Mobile Manipulation: A Case Study

A. HENTOUT[1], B. BOUZOUIA[2], I. AKLI[3] and R. TOUMI[4]

[1, 2, 3] *Division of Computer-Integrated Manufacturing and Robotics (DPR)*
Advanced Technologies Development Centre (CDTA)
BP 17, Baba Hassen, Algiers 16303
Algeria

[4] *Laboratory of Robotics, Parallelism and Electro-energy (LRPE)*
University of Sciences and Technology Houari Boumediene (USTHB)
BP 32, El Alia, Bab Ezzouar, Algiers 16111
Algeria

1. Introduction

Classically, manipulators consist of several links connected together by joints. The main purpose in using these robots is to manumit the human from tedious, arduous and repetitive tasks. Nevertheless, the limited dimensions of the links and the morphology of the fixed-base manipulators, create, therefore, limited accessible workspaces.

To support the development and the new application fields of manipulators, the locomotion had to be combined to the manipulation creating, thus, mobile manipulators. This kind of robots consists of coupling manipulation (represented by a manipulator) and locomotion (represented by a mobile base). The conventional structure of this type of robots is a manipulator mounted upon a mobile base. The mobility extends the workspace of the manipulator and increments its operational capability and flexibility(Sugar & Kumar, 1998).

Mobile manipulators allow the most usual missions of robotics that require both abilities of locomotion and manipulation. They have applications in many areas such as grasping and transporting objects, mining, manufacturing, forestry, construction, etc. Recently, target environment for for activity of such robots has been shifting from factory environment to human environment (Nagatani et al., 2002) (offices, hospitals, homes, assistant for disabled and elderly persons, etc.) because they are particularly well suited for human-like tasks (Alfaro et al., 2004).

However, the motion study of these robots is different and more difficult than that of manipulators. Firstly, combining a mobile base and a manipulator creates redundancy. Secondly, the mobile base has a slower dynamic response than the manipulator. Thirdly, the mobile base is often subject to non-holonomic constraints while the manipulator is usually unconstrained. Finally, the task to be carried out by the robot must be decomposed into tiny movements to be executed by the manipulator and large movements to be carried out by the mobile base (Chen et al., 2006).

In recent years, there are a number of researchers studying mobile manipulators control. These studies led to different approaches.

One of the general approaches is to consider the locomotion as extra joints of the manipulator (Nagatani et al., 2002). In this case, the mobile manipulator is regarded as a redundant robot where the redundancy is introduced by the motion of the mobile base (Sasaki et al., 2001). Erden and colleagues (Erden et al., 2004) describe a multi-agent control system to a service mobile manipulator that interacts with human during an object delivery and hand-over task in two dimensions. The identified agents of the system are controlled using fuzzy control. The membership functions of the fuzzy controller are tuned by using genetic algorithms. The authors in (Chen et al., 2006) propose a three-level neural network-based hierarchical controller. The bottom-level controls each joint motor independently. The middle-level consists of a neural network and two sub-controllers. The high-level is a task-planning unit that defines the desired motion trajectories of each degree of freedom (dof). Colle et *al*. (Colle et al., 2006) propose a multi-agent system for controlling their mobile manipulator *ARPH*. For each articulation is affected a reactive agent that realize in parallel a local task without a priori knowledge on the actions of the other agents. Each agent computes the current position of the end-effector and attempts by tiny local movements to match that position with the desired one.

The other type of approaches controls separately the mobile base and the manipulator neglecting the dynamic interaction between the two sub-systems. Such strategies are appropriate when the coupled dynamics is not significant (ex. when the robot moves at low speed) (Chen et al., 2006). The authors in (Waarsing et al., 2003) implement a behaviour-based controller over a mobile manipulator to make it able to open a door. The locomotion control system, the manipulator control system and the sensor systems cooperate in order to realize such a behaviour. Petersson et *al*. (Petersson et al., 1999) propose an architecture that enables the integration of the manipulator into a behaviour-based control structure of the mobile base. This architecture combines existing techniques for navigation and mobility with a flexible control system for the manipulator.

The robot, as human, must have the ability to obtain information about its environment in order to achieve each step of the manipulation task. The most important sensor which provides rich and varied information on the environment is the vision sensor (the camera) (Trabelsi et al., 2005). Based on hand-eye relation, visual servo system has two types of camera configuration (*i*) Eye-in-hand configuration and (*ii*) Eye-to-hand configuration (Flandin et al., 2000). The manipulator behaves as a hand and the camera as its eye. The camera is said as Eye-in-hand when rigidly mounted on the end-effecter. Here, there exists a known, often constant relationship between the position of the camera and that of the end-effecter. The camera is said as Eye-to-hand when it observes both of the robot and the (Muis & Ohnishi, 2005). Visionbased servoing schemes are flexible and effective methods to control robot motion from camera observations (Hutchinson et al., 1996). Many applications in vision-based robotics, such as mobile robot localization (Blaer & Allen, 2002), object grasping (Muis & Ohnishi, 2005) (Janabi-Sharifi & Wilson, 1998) and manipulation (Trabelsi et al., 2005), handling and transporting objects from one place to another (Trabelsi et al., 2005), navigation (Winter et al., 2000), etc.

This chapter highlights several issues around mobile manipulation in indoor environments. The first aspect consists of planning a coordinated trajectory for the non-holonomic mobile base and the manipulator so that the end-effector of the robot can be as near as possible, from a predefined operational trajectory. The second aspect deals with a position-based servoing control of mobile manipulators by using an eye-in-hand camera and a *LMS* sensor. These applications are developed within the framework of control architecture of such robots while taking into account the constraints and difficulties mentioned above. The architecture consists of a multi-agent system where each agent models a principal function and manages a different sub-system of the robot. The unified models of the mobile manipulator are derived from the sub-models describing the manipulator and the mobile base. These applications are considered in the case of the *RobuTER/ULM* mobile manipulator of the *Division of Computer-Integrated Manufacturing and Robotics* of the *Advanced Technologies Development Centre*.

The second section of the chapter describes the hardware and the software architecture of the experimental robotic system. Section three explains the multi-agent architecture proposed to control mobile manipulators. Section four describes the implementation of the control architecture. The agents are implemented as a set of concurrent threads communicating by *TCP/IP* sockets. In addition, the threads of each agent communicate by shared variables protected by semaphores. The autonomy of decision-making and the cooperation between the agents are presented in section five through two problems. The first one focuses on trajectory planning and control for mobile manipulators. The end-effector of the robot has to follow a predefined operational trajectory (given by a set of Cartesian coordinates (x, y, z)) while the mobile base avoids obstacles present in the environment. The second part of the experiments, in order to give the robot the ability to manipulate in an indoor environment, deals with position-based servoing control of mobile manipulators using a single camera mounted at its end-effector (eye-in-hand camera) and a *LMS* sensor. Conclusions and future works are presented at the end of the chapter.

2. Architecture of the experimental

The experimental robotic system, given by Fig. 1, consists of a *Local* (*Operator*) *site* and a *Remote site*, connected by wireless communication systems:

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

2.1 Architecture of the *RobuTER/ULM* mobile manipulator

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. The on-board *PC* is running under *Linux* 6.2 with *RTAI* layer 1.3. This layer interfaces *C/C*++ application with that developed under *SynDEx* (http://www.syndex.org). The first *MPC555* card controls the mobile base. The second and the third control the first three and the last three joints of the manipulator. The last *MPC555* controls the effort sensor.

The mobile base has two driven wheels ensuring 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-finger 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 placed on the gripper (eye-in-hand camera) with an acquisition card. The resolution of the camera is 352*240 pixels. Images are directly transmitted to the off-board *PC* via the wireless video transmission system. The camera is maneuverable enough to explore the environment of the robot due to the six dof of the manipulator.

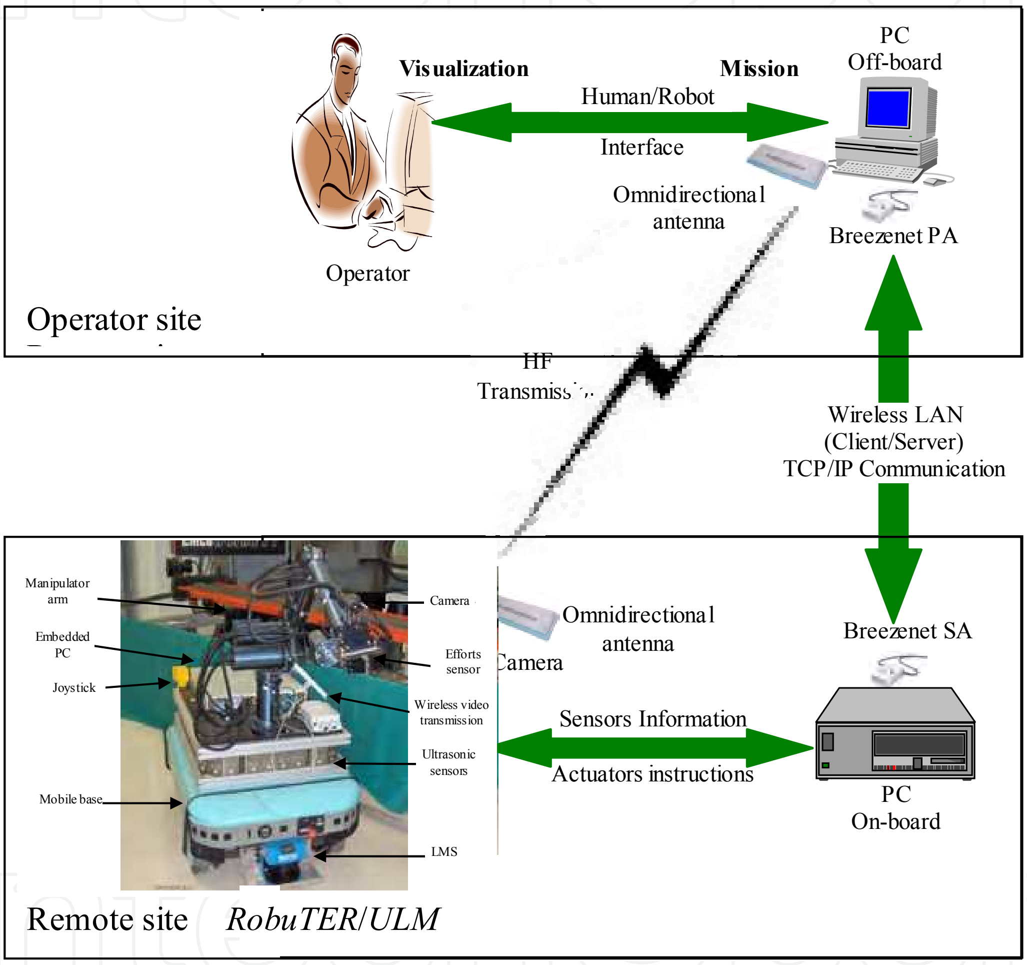


Fig. 1. Architecture of the experimental robotic system

2.2 Kinematic analysis of *RobuTER/ULM*

2.2.1 Main reference frames

The kinematic analysis of the robot needs to focus on the following main reference frames and transformation matrices (Fig. 2) (Hentout et al., 2009a):

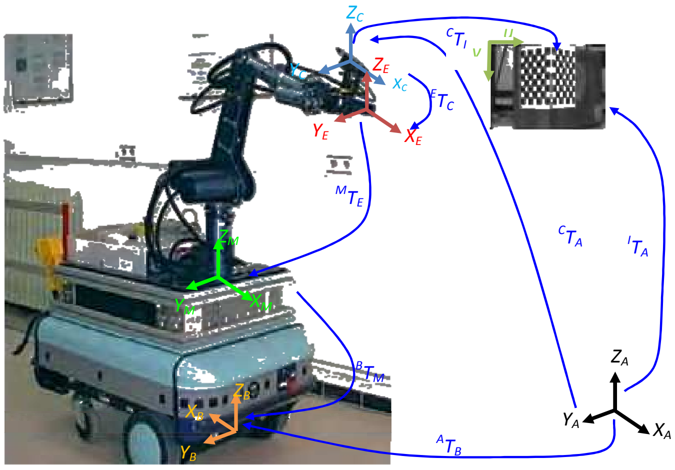


Fig. 2. Reference frames of the *RobuTER/ULM* and the transformation matrices

- $R_A = (O_A, \overrightarrow{x_A}, \overrightarrow{y_A}, \overrightarrow{z_A})$: The absolute reference frame.
- $R_B = (O_B, \overrightarrow{x_B}, \overrightarrow{y_B}, \overrightarrow{z_B})$: The mobile base reference frame.
- $R_M = (O_M, \overrightarrow{x_M}, \overrightarrow{y_M}, \overrightarrow{z_M})$: The manipulator reference frame.
- $R_E = (O_E, \overrightarrow{x_E}, \overrightarrow{y_E}, \overrightarrow{z_E})$: The end-effector reference frame.
- $R_C = (O_C, \overrightarrow{x_C}, \overrightarrow{y_C}, \overrightarrow{z_C})$: The camera reference frame.
- $R_I = (O_I, \overrightarrow{u_I}, \overrightarrow{v_I})$: The image reference frame.
- ^MT_E: The transformation matrix defining R_E in R_M. It corresponds to the *Kinematic Model* of the manipulator(see 2.3.2).
- ^BT_M: This matrix defines R_M in R_B(see 2.3.4).
- ^AT_B: This matrix defines R_B in R_A(see 2.3.4).
- ^AT_E: is the matrix defining R_E in R_A(see 2.3.4).
- ^IT_C: The camera intrinsic parameters matrix (see 4.3.1).
- ^CT_A: The camera extrinsic parameters matrix (see 4.3.1).
- ^IT_A: The camera projection matrix (see 4.3.1).
- ^ET_C: The Camera/Gripper transformation matrix (see 4.3.2).

2.2.2 Kinematic analysis of the *ULM* manipulator

The position coordinates and orientation angles of the end-effector are calculated in R_M by (1) following the *Modified Denavit-Hartenberg* (*MDH*) representation (Khalil & Kleinfinger, 1986) where $^MT_2$ defines R_2 in R_M, $^{k\text{-}1}T_k$ (k=3… 6) defines R_k in $R_{k\text{-}1}$ and ^6T_EdefinesR_E in R_6.

$$^MT_E = {}^MT_2 * {}^2T_3 * {}^3T_4 * {}^4T_5 * {}^5T_6 * {}^6T_E \quad (1)$$

$^MT_2$, ^6T_E and $^{k\text{-}1}T_k$ are given by (2) (Dombre & Khalil, 2007):

$$^{k-1}T_k = \begin{bmatrix} \cos\theta_k & -\sin\theta_k & 0 & a_k \\ \cos\alpha_k * \sin\theta_k & \cos\alpha_k * \cos\theta_k & -\sin\theta_k & -d_k * \sin\alpha_k \\ \sin\alpha_k * \sin\theta_k & \sin\alpha_k * \cos\theta_k & \cos\alpha_k & d_k * \cos\alpha_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

The different *MDH* parameters α_k, d_k, θ_k, a_k and the joints limits of the *ULM* manipulator are given in Table 1 (Hentout et al., 2009a).

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	$\pi/2$	d_2=108.49	θ_2	0	-24	88
3	$-\pi/2$	d_3=113	0	a_3=402	–	–
4	$\pi/2$	0	θ_3	0	-2	160
5	$\pi/2$	d_4=389	θ_4	0	-50	107
6	$-\pi/2$	0	θ_5	0	-73	40
7	$\pi/2$	d_{eff}=220	θ_6	0	-91	91

Table 1. The *MDH* parameters and the joints limits of the *ULM* manipulator

2.2.3 Kinematic analysis of the mobile base

Assuming that the robot moves on the plane, the kinematic model of the non-holonomic mobile base can be decided, in R_A, by three parameters: X_B, Y_B and 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 as shown in (Hentout et al., 2009a).

2.2.4 Kinematic analysis of the mobile manipulator

It involves the interaction between the mobile base and the manipulator. The location of the end-effector is given in R_A by:

$$ {}^{A}T_E = {}^{A}T_B * {}^{B}T_M * {}^{\mathrm{M}}T_{\mathrm{E}} \tag{3} $$

${}^{A}T_B$ and ${}^{B}T_M$ are given by (4) and (5) respectively. (X_B, Y_B, Z_B) are the Cartesian coordinates of O_B in R_A and (X_M, Y_M, Z_M) are the Cartesian coordinates of O_M in R_B.

$$ {}^{A}T_B = \begin{bmatrix} \cos\theta_B & -\sin\theta_B & 0 & X_B \\ \sin\theta_B & \cos\theta_B & 0 & Y_B \\ 0 & 0 & 1 & \mathrm{Z}_B \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4} $$

$$ {}^{B}T_M = \begin{pmatrix} 1 & 0 & 0 & X_M \\ 0 & 1 & 0 & Y_M \\ 0 & 0 & 1 & \mathrm{Z}_M \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{5} $$

For *RobuTER/ULM*, as shown in Fig. 3, Z_B=120mm, X_M=30mm, Y_M=0mm and Z_M=520mm (Hentout et al., 2009a).

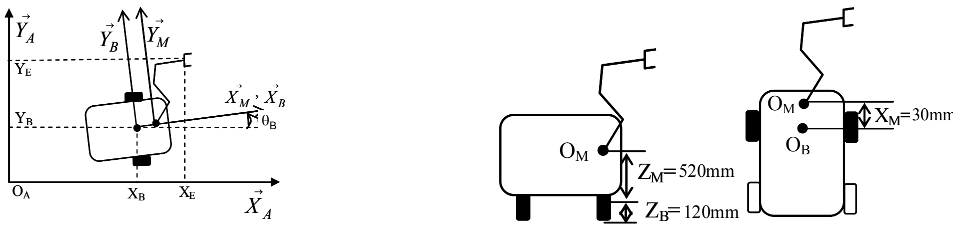


Fig. 3. Kinematics parameters of *RobuTER/ULM*

3. Multi-agent control architecture of mobile manipulators

Fig. 4 shows the proposed multiagent control architecture of mobile manipulators as proposed in (Hentout et al., 2008).The architecture consists of six agents: *Supervisory Agent* (*SA*), *Local Mobile Robot Agent* (*LMRA*), *Local Manipulator Robot Agent* (*LARA*), *Remote Mobile Robot Agent* (*RMRA*) and *Remote Manipulator Robot Agent* (*RARA*). Each agent models a principal function of the mobile manipulator and manages a different sub-system. In addition, for each agent corresponds a mechanism connecting the four capacities *Supervision, Perception*, *Decision* and *Action* explained in more details in (Hentout et al., 2008). The *Supervision* capacity is a virtual entity that select modules which result in the necessary behaviour facing a given situation.

The following are the basic functions of the architecture agents (Hentout et al., 2009b):

- *SA, Supervisory Agent*: *SA* receives the mission to be executed, decides on its feasibility according to the status and the availability (*Perception* + *Decision*) of the required equipments and resources of the robot (sensors, mobile base, manipulator, camera, etc.). If the mission is accepted, *SA* distributes it on on the corresponding agents for execution (*Action*).
- *LMRA, Local Mobile Robot Agent*/*LARA*, *Local Manipulator Robot Agent*: It receives the remote environment information of the mobile base/manipulator in order to build an up-to-date image on the environment where the robot evolves and, obtains feedback (reports) from *RMRA/RARA* on the execution of operations (*Perception*). In addition, the agent cooperates with the other agents (*LARA/LMRA*, *VSA*) in order to make a decision (*Decision*) according to the received information (sensors information, reports, etc.) and the status of the other agents of the architecture. At the end, it sends requests to *RMRA/RARA* for execution (*Action*).
- *VSA, Vision System Agent*: This agent observes the environment of the robot (*Perception*) by the vision system (the camera installed on the robot) and extracts useful and required information for the execution of the mission (*Decision* + *Action*) from captured images (images processing, localization and recognition of objects, etc.).
- *RMRA, Remote Mobile Robot Agent/RARA, Remote Manipulator Robot Agent:* This agent scans the various proprioceptif and exteroceptif sensors equipping the mobile base/manipulator (*Perception*) and sends the useful information to *LMRA/LARA* in order to maintain a correct representation of the environment. In addition, this agent ensures the local control of the mobile base/manipulator by sending instructions to its actuators and executing the multiple control strategies (navigation of the mobile base/motion of the manipulator) offered by *LMRA/LARA* (*Decision* + *Action*).

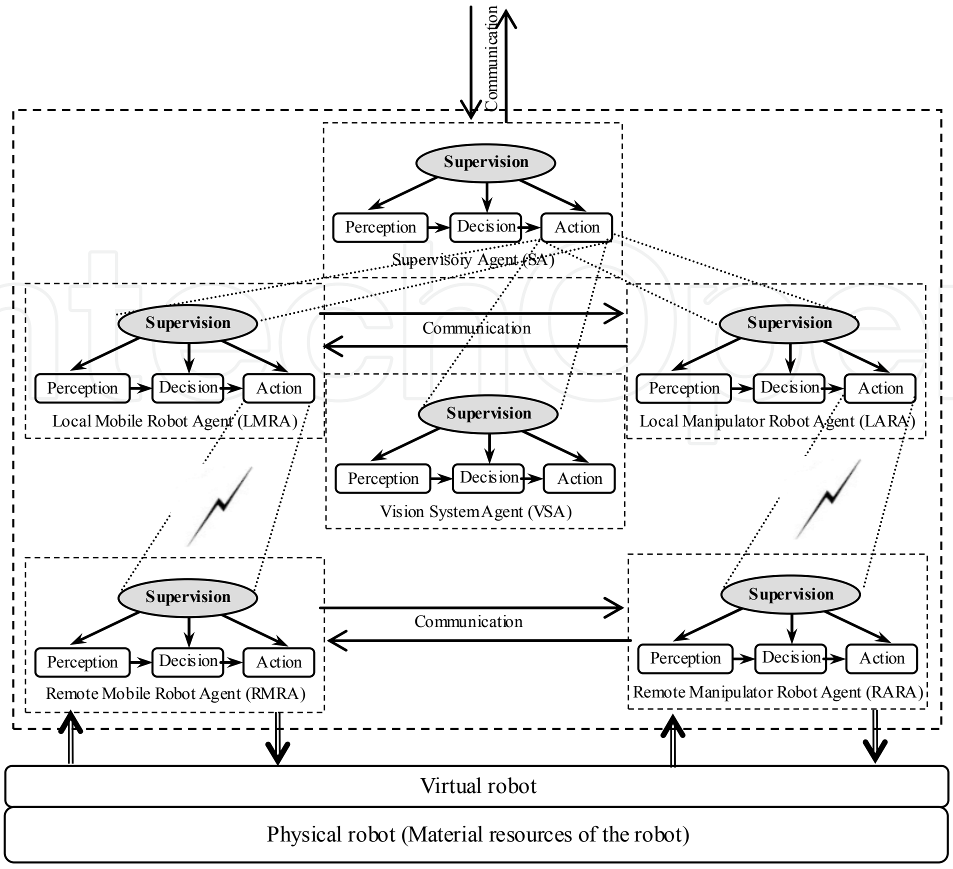


Fig. 4. Multi-agent control architecture

4. Implementation of the control architecture

The agents must be able to respond to asynchronous and external events, and to deal with requests, as soon as possible, according to the dynamics of the robot. Consequently, each agent is implemented as a set of concurrent communicating entities (a set of threads) executing autonomously and in parallel.

The agents communicate by sockets using *TCP/IP* protocol. Furthermore, *semaphores* are used to protect the access to the shared variables between the threads of the agent. *P*(*Variable*) to lock and *V*(*Variable*) to unlock the access to these variables. In addition, each agent has a *Knowledge Base* that describes its configuration. More details on the implementation of the multi-agent control architecture can be found in (Hentout et al., 2009c) (see Fig. 5 for the legend of the next figures of this section).

LMRA, *LARA*, *VSA* and *SA* agents are developed in *Visual studio C# 2008* and installed on the off-board *PC*. *RMRA* and *RARA* agents are installed on the off-board *PC*. They are developed in *C/C++* and *SynDEx*.

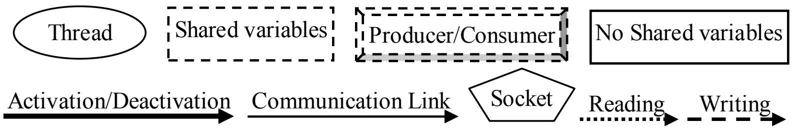


Fig. 5. Legend of the different components of the agents

4.1 Supervisory Agent

The *Supervisory Agent* (Fig. 6) consists of the following threads:

- *Configuration:* This thread allows the operator to configure the agent (ID, port, competences, etc.) and to introduce all the information on the knowledge of the agent, and the partial knowledge about its acquaintances and its environment.
- *Communication*: it interfaces the agent with its acquaintances. It contains traditional communication functions as *Connect*, *Disconnect*, *Send* and *Receive.*
- *Supervision*: Its role is to activate threads that result in the necessary behavior (facing a given situation) and deactivate the others.
- *Human/Robot Interface*: it displays data of odometer, LMS, US, effort and joints positions sensors, the state of the gripper, CCD camera images, etc. In addition, it allows the operator to introduce the mission to be executed by the robot and to dialog with the control architecture.
- *Mission Decision*: This thread decides either the mission to be carried out by the robot is accepted or not. For this aim, it checks the availability and status of all the required resources. If the mission is accepted, it is sent to the other agents for execution. Otherwise, this thread informs the operator of its incapacity to accomplish this mission.

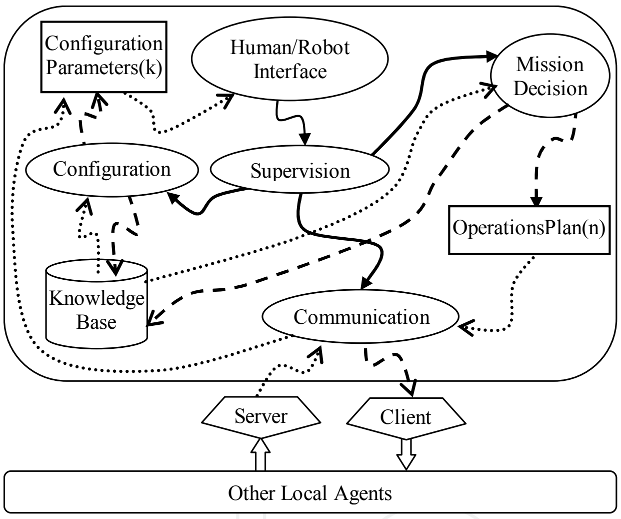


Fig. 6. Multithreading architecture of *SA*

4.2 Local Mobile/Manipulator Robot Agent

The *Local Mobile/Manipulator Robot Agent* (Fig. 7) consists of six threads:

- *Communication High-level*: communication with the other local agents.
- *Communication Low-Level*: it allows communicating with *RMRA*/*RARA* agent.
- *Configuration*.
- *Supervision*.
- *Sensors Processing*: it receives information on the environment of the remote robot. LMS, US and Odometer sensors for the mobile base; Positions sensors, Effort sensors and the State of the gripper for the manipulator.
- *Position Calculation*: for the *Mobile Robot* agent, this thread calculates the position of the mobile base on a plan relatively to any frame (R_B or R_A).

For the *Manipulator Robot* agent, this thread tests either a given Cartesian position belongs of the current workspace of the manipulator or not. In addition, this thread calculates the *Direct Kinematic Model* (*DKM*) and the *Inverse Kinematic Model (IKM)* of the manipulator.

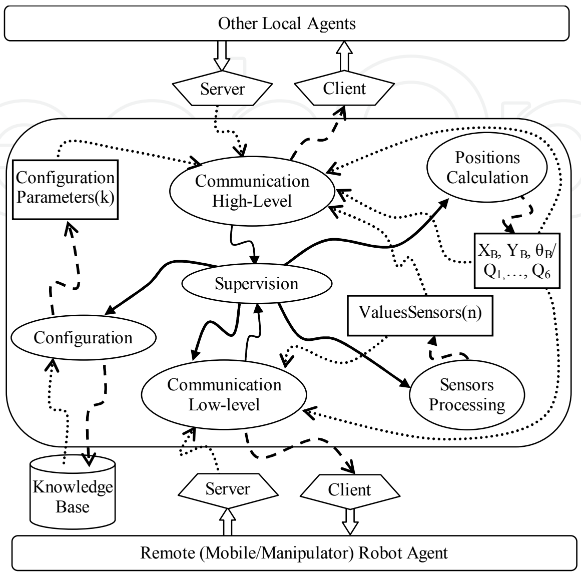


Fig. 7. Multithreading architecture of *LMRA*/*LARA*

4.3 Vision System Agent

The images captured by the *CCD* camera of the robot must undergo several operations to extract useful information and to calculate the coordinates of the objects of the scene. The *Vision System Agent* (Fig. 8) is composed of the following threads:

- *Communication*.
- *Supervision*.
- *Image Capturing*: The *CCD* camera of the robot delivers continuous video images of the scene. Images are stocked in *Image*.
- *Configuration*: the configuration of this agent consists of the *Camera calibration* (see 4.3.1) and *Camera/Gripper calibration* (see 4.3.2). These parameters are stored in the *knowledge base* of the *VSA* agent.
- *Image Processing*: Firstly, the *Median filter* is applied to remove the noise. It consists of replacing the value of a pixel by the median value of its neighbor pixels. Secondly, the resulting image is *binarised*. The *binarisation* consists of transforming the image into another format with two colors only: *black* for the *objects* and *white* for the *background*. Thirdly, *objects contours* are detected. The *contours* consist of finding pixels in the image that correspond to changes of the luminance intensity. The *algorithm of Canny* (Canny, 1986) has been used. Finally, the *forms recognition* (*characterization*) consists of identifying the forms and classifying them in the corresponding category (triangles, rectangles, circles, etc.). To this aim, the *Hough transformation* (Duda & Hart, 1972) has been used.The result is saved in *ProcessedImage*.
- *2D Extraction*: this thread calculates the *2D* coordinates of the gravity center (u_i, v_i) of all the objects. The results are stocked in *UV* array of *n* elements.

- *3D Extraction*: The aim of this thread is to compute the *3D* real coordinates (x, y, z) of the objects of the scene using the calibration parameters of the *CCD* camera.With a single camera, it is possible to estimate only two coordinates (y, z). Thus, to get the other one (x), another measurement system is needed. The used approach is that developed in (Bouzouia & Rahiche, 2009). It is as follows:
  - From the captured and processed image, the (y, z) coordinates of the selected object are calculated by using the camera model obtained by the calibration process.
  - The measure representing the other component (x) is obtained from the *LMS* sensor.

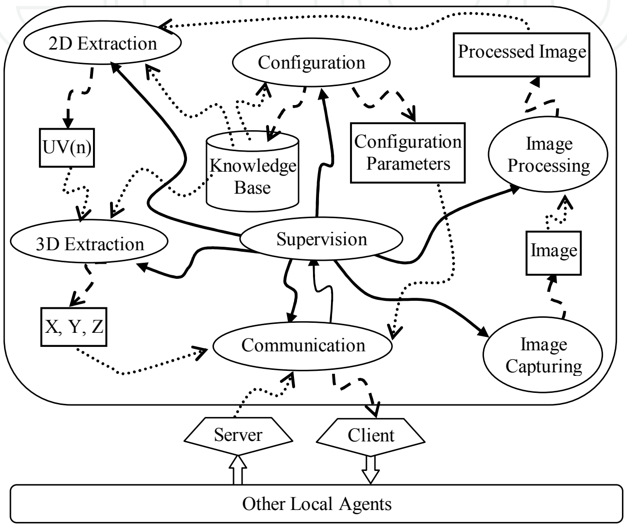
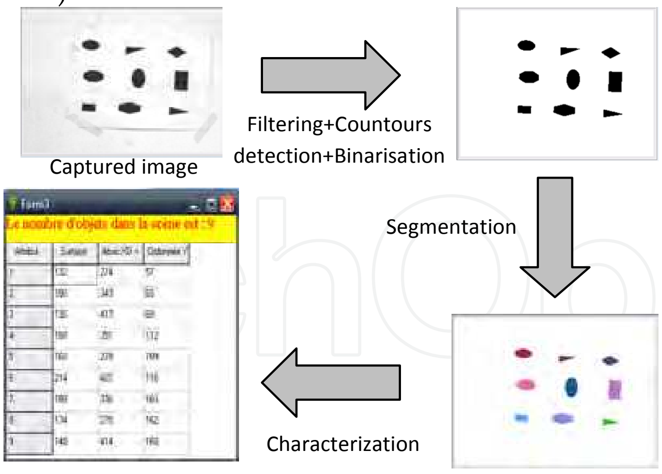


Fig. 8. Multithreading architecture of *VSA*

The *acquisition*, *Filtering*, *Segmentation* and *Characterization* steps are illustrated in Fig. 9 (Bouzouia &Rahiche, 2009).



Le nombre d'objets dans la scène est :9

[illegible]	Surface	[illegible]	Ordonnée Y
1	132	274	57
2	191	343	60
3	135	417	69
4	190	281	112
5	160	229	109
6	214	402	116
7	193	336	161
8	134	279	162
9	140	414	166

Fig. 9. Result of the image processing process

4.3.1 Camera calibration

Camera calibration consists in determining the 3×4 transformation matrix ${}^{I}T_{A}$ that maps a 3D coordinates of a point in the space $P\begin{bmatrix}x\\y\\z\end{bmatrix}$ expressed inR_A using a calibration grid, onto its 2D image projection whose coordinates $m\begin{bmatrix}u\\v\end{bmatrix}$ are expressed in pixel in R_I (Telle et al., 2003).
The relation between P and m is given by (6) where s is an arbitrary scale factor (Muis & Ohnishi, 2005):

$$\begin{bmatrix}su\\sv\\s\end{bmatrix} = {}^{I}T_{A} * \begin{bmatrix}x\\y\\z\\1\end{bmatrix} \tag{6}$$

For the camera calibration, the method proposed in (Bénallal, 2002) is adopted. It consists of solving (7) with $n \geq 6$ (Hartley & Zisserman, 2001) and m_{11}, m_{12} ... m_{33} are the elements of the matrix ${}^{I}T_{A}$.

$$\begin{bmatrix}x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & -u_1.x_1 & -u_1.y_1 & -u_1.z_1\\ 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & -v_1.x_1 & -v_1.y_1 & -v_1.z_1\\ x_2 & y_2 & z_2 & 1 & 0 & 0 & 0 & 0 & -u_2.x_2 & -u_2.y_2 & -u_2.z_2\\ 0 & 0 & 0 & 0 & x_2 & y_2 & z_2 & 1 & -v_2.x_2 & -v_2.y_2 & -v_2.z_2\\ . & . & . & . & . & . & . & . & . & . & .\\ . & . & . & . & . & . & . & . & . & . & .\\ x_n & y_n & z_n & 1 & 0 & 0 & 0 & 0 & -u_n.x_n & -u_n.y_n & -u_n.z_n\\ 0 & 0 & 0 & 0 & x_n & y_n & z_n & 1 & -v_n.x_n & -v_n.y_n & -v_n.z_n\end{bmatrix} * \begin{bmatrix}m_{11}\\m_{12}\\m_{13}\\m_{14}\\m_{21}\\m_{22}\\m_{23}\\m_{24}\\m_{31}\\m_{32}\\m_{33}\end{bmatrix} = \begin{bmatrix}u_1\\v_1\\u_2\\v_2\\.\\.\\u_n\\u_n\end{bmatrix} \tag{7}$$

The obtained matrix M is given by (8) (Hentout et al., 2009d):

$$M \approx \begin{pmatrix}-0.2057 & 0.8110 & -0.079 & 69.4449\\ 0.6498 & 0.1155 & -0.434 & 139.8394\\ 0.0005 & 0.0002 & 0.0006 & 1\end{pmatrix} \tag{8}$$

4.3.2 Camera/Gripper Calibration

Camera/Gripper calibration consists of finding the matrix${}^{E}T_{C}$defining R_C in R_E.
Let ${}^{C}T_{A1}$and ${}^{C}T_{A2}$ be the transformation matrices defining a *first* and a *secondposition* of the camera in R_A. Let ${}^{M}T_{E1}$ and ${}^{M}T_{E2}$ be the transformation matrices defining the two positionsof the end-effector in R_M corresponding to the first and the second position of the camera.
To find the Camera/Gripper calibration matrix ${}^{E}T_{C}$, the method developed in (Tsai & Lenz, 1989) is chosen. It is based on the *Least squares method* and consists of solving (9) where:

- $A=({}^{C}T_{A2})*({}^{C}T_{A1})^{-1}$: The measurable transformation matrix of the camera from its first to its second location (relative camera motion).
- $B=({}^{M}T_{E2})^{-1}*({}^{C}T_{E1})$: The measurable transformation matrix of the gripper from its first to its second location (relative robot gripper motion).

$$A * {}^{E}T_{C} = {}^{E}T_{C} * B \tag{9}$$

The obtained matrix ${}^{E}T_{C}$ is given by (10) (Hentout et al., 2009e):

$$
{}^{E}T_{C} \approx \begin{pmatrix} 0.7322 & -0.6573 & 0.1787 & -183.1608 \\ 0.6595 & 0.7497 & 0.0555 & -69.4161 \\ -0.1704 & 0.0772 & 0.9823 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{10}
$$

4.4 Remote Mobile Robot Agent

The *Remote Mobile Robot Agent*, given by Fig. 10, consists of seven threads:

- *Configuration.*
- *Communication.*
- *Supervision.*
- *Reading LMS Sensors*: it scans continuously the serial port of the *LMS sensors* and stores data in *ValuesLMS* of 181 items.
- *Reading US Sensors*: This thread, in its turn, reads constantly the *US sensors* and stores *US* data in *ValuesUS* of 24 items.
- *Odometry*: This thread reads the values of the incremental encoders (*E_R*, *E_L*), installed on the driven wheels of the mobile base, and calculates its current position and orientation angle (*New_X*, *New_Y*, *New_θ*) as shown in (Hentout et al., 2009a).
- *Navigation*: it consists of the main role of this agent. It uses data of all the other threads. *Navigation* calculates velocities (*Spd_R*, *Spd_L*) to be sent to the actuators of the mobile base in order to move to a *Target position* given by (X_{Target}, Y_{Target}, θ_{Target}) while avoiding possible obstacles.

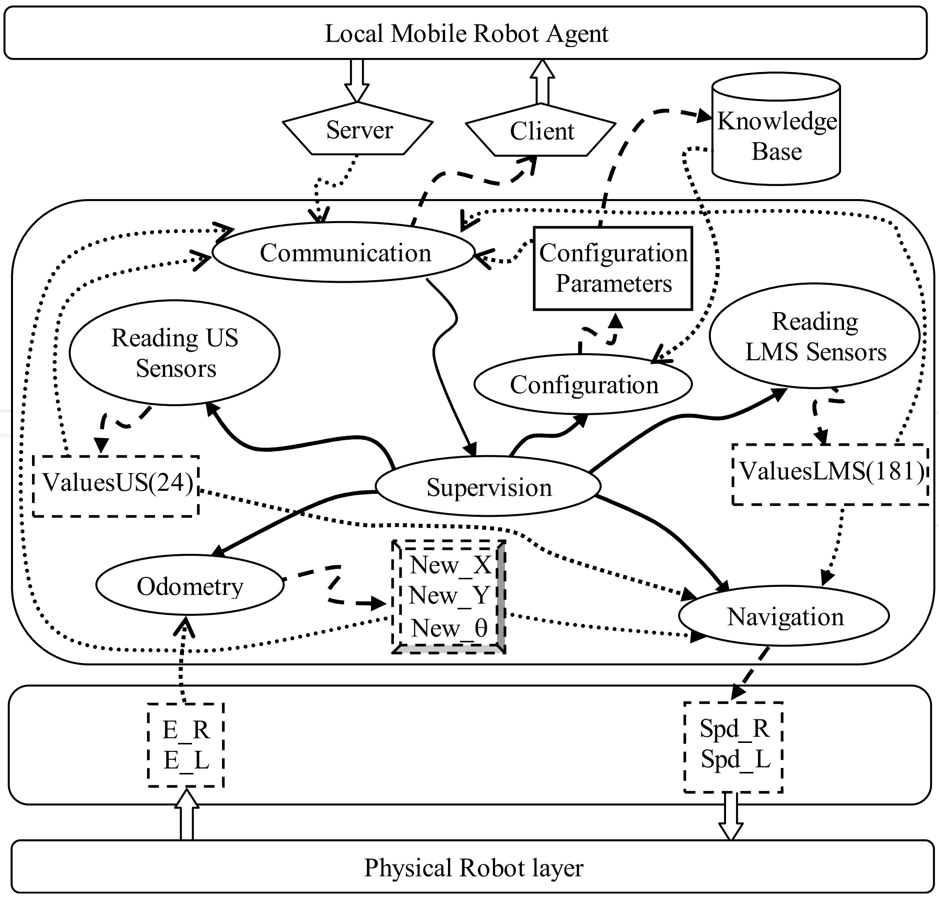


Fig. 10. Multithreading architecture of *RMRA*

4.5 Remote Manipulator Robot Agent

The *Remote Manipulator Robot Agent* (Fig. 11) is composed, in its turn, of seven threads:

- *Communication.*
- *Configuration.*
- *Supervision.*
- *Reading Positions Sensors*: it reads the incremental positions sensors (*PcIn*) installed on each articulation of the manipulator and saves data in *ValuesPos* of 6 elements.
- *Reading Effort Sensors/State Gripper*: this thread, in its turn, reads the effort sensor data (*Gripper*) and the state of the gripper (Opened or Closed). This thread stores read data in *ValuesGripper* of 7 items (*Forces (Fx, Fy, Fz)*, *Torques (Tx, Ty, Tz)*, *State of the Gripper*).
- *Open/Close gripper*: it opens or closes the gripper.
- *Movement*: This thread consists of the main role of this agent. It calculates orders (*PcOut*) to be sent to the actuators of the manipulator in order to move to a *Target position* given by ($Q_1 \ldots Q_6$) with a given velocity V_i (i=1…6) for each joint.

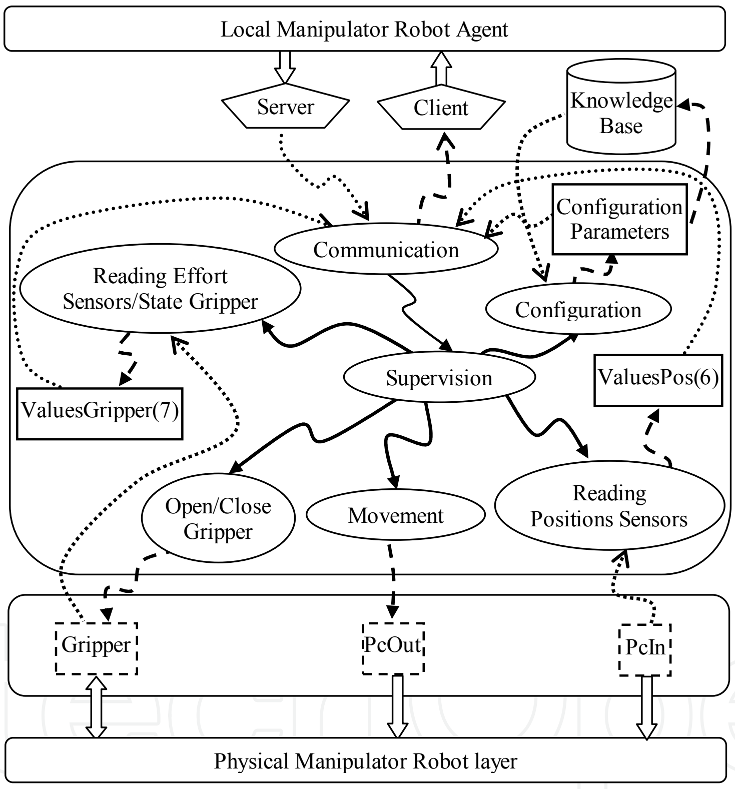


Fig. 11. Multithreading architecture of *RARA*

5. Experimental part

The core thinking of modeling and controlling mobile manipulators using a multi-agent system is that of realizing cooperation between the manipulator, the mobile base and the sensors system. In order to show the validity of the implementation of the *SA*, *LMRA*, *LARA*, *VSA*, *RMRA* and *RARA* agents, two different missions are considered in this section.

For the envisaged experiments, all the positions and orientations are given in R_A. In addition, two cases are distinguished (Fig. 12):

- *The Target belongs of the current workspace of the robot*: this task requires only the motion of the manipulator.
- *The Target is outside the current workspace of the robot*: In this case, the mobile base moves until the *Target* is within the new workspace of the manipulator. Then, the robot manipulates the *Target* with its end-effector.

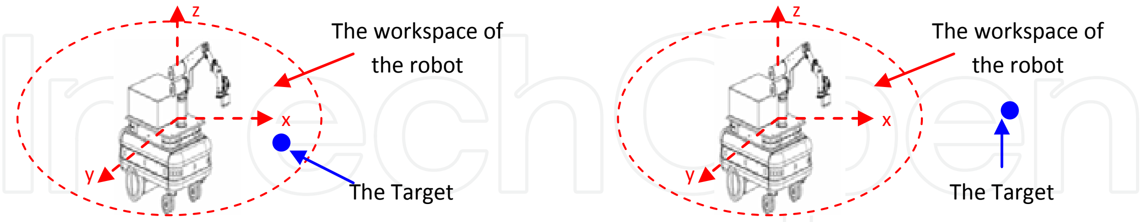


Fig. 12. The two possible cases for a given *Target*

5.1. Following a predefined operational trajectory

This experiment presents trajectory planning and control for mobile manipulators. The end-effector of the robot has to be, as near as possible, from a predefined operational trajectory (given by a set of Cartesian coordinates (*X*, *Y*, *Z*)) while the non-holonomic mobile base has to avoid the obstacles present in the environment.

5.1.1 Straight-line following

In this experiment, as shown in Fig. 13, the operational trajectory to be followed by the end-effector of the robot consists of a straight-line connecting an initial position $P_i(X_i, Y_i, Z_i)$ to a final position $P_f(X_f, Y_f, Z_f)$ (Hentout et al., 2009e).

To compute the imposed positions (*Targets*) to be reached by the end-effector, let's consider:

- $[P_i, P_f]$: the segment connecting P_i to P_f.
- $p(X_p, Y_p, Z_p)$: the current position coordinates of the end-effector.
- $h(X_h, Y_h, Z_h)$: the projection of p on the segment $[P_i, P_f]$.
- *New_X, New_Y, New_θ*: the current position coordinates and orientation angle of the mobile base.
- $Position_{Init}(X_{BInit}, Y_{BInit}, \theta_{BInit})$: the initial position coordinates and orientation angle of the mobile base.
- $Position_{Fin}(X_{BFin}, Y_{BFin}, \theta_{BFin})$: the final position coordinates and orientation angle of the mobile base.
- $m(X_m, Y_m, Z_m)$: a point in the space.

$$m \in [P_i, P_f] \Leftrightarrow \begin{cases} X_m = t * (X_f - X_i) + X_i \\ X_m = t * (Y_f - Y_i) + Y_i \\ Z_m = t * (Z_f - Z_i) + Z_i \\ \qquad t\epsilon[0, 1] \end{cases} \tag{11}$$

$\overrightarrow{hp}$ is orthogonal to $\overrightarrow{P_i P_f}$ so:

$$(X_h - X_p) * (X_f - X_i) + (Y_h - Y_p) * (Y_f - Y_i) + (Z_h - Z_p) * (Z_f - Z_i) = 0 \tag{12}$$

From (11) and (12), the position of h (the next *Target* to be reached by the end-effector of the robot) on the segment $[P_i, P_f]$ is given by (13):

$$\begin{aligned} X_h &= t*(X_f - X_i) + X_i \\ Y_h &= t*(Y_f - Y_i) + Y_i \\ Z_h &= t*(Z_f - Z_i) + Z_i \\ t &= \frac{(X_f - X_i)*(X_p - X_i) + (Y_f - Y_i)*(Y_p - Y_i) + (Z_f - Z_i)*(Z_p - Z_i)}{(X_f - X_i)^2 + (Y_f - Y_i)^2 + (Z_f - Z_f)^2} \end{aligned} \tag{13}$$

The positioning error of the end-effector is calculated by (14):

$$Err = \sqrt{(X_h - X_p)^2 + (Y_h - Y_p)^2 + (Z_h - Z_p)^2} \tag{14}$$

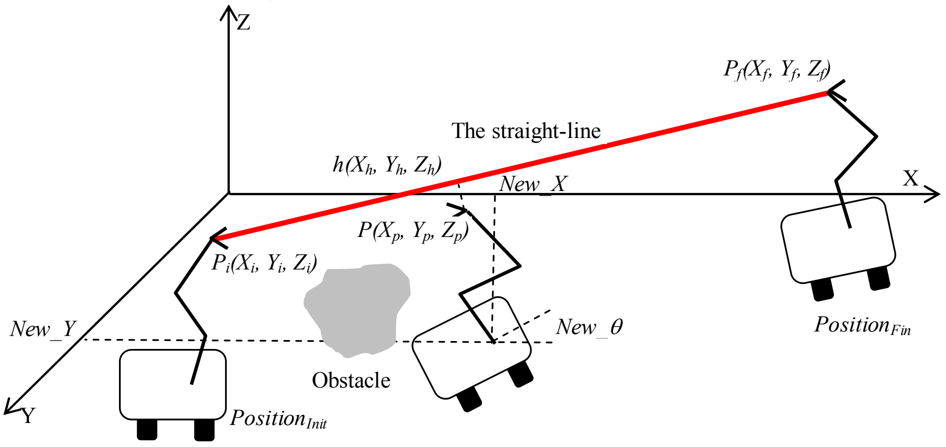


Fig. 13. The straight-line following mission and its parameters

The straight-line following algorithm for *LARA* is given as follows:

```
Straight_Line_Following_LARA($P_i$, $P_f$){
    if($P_i \notin$ Current workspace of the manipulator)
        wait for message (Mobile base in $Position_{Init}$) from LMRA;
    Generate the possible orientations for $P_i$ using the IKM;
    $Q_i$(i=1…6) = Choose the best configuration;
    Send Move($Q_i$(i=1…6)) to RARA;
    while ($P$ != $P_f$){
        Receive (New_X, New_Y, New_θ) from LMRA;
        Calculate $P$ in $R_A$ according to (New_X, New_Y, New_θ);
        if ($P \notin [P_i, P_f]$){
            Calculate $h$, the projection of $P$ on $[P_i, P_f]$;
            Generate the possible orientations for $h$ using the IKM;
            $Q_i$(i=1…6) = Choose the best configuration;
        }
        Send Move(Qi(i=1…6)) to RARA;
    }
}
```

The straight-line following algorithm for *LMRA* is given here below:

```
Straight_Line_Following_LMRA($P_i$, $P_f$){
        Calculate $Position_{Init}$ corresponding to $P_i$;
        Send $Move(Position_{Init})$ to $RMRA$;
        wait for (Mobile base in $Position_{Init}$) from $RMRA$;
        Send (Mobile base in $Position_{Init}$) to $LARA$;
        Calculate $Position_{Fin}$ corresponding to $P_f$;
        Send $Move(Position_{Fin})$ to $RMRA$;
        while ((New_X, New_Y, New_θ)!=$Position_{Fin}$){
                Receive (New_X, New_Y, New_θ) from $RMRA$;
                Send (New_X, New_Y, New_θ) to $LARA$;
        }
}
```

These two previous algorithms are executed in parallel on the off-board *PC* by the corresponding agents. In addition, *LMRA* and *LARA* send requests and receive sensors data and reports from the corresponding agent (*RMRA* and *RARA*). At the same time, *RMRA* and *RARA* move towards the received positions: $Position_{Fin}$ for the mobile base and (Q_1 … Q_6) for the manipulator.

5.1.2 Experimental result

The straight-line following algorithms proposed previously for *LMRA* and *LARA* are implemented to the *RobuTER/ULM*. (13) is used to generate the *Target* positions so that the end-effector of the mobile manipulator follows the desired line (Hentout et al., 2009e).

For this experiment $P_i(X_i, Y_i, Z_i)$ =(-691.72mm, -108.49mm, 1128.62mm) and $P_f(X_f, Y_f, Z_f)$ = (-2408.17mm, -108.49mm, 1472.30mm). Therefore, the operational trajectory consists of a straight-line with a slope of about 350mm (343.68mm).

The initial posture of the mobile base and that of the end-effector corresponding to P_i is $Target_{Init}(X_{BInit}, Y_{BInit}, \theta_{BInit}, X_{EInit}, Y_{EInit}, Z_{EInit}, \psi_{EInit}, \theta_{EInit}, \varphi_{EInit})$ = (0mm, 0mm, 0°, -691.72mm, -108.49mm, 1128.62mm, -90°, -90°, -90°). For this initial position, the initial joint angles ($Q_{1Init}, Q_{2Init}, Q_{3Init}, Q_{4Init}, Q_{5Init}, Q_{6Init}$) = (0°, 60°, 0°, 0°, 32°, 0°).The final position of the mobile base and that of the end-effector corresponding to $P_f$is $Target_{Fin}(X_{BFin}, Y_{BFin}, \theta_{BFin}, X_{EFin}, Y_{EFin}, Z_{EFin}, \psi_{EFin}, \theta_{EFin}, \varphi_{EFin})$ = (-1920mm, 2mm, 15°, -2408.17mm, -108.49mm, 1472.30mm, 0°, -90°, 0°). For this final position, the final joint angles ($Q_{1Fin}, Q_{2Fin}, Q_{3Fin}, Q_{4Fin}, Q_{5Fin}, Q_{6Fin}$) = (37°, 52°, 61°, 73°, -57°, 28°).

Two cases are tested for this example(Hentout et al., 2009e):

- The environment of the robot is free (no obstacles are considered). The motion of the mobile base consists also of a straight-line connecting $Position_{Init}$ to $Position_{Fin}$. For this case, the robot follows perfectly the imposed straight-line.
- The second case is more difficult. The non-holonomic mobile base has to avoid an obstacle present in the environment while the end-effector has to be always at the desired configuration (on the straight-line).For the second case of this experiment, the operational trajectory followed by the end-effector and the imposed trajectory for the end-effector are shown on Fig. 14.

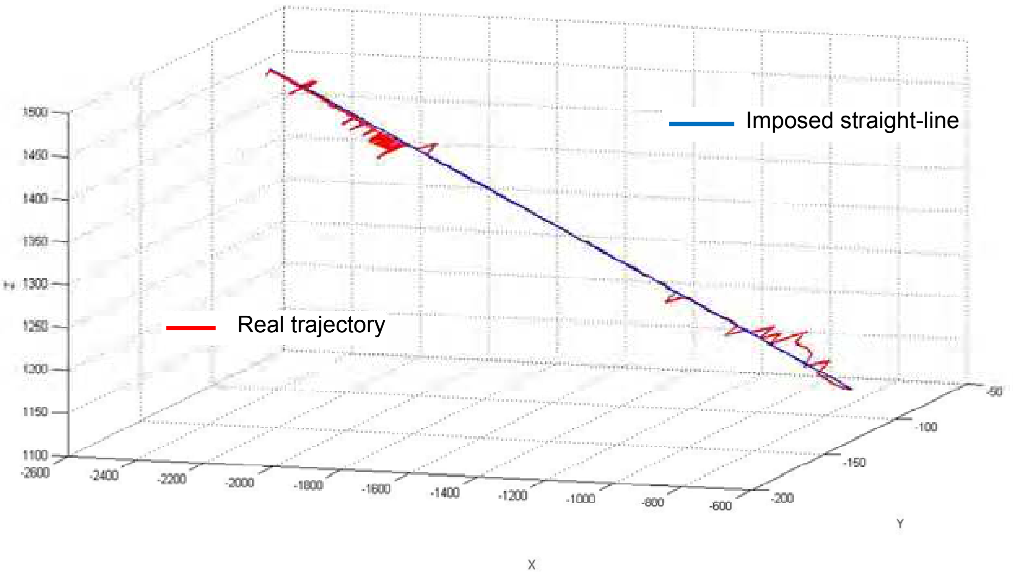


Fig. 14. Imposed operational trajectory and the real trajectory of the end-effector

The real joints variations rather than the desired trajectory for some joints (1, 2, 3 and 5 respectively) are shown on Fig. 15.

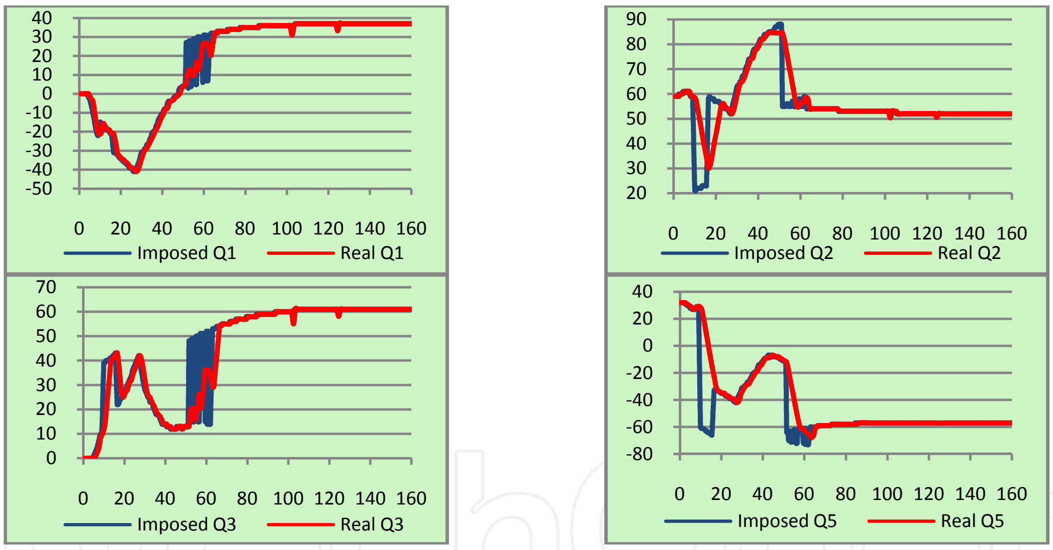


Fig. 15. Joints variations and desired trajectories of some joints

Fig.16 shows the trajectory followed by the mobile base and the avoidance of the obstacle.

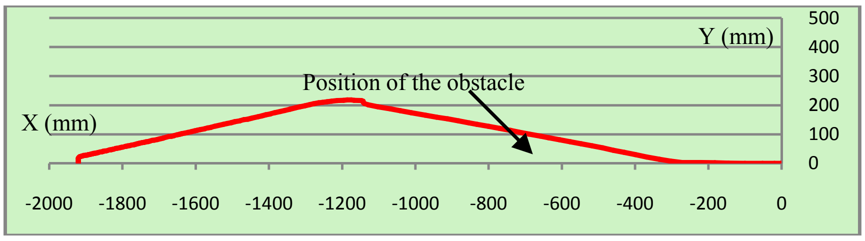


Fig. 16. Real trajectory followed by the mobile base

5.1.3 Discussion of results

Figs. 14, 15 and 16 showed the operational trajectory of the end-effector, the variations of some joints of the manipulator and the motion of the mobile base respectively. The mission took about 160 seconds.

The maximum positioning error calculated by (14) is 24.43mm while the average error is 3.41mm. The errors show that it is difficult to follow the desired straight-line.

The first reason for this error is the initial positioning error of the mobile base ($Position_{Init}$). It causes straying from the initial position for the end-effector in the trajectory. To solve this problem, the mobile manipulator must absorb this error by the motion of its manipulator. Secondly, an estimated positioning error of the mobile base, calculated by odometry (*New_X, New_Y, New_*θ), during its motion effects the tip position of the end-effector directly. To absorb this error, the manipulator should move quickly to adjust itself when the error is detected. Finally, the low velocity of the manipulator's motion during the motion of the mobile base causes a delay in the positioning of the end-effector. This problem can be solved by incrementing the velocity of the manipulator according to that of the mobile base.

5.2 Aligning the end-effector of the robot to different objects by using the eye-in-hand camera and the LMS sensor

A position-based servoing control of mobile manipulators by using the eye-inhand camera and the *LMS* sensor is considered. The working mission is to reach different positions (corresponding to various objects) by the end-effector of the robot.

To reach an object, it is necessary to capture an image of this object. *VSA* sends, for this aim, a request *Move Gripper* (*Position*) to *LARA* in order to position the manipulator. *Position* is read from the *Knowledge Base* of *VSA*. After the positioning of the manipulator, *VSA* captures an image and carries out the necessary processing to extract the *2D* coordinates (u, v) of the gravity center of the object in the image. At the end, *VSA* extract (y, z) coordinates and sends them to *LARA*. *LARA* always needsthe (x) coordinate. To this aim, a request is sent, in parallel, to *LMRA* (*Read LMS*) which transmits it to *RMRA*. Receiving this request, *RMRA* send *LMSValues* data to *LMRA*. This latter selects the minimum value from the 60th element to 120th element (corresponding to 60° to 120°). This value corresponds to the (x) coordinate. It is sent to *LMRA* which has now (x, y, z) coordinates of the position to be reached.

5.2.1 Experimental result

For this experiment, as shown in Fig. 18, the initial posture of the mobile base and that of the end-effector is $Target_{Init}(X_{BInit}, Y_{BInit}, \theta_{BInit}, X_{EInit}, Y_{EInit}, Z_{EInit}, \psi_{EInit}, \theta_{EInit}, \varphi_{EInit})$ = (0mm, 0mm, 0°, -546.62mm, -110.36mm, 1200.73mm, -90°, -90°, -90°). The position to be reached by the end-effector of the robot are at a distance x=-2470. For this initial position, the initial joint angles (Q_{1Init}, Q_{2Init}, Q_{3Init}, Q_{4Init}, Q_{5Init}, Q_{6Init}) = (0°, 87°, 0°, 0°, 5°, 0°). Table 2 shows the different parameters of this experiment.

Image Coordinates (u, v) (pixel)	Real Coordinates (x, y, z) (mm)	LMS value (x) (mm)	Calculated Coordinates (x, y, z) (mm)	Mobile Base coordinates (X_B, Y_B, θ_B)
166, 176	-2470, -63, 1325	2470	-2470, -63.01, 1328.56	-1670mm, 0mm, 0°
182, 228	-2470, 52, 1295	2470	-2470, 46.40, 1294.85	-1670mm, 0mm, 0°
228, 178	-2470, -58, 1200	2470	-2470, -58.93, 1197.94	-1670mm, 0mm, 0°
234, 227	-2470, 47, 1185	2470	-2470, 44.25, 1185.30	-1670mm, 0mm, 0°

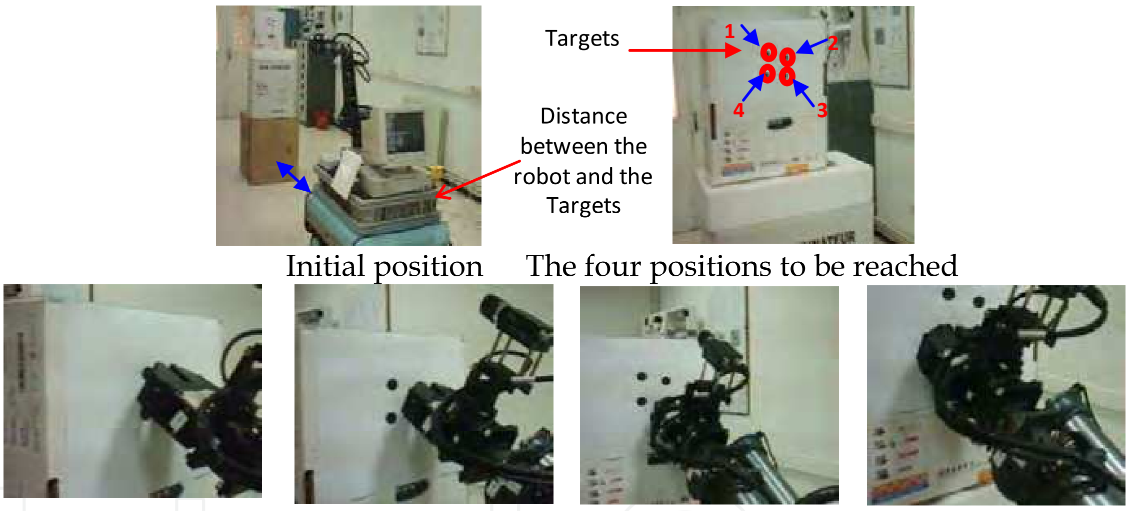
Table 2. Different parameters of the experiment

The joints angles and the corresponding end-effector coordinates are given in Table 3.

Target	Joints angles Q_i (i=1..6) (°)	End-effector coordinates $(X_E, Y_E, Z_E, \psi_E, \theta_E, \varphi_E)$
Target₁	(5, 49, 63, -13, -22, -78)	-2466.52mm, -60.45mm, 1335.20mm, -90°, -90°, 180°
Target₂	(16, 51, 54, -48, -22, -44)	-2469.86mm, 44.17mm, 1293.73mm, -90°, -90°, 180°
Target₃	(5, 53, 37, 89, 5, 1)	-2468.08mm, -60.95mm, 1199.39mm, -90°, -90°, 0°
Target₄	(16, 53, 35, 83, 16, 7)	-2469.41mm, 45.15mm, 1185.48mm, -90°, -90°, 0°

Table 3. Joints angles and end-effector postures for the different *Targets*

The following snapshots (Fig. 18) show the obtained result (Bouzouia & Rahiche, 2009):



Initial position The four positions to be reached

The 1st point The 2nd point The 3rd point The 4th point

Fig. 18. Position-based servoing control by using the camera and the LMS sensor of *RobuTER/ULM*

5.2.2 Discussion of results

The *VSA* agent uses the eye-in-hand camera to extract the two last coordinates (y and z) of the object to be manipulated by the robot. The *LMRA* and the *RMRA* agent use the *LMS* sensor to obtain the first coordinate (x).

The maximum 3D reconstruction error calculated by (14) is 5.60mm while the minimum error is 2.26mm. These errors are acceptable. They are due to the weak precision of the measured real values, to the low rigidity of the manipulator, to the accumulation errors of the calibration processand to the feeble precision of the *LMS* sensor (±15mm).

The maximum positioning error is 11.07mm while the minimum error is 2.00mm. The errors are principally due to the error in the estimated positioning of the mobile base, calculated by odometry (*New_X, New_Y, New_θ*), during its motion. These errors are also due the accumulated errors in the *IKM* of the manipulator.

6. Conclusions and future works

This chapter presented a multi-agent control architecture of mobile manipulators. The architecture consists of six agents: *Supervisory Agent* (*SA*), *Local Mobile Robot Agent* (*LMRA*), *Local Manipulator Robot Agent* (*LARA*), *Vision System Agent* (*VSA*), *Remote Mobile Robot Agent* (*RMRA*) and *Remote Manipulator Robot Agent* (*RARA*). The first four agents are installed on an off-board *PC* while the two other agents are installed on the on-board *PC* of the robot.

The controller was applied successfully to follow a predefined straight-line operational trajectory by the end-effector of a differentially-driven *RobuTER/ULM* mobile manipulator while considering obstacles in its environment. The controller was shown to be relatively effective when the robot moves with small velocities.

To realize the operational trajectory following, one of the biggest problems is that an accumulated error of the estimated position of the mobile base affects the position accuracy of the end-effector. Therefore, the manipulator should have a capability to adjust its position when the mobile base detects positioning errors.

The results obtained the position-based servoing control of the robot by using the eye-in-hand camera and the *LMS* sensor are satisfactory since the positioning error of the end-effector is less than 15mm. The calculation of the 3D coordinates is based on the eye-in-hand camera (for (*y*, *z*) coordinates) and on the *LMS* sensor (for (*x*) coordinate).

In future works, the performances and the robustness of the implemented agents of the control architecture should be shown and discussed through examples of other types of trajectories (circular, etc.). Furthermore, and especially for the *VSA* agent, a moving target tracking problem should be performed. In addition, the real time constraint for the *VSA* agent will be verified and discussed. Another extension of this work is to introduce a virtual reality system (a graphic simulator) to give more effective action for the developed architecture.

7. References

Alfaro, C., Ribeiro, M. I.& Lima, P. (2004). “Smooth Local Path Planning for a Mobile Manipulator”. *Robotica 2004, The 4th Portuguese Robotics Festival*, Protugal, pp. 127-134.

Bénallal, M. (2002). “Système de Calibration de Caméra: Localisation de Forme Polyédrique par Vision Monoculaire”, *Ph. D. Thesis*, Ecole des mines de Paris, France.

Blaer, P.& Allen, P.K. (2002). “Topological Mobile Robot Localization Using Fast Vision Techniques”. *IEEE International Conference on Robotics and Automation*, pp. 1031-1036, Washington, USA.

Bouzouia, B. & Rahiche, A. (2009). "Teleoperation System of the Mobile Manipulator Robot RobuTER/ULM: Implementation Issues". *The XXII International Symposium on Information, Communication and Automation Technologies (ICAT 2009)*, Sarajevo, Bosnia & Herzegovina, October 29-31.

Canny, J. (1986). “A Computational Approach to Edge Detection”, *IEEE Transactions on Pattern analysis and machine intelligence, 8(6)*, pp. 679-698.

Chen, Y., Liu, L., Zhang, M.& Rong, H. (2006). "Study on Coordinated Control and Hardware System of a Mobile Manipulator". *Proceedings of the 6th World Congress on Intelligent Control and Automation*, Dalian, China. pp. 9037-9041, June 21-23.

Colle, E., Nait Chabane, K., Delarue, S. & Hoppenot, P. (2006). "Comparaison d'une méthode classique et d'une méthode utilisant la coopération homme-machine pour exploiter la redondance de l'assistant robotisé". *HANDICAP 2006*, France, June 7-9.

Dombre, E.& Khalil, W. (2007). "Modeling, Performance Analysis and Control of Robot Manipulators". *Control Systems, Robotics and Manufacturing Series*. Hermes Science Publishing.

Duda, R.O.& Hart, P.E. (1972). "Use of the Hough transformation to detect lines and curves in pictures", *Communications of ACM, 15(1)*, pp. 11-15.

Erden, M. S., Leblebicioglu, K. & Halici, U. (2004). "Multi-Agent System-Based Fuzzy Controller Design with Genetic Tuning for a Mobile Manipulator Robot in the Hand Over Task". *Journal of Intelligent and Robotic Systems, 39(3)*, pp. 287-306.

Flandin, G., Chaumette, F. & Marchand, E. (2000). "Eye-in-hand/eye-to-hand cooperation for visual servoing," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2741–2746.

Hartley, R. & Zisserman, A. (2001). "Multiple View Geometry in Computer Vision". *Cambridge*.

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

Hentout, A., Bouzouia, B., Akli, A., Toumi, R.,& Toukal, Z. (2009a). "Agent-Based Coordinated Control of Mobile Manipulators: Study on Hardware System of a Mobile Manipulator". *International Conference on Systems and Processing Information (ICSIP'09)*, Guelma, Algeria, May 2-4.

Hentout, A., Bouzouia, B., Toumi, R. & Toukal, Z. (2009b). "Agent-Based Control of Remote Mobile Manipulator Robotic System: Study of Cooperation". *Proceeding of theInternational Congress ofIndustrial Engineering 2009 (CIGI'09)*, Toulouse, France, June 10-12.

Hentout, A., Akli, I., Bouzouia, B., Ouzzane, M. E., Benbouali, R. & Bouskia, M. A. (2009c). "Implementation of a Multi-agent Architecture for Remote Control of Mobile Manipulators". *The IEEE International Conference on Applied Informatics (ICAI'09),* Bordj Bou Arréridj, Algeria, November 15-17.

Hentout, A., Bouzouia, B., Akli, I., Bouskia, M. A., Benbouali, R. & Ouzzane, M. E. (2009d). "Multi-Agent Control Architecture of Mobile Manipulators: Extraction of 3D Coordinates of Object Using an Eye-in-Hand Camera", *The 3rd 2009 IEEE International Conference on Signals, Circuits and Systems (SCS'09)*, Jerba, Tunisia, November 6-8.

Hentout, A., Bouzouia, B., Akli, I., Ouzzane, E., Benbouali, R. & Bouskia, M. A. (2009e). "Multi-Agent Control Architecture of Mobile Manipulators: Following an Operational Trajectory". *The 3rd 2009 IEEE International Conference on Signals, Circuits and Systems (SCS'09)*, Jerba, Tunisia, November 6-8.

Hutchinson, S., Hager, G. D.& Corke, P.I. (1996). "A tutorial on visual servo control". *IEEE Transactions on Robotics and Automation, 12(5)*, pp. 651-670.

Janabi-Sharifi, F.& Wilson, W. J. (1998). "Automatic Grasp Planning for Visual-Servo Controlled Robotic Manipulators". *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 28(5)*.

Katz, D., Horrell, E., Yang, Burns, Y., Buckley, B., Th., Grishkan, A., Zhylkovskyy, V., Brock, O.& Learned-Miller,E. (2006). "The UMass Mobile Manipulator UMan: An Experimental Platform for Autonomous Mobile Manipulation". *IEEE Workshop on Manipulation for Human Environments*, Philadelphia, USA.

Khalil, W. & Kleinfinger, J. F. (1986). "A new geometric notation for open and closed-loop robots". *The International Conference on Robotics and Automation*, USA, pp. 1175-1180.

Muis, A.& Ohnishi, K. (2005). "Eye-to-Hand Approach on Eye-in-Hand Configuration within Real-Time Visual Servoing". *IEEE/ASME Transactions on Mechatronics, 10(4*).

Nagatani, K., Hirayama, T., Gofuku, A. & Tanaka, Y. (2002). "Motion planning for mobile manipulator with keeping manipulability". *Proceeding of the 2002 IEEE/RSJ International conference on Intelligent Robots and Systems (IROS 2002)*, Switzerland.

Petersson, L., Egerstedt, M. & Christensen, H.I. (1999). "A hybridcontrol architecture for mobile manipulation". *Proceedings of IROS'99,* Kyongju, Korea, pp. 1285-1291.

Sasaki, H., Takahashi, T. & Nakano E. (2001). "A door opening method by a mobile manipulator with passive joints". *Journal of Robotic Society of Japan, 19(2)*, pp. 129-136.

Sugar, T. & Kumar, V., (1998). "Decentralized Control of Cooperating Mobile Manipulators," *Proceedings of the IEEE International Conference on Robotics and Automation*, Leuven, Belgium, pp. 2916-2921.

Telle, B., Aldon, M.-J. & Ramdani, N. (2003). "Guaranteed 3D Visual Sensing Based on Interval Analysis". *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*. pp. 1566-1571, October 27-31.

Trabelsi, M., Aitoufroukh, N.& Lelandais, S. (2005). "Saisie d'objets à l'aide d'une caméra et des ultrasons pour la robotique de service". *The 3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications (SETIT 2005)*, Tunisia, March 27-31.

Tsai, R.Y. & Lenz R.K. (1989). "A New Technique for Fully Autonomous and Effcient 3D Robotics Hand/Eye Calibration", *IEEE Transactions on robotics and automation, 5(3)*, pp. 345-358.

Waarsing, B. J. W., Nuttin, M.& Van Brussel,H. (2003) "Behaviour-based mobile manipulation: the opening of a door". *ASER'03, Bardolino (Italy)*, pp. 168-175.

Winter, N., Gaspar, J., Lacey, G.& Santos-Victor, J. (2000). "Omnidirectional Vision for Robot Navigation". *Proceedings of the IEEE Workshop on Omnidirectional Vision (OMNIVIS)*, pp. 21-28, South Carolina, USA.

Robot Manipulators New Achievements

Edited by Aleksandar Lazinica and Hiroyuki Kawai

ISBN 978-953-307-090-2
Hard cover, 718 pages
Publisher InTech
Published online 01, April, 2010
Published in print edition April, 2010

Robot manipulators are developing more in the direction of industrial robots than of human workers. Recently, the applications of robot manipulators are spreading their focus, for example Da Vinci as a medical robot, ASIMO as a humanoid robot and so on. There are many research topics within the field of robot manipulators, e.g. motion planning, cooperation with a human, and fusion with external sensors like vision, haptic and force, etc. Moreover, these include both technical problems in the industry and theoretical problems in the academic fields. This book is a collection of papers presenting the latest research issues from around the world.

How to reference

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

A. Hentout, B. Bouzouia, I. Akli and R. Toumi (2010). Mobile Manipulation: A Case Study, Robot Manipulators New Achievements, Aleksandar Lazinica and Hiroyuki Kawai (Ed.), ISBN: 978-953-307-090-2, InTech, Available from: http://www.intechopen.com/books/robot-manipulators-new-achievements/mobile-manipulation-a-case-study

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

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