuous video images (of the scene) delivered by the *CCD* camera of the robot.
- *Image Processing*: eliminates noise, binaries the image, detects contours and extracts forms corresponding to objects.
- *2D Extraction*: calculates the *2D* coordinates of the gravity center (u_i, v_i) of all the objects.
- *3D Extraction*: calculates the real coordinates (x, y, z) of objects by using two different images (captured from two different locations).

6.5 Supervisory agent

SA consists of the following threads:

- *Configuration.*
- *Communication.*
- *Supervision.*
- *Human/Robot Interface*: allows the operator to introduce the mission to be executed by the robot and to dialog with the architecture. In addition, it displays data of *Odometer*, *LMS*, *US*, *Effort* and *Joints positions* sensors, *State of the gripper*, *CCD* camera images, etc.
- *Mission Decision*: decides either the mission to be carried out by the robot is accepted or not. If the mission is accepted, it is sent to the piloting layers of the other agents for execution. Otherwise, this thread informs the operator of its incapacity to accomplish this mission.

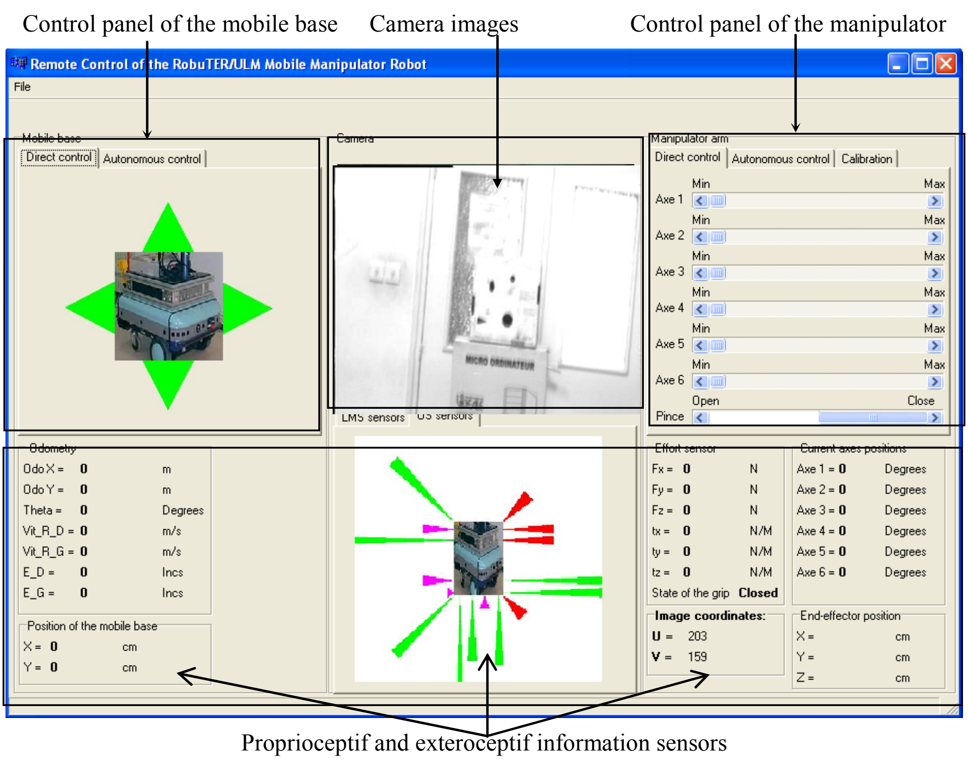

Fig. 8. The Human/Robot interface.

6.6 Human/Robot Interface

The operator interface, given by Fig. 8, is designed with the intention of making it easy for the operator to interact with the mobile robot manipulator. The operator can control the robot through two control panels [47]:

- *Control panel of the mobile base*: consists of four direction buttons for the *direct mode*. For the *autonomous mode*, the operator gives only a *Target*(X_{Target}, Y_{Target}, θ_{Target}) coordinates.
- *Control panel of the manipulator*: consists of seven scroll bars for the *direct mode*. For the *autonomous mode*, the operator specifies a *Target*(X_{Target}, Y_{Target}, Z_{Target}) Cartesian coordinates.

Sensors data are accumulated in the on-board *PC* of the robot. Then, data are transmitted to the off-board *PC* to be processed and displayed. The operator has all the data sensors displayed in a convivial and ergonomic manner as shown on Fig. 8 (the bottom of the Interface).

7. AUTONOMY OF DECISION-MAKING

The core thinking of modeling and controlling mobile robot manipulators using a multi-agent system is realizing cooperation between the manipulator, the mobile base and the sensors system.

7.1 Direct control mode

In this mode, the operator is in total control of the robot. There is no intelligence and hence, the operator has to rely on visual information to control the robot through the *direct control mode* panels of its interface.

7.2 Autonomous control mode

To show the autonomy of decision-making of *SA*, *PLMRA*, *PLARA*, *OLMRA* and *OLARA*, a cooperative mission is considered. It consists of a position-based control of the robot using the eye-in-hand camera. The working mission is to reach different positions (corresponding to different objects) by the end-effector of the robot. First of all, *SA* receives the mission and distributes it on *PLMRA* and *PLARA*. To reach the desired position (*Target*), it is necessary to capture *Left* and *Right* images. *PLARA* begins by reading the *Left* and *Right* positions from its *Knowledge Base*. After that, *PLARA* sends a request *Move Gripper(Left Position)* to *OLARA* to position the manipulator. Next, a second request *Move Gripper (Right Position)* is sent to *OLARA*. For each position, *PLARA* captures an image and carries out the necessary processing to extract *2D* coordinates (u_1, v_1) and (u_2, v_2) in the two images. Using these *2D* coordinates, *PLARA* extracts the *3D* coordinates (x, y, z) and transmits them to *SA* which decides on the feasibility of the mission.

If *Target* is not reachable, an error message is displayed to the operator (*Mission impossible*). Otherwise, a message is sent to *PLARA* to start the mission. Two cases are distinguished:

- *Target belongs of the current workspace of the robot*: The robot does not need its mobile base, only the motion of the manipulator is needed. *PLARA* calculates the different Q_i(i=1…6) corresponding to (x, y, z) and sends them to *OLARA*. This latter moves towards *Target* and, when arrived, informs *PLARA* that the end-effector is in *Target*.
- *Target is outside the current workspace of the robot*: *PLARA* sends a request to *PLMRA* to move towards a position where *Target* belongs of the new workspace of the manipulator. *PLMRA* calculates *New Position* (X_B, Y_B, θ_B) of the mobile base and sends a request to *OLMRA*. *OLMRA* moves towards *New Position* while avoiding obstacles. At the arrival, it informs *PLMRA* that the robot has reached the position successfully.

PLMRA informs *PLARA* that *Target* is now reachable. The process continues as in the first case.

The diagram of Fig. 9 shows the execution process.

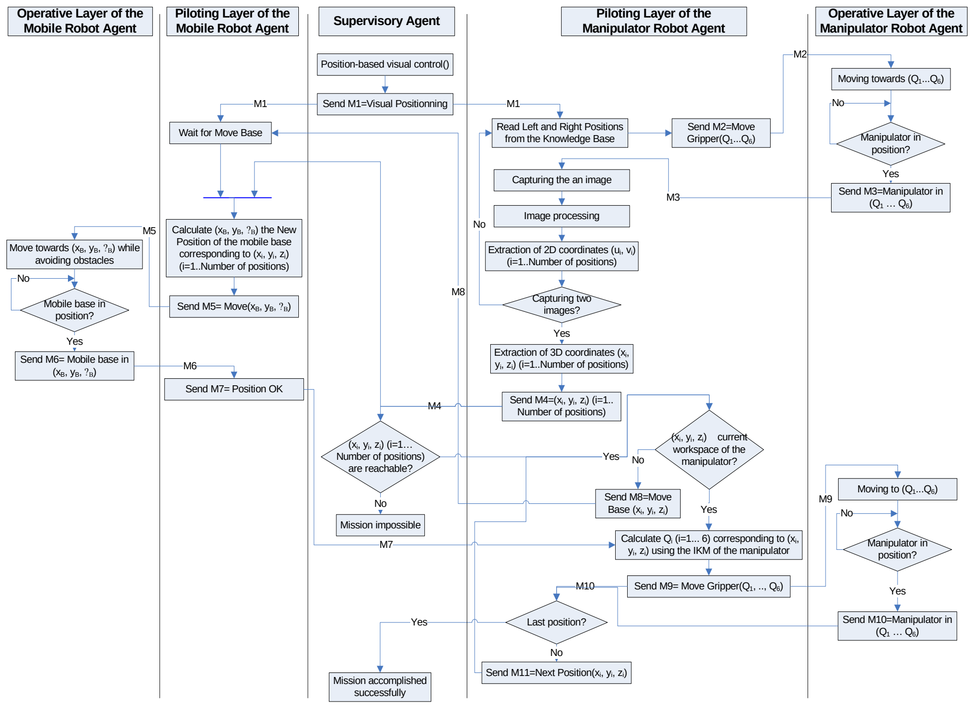

Fig. 9. Mission of position-based control of the mobile robot manipulator using the eye-in-hand camera.

8. CONCLUSION

This paper presented a multi-agent architecture for telerobotics of mobile robot manipulators. The architecture contains proposals to the agent topology and to the control structure. The mission to be carried out is decomposed to various independent sub-problems between the various agents of the multi-agent system.

Cooperation and synchronization between the sub-systems of the robot are insured by messages exchanged between the controlling agents of the architecture. In addition, decisions are made in the local site by the *Piloting Layers* of the agents (*PLMRA*, *PLARA*) and, then, transmitted to the *Operative Layers* (*OLMRA*, *OLARA*) for execution. The implementation of the architecture is done on the *RobuTER/ULM* mobile manipulator.

The validation of the other control modes non-considered in this work (*traded mode* and *shared mode*) will be presented in future works. In addition, the most important extension of this work is to implement a multi-agent mission specification and interpretation to automate, as possible, the piloting layers of the agents.

REFERENCES

[1] T. Fong, Ch. Thorpe, "Vehicle Teleoperation Interfaces, Autonomous Robots", 11(1), pp. 9-18, 2001.

[2] H. Filippi,, "Wireless Teleoperation of Robotic Arms", Master Thesis, Luleå University of Technology, Kiruna, Espoo-Finland, 2007.

[3] P. Batsomboon, S. Tosunoglu, D. W. Repperger, "A Survey of Telesensation and Teleoperation Technology with Virtual Reality and Force Reflection Capabilities", International Journal of Modeling and Simulation, 20(1), pp. 79-88, 2000.

[4] E. Duff, K. Usher, K. Taylor, C. Caris, "Web-Based Tele-Robotics Revisited", Australasian Conference on Robotics and Automation, Australia, 2007.

[5] http://books.nap.edu/openbook.php?recordid=4761page=304

[6] L. E. Parker, J. V. Draper, "Robotics Applications in Maintenance and Repair", Handbook of Industrial Robotics, 2nd Edition, 1998.

[7] K. Taylor, B. Dalton, Internet robots: A new robotics niche, IEEE Robotics and Automation Magazine, 7(1), pp. 27-34, 2000.

[8] E. Halberstam, L. Navarro-Serment, R. Conescu, S. Mau, G. Podnar, A. D. Guisewite, H. B. Brown, A. Elfes, J. M. Dolan, M. Bergerman, A Robot Supervision Architecture for Safe and Efficient Space Exploration and Operation, 10th International Conference on Engineering, Construction, and Operations in Challenging Environments: Earth & Space Conference, ASCE publisher, USA, 2006.

[9] G. Hirzinger, Multisensory shared autonomy and telek-sensor programming - key issues in space robotics, Robotics and Autonomous System, 11(1), pp. 141-162, 1993.

[10] D. Gallimore, A. Madsen, Remotely Operated Vehicles of the World. Ledbury, Herfordshire, England: Oilfield Publications Limited, 1994.

[11] D. C. Bentivegna, Kh. S. Ali, R. C. Arkin, T. Balch, Design and Implementation of a Teleautonomous Hummer, The SPIE Conference on Mobile Robots XII, USA, pp. 1[27-138, 1998.

[12] P. Desbats, P. Garrec, Y. Perrot, O. David, L. Gargiulo, G. Piolain, Y. Measson, Overview of Robotic Technologies for Nuclear Hazardous Environments, The 12th Topical Meeting on Robotics and Remote Systems for Hazardous Environments, USA, 2008.

[13] R. M. Satava, Telemanipulation, telepresence, and virtual reality for surgery in the year 2000, SPIE Telemanipulator and Telepresence Technologies, USA, pp. 162-171, 1995.

[14] Y. Rybarczyk, O. Ait Aider, Ph. Hoppenot, E. Colle, Remote control of a biomimetics robot assistance system for disabled persons, AMSE, special issue - bioengineering applied to handicap rehabilitation, pp. 47-56, 2002.

[15] A. Cesta, S. Bahadori, G. Cortellessa, G. Grisetti, M. V. Giuliani, L. Iocchi, G. R. Leone, D. Nardi, A. Oddi, F. Pecora, R. Rasconi, A. Saggese, M. Scopelliti, The RoboCare Project: Cognitive Systems for the Care of the Elderly, International Conference on Aging, Disability and Independence (ICADI), USA, 2003.

[16] J. Casper, R. Murphy, Workflow study on human-robot interaction in USAR, The International Conference on Robotics and Automation, USA, pp. 1997-2003, 2002.

[17] Y.Y. Liao, L.R. Chou, T.J. Horng, Y.Y. Luo, K.Y. Young, S.F. Su, Force Reflection and Manipulation for a VR-Based Telerobotic System, The NSC - Part A: Physical Science and Engineering, pp. 382-389, 2000.

[18] R. C. Goertz, Fundamentals of General-Purpose Remote Manipulators, Nucleonics, 10(11), pp. 36-45, 1952.

[19] H. F. Schaude, J. Fröhlich, R. Dillmann, Integration of Sensor Simulation for Telerobotic and Telepresence, The International Conference on Modelling, Simulation and Identification, Japan, 1994.

[20] P. Lim, J. Yang, N. Hildreth, W. Friedrich, Application of Shared Telerobotic Control for Contour Following Processes, The Australasian Conference on Robotics and Automation, Auckland, 2002.

[21] R. Chatila, R. Alami, G. Giralt, Task-level programmable intervention autonomous robots, In Metronics and Robotics I. K.-H. Robrock P. A. Mac-Conaill, P. Drews, Ed., pp. 77-87, 1991.

[22] I-.S Lin, F. Wallner, R. Dillmann, A Advanced Telerobotic Control System for a Mobile Robot with Multisensor Feedback, Intelligent Autonomous Systems (IAS-4), Germany, pp. 365-372, 1995.

[23] D. J. Bruemmer, D. D. Dudenhoeffer, J. L. Marble, Dynamic-Autonomy for Urban Search and Rescue, AAAI Mobile Robot Competition, Canada, 2002.

[24] T. Fong, Ch. Thorpe, Ch Baur, A Safeguarded Teleoperation Controller, International Conference on Advanced Robotics, Hungary, 2001.

[25] K. W. Ong, G. Seet, S. K. Sim, An Implementation of Seamless Human-Robot Interaction for Telerobotics, International Journal of Advanced Robotic Systems, 5(2), pp. 167-176, 2008.

[26] D. P. Barnes, M. S. Counsell, Haptic Communication for Remote Mobile Manipulator Robot Operations, American Nuclear Society, 8^{th} International Topical Meeting on Robotics and Remote System, USA, 1999.

[27] J. Aleotti, A. Melzi, S. Caselli, A virtual reality based system for programming mobile manipulation tasks, The 8^{th} International IFAC Symposium on Robot Control, Italy, 2006.

[28] A. M. Howard, C. H. Park, Haptically Guided Teleoperation for Learning Manipulation Tasks, Robotics: Science and Systems: Workshop on Robot Manipulation, USA, 2007.

[29] T. Urbancsek, F. Vajda, Internet Telerobotics for Multi-Agent Mobile Microrobot Systems – A New Approach, The INES International Conference on Intelligent Engineering Systems, Egypt, 2003.

[30] Kh. S. Ali, Multiagent Telerobotics: Matching Systems to Tasks. Ph.D. Thesis in Computer Science, Georgia Institute of Technology, 1999.

[31] J. Adams, R. Bajcsy, J. Kosecka, V. Kumar, R. Mandelbaum, M. Mintz, R. Paul, C. Wang, Y. Yamamoto, X. Yun, Cooperative material handling by human and robotic agents: Module development and system synthesis, The International Conference on Intelligent Robots and Systems, USA, 1995.

[32] H. Asama, A. Matsumoto, Y. Ishida, Design of an autonomous and distributed robot system: Actress, The International Workshop on Intelligent Robots and Systems, Japan, pp. 283-290,1989.

[33] P.E. Rybski, S.A. Stoeter, M. Gini, D.F. Hougen, N.P. Papanikolopoulos, Performance of a Distributed Robotic System Using Shared Communications Channels, IEEE Transactions on Robotics and Automation, pp. 713-727, 2002.

[34] R. Wegner, J. Anderson, Agent-Based Support for Balancing Teleoperation and Autonomy in Urban Search and Rescue, The International Journal of Robotics and Automation, 21(2), pp. 120-128, 2006.

[35] M. Trivedi, B. Hall, G. Kogut, S. Roche, Web-based teleautonomy and telepresence, The 45^{th} Optical Science and Technology Conference, San Diego, pp. 81-85, 2000.

[36] W. Dickson, R. Cannon Jr., Experimental Results of Two Free-Flying Robots Capturing and Manipulating a Free-Flying Object, International Conference on Intelligent Robots and Systems, USA, pp. 51-58, 1995.

[37] M. Stein, R. Paul, Operator interaction, for time-delayed teleoperation, with a behavior-based controller, The International Conference on Robotics and Automation, pp. 2[28-236, 1994.

[38] T. Sheridan, Telerobotics, Automation, and Human Supervisory Control, The MIT Press, Cambridge, Massachusetts, 1992.

[39] S. Skaar, C. Ruoff, Teleoperation and Robotics in Space, Institute of Aeronautics and Astronautics, Washington, D.C., USA, 1994.

[40] D. J. Bruemmer, J. L. Marble, D. D. Dudenhoeffer, M. O. Anderson, M. D. McKay, Mixed-Initiative Control for Remote Characterisation of Hazardous Environments, The 36th International Conference on System Sciences, Hawaii, 2003.

[41] A. Hentout, B. Bouzouia, Z. Toukal, Multi-agent Architecture Model for Driving Mobile Manipulator Robots, The International Journal of Advanced Robotic Systems, 5(3), pp. 257-269, 2008.

[42] http://www.syndex.org/

[43] A. Hentout, B. Bouzouia, R. Toumi, Z. Toukal, Agent-Based Coordinated Control of Mobile Manipulators, The International Conference on Systems and Processing Information (ICSIP'09), Algeria, 2009.

[44] W. Khalil, J. F. Kleinfinger, A new geometric notation for open and closed-loop robots, The International Conference on Robotics and Automation, USA, 1986.

[45] E. Dombre, W. Khalil, Modeling, Performance Analysis and Control of Robot Manipulators: Control Systems, Robotics and Manufacturing Series, Hermes Science Publishing, 2007.

[46] M. Kadri, O. Azouaoui, R. Kezerli, A. H. Hariz, Comportement d'évitement d'obstacles basé sur les réseaux de neurones et la logique floue pour un robot mobile autonome, The 4th International Conference on Computer-Integrated Manufacturing (CIP'2007), Setif, Algeria, 2007.

[47] A. Hentout, B. Bouzouia, R. Toumi, Z. Toukal, Multi-agent Remote Control of the RobuTER/ULM Mobile Manipulator Robot, 5th IEEE International Conference on Mechatronics (ICM'09), Spain, 2009.