OPERATIONAL TRAJECTORY FOLLOWING FOR WHEELED MOBILE MANIPULATORS

A. HENTOUT, B. BOUZOUIA, R. BENBOUALI
Division of Computer-Integrated Manufacturing and Robotics (DPR)
Centre for Development of Advanced Technologies (CDTA)
BP 17, Baba Hassen, Algiers 16303, Algeria
hentout_abdelfetah@hotmail.com, ahentout@cdta.dz

R. TOUMI, E. OUZZANE
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

Abstract: *This paper presents an approach for the trajectory following problem for wheeled mobile manipulators. The end-effector of the mobile manipulator has to follow a predefined operational trajectory in cluttered environments.*

The control architecture of the robot consists of six independent agents. Four agents are installed on an off-board PC and the other agents are installed on the on-board PC of the mobile manipulator. Each agent models a principal function of the mobile manipulator and manages a different sub-system.

The validity of the approach is demonstrated using the RobuTER/ULM mobile manipulator. The end-effector of the manipulator is asked to follow a straight-line while the non-holonomic differentially-driven wheeled mobile base has to avoid the obstacles present in the environment. As the mobile base moves, the end-effector of the robot is positioned, as near as possible, at the preferred configuration (the straight-line) due to the different messages exchanged between the agents of the architecture (current position coordinates, orientation angles, etc.).

Keywords**:** *Operational trajectory following, Multi-agent control architecture, Differentially-driven wheeled mobile manipulator, RobuTER/ULM.*

1. Introduction

A mobile manipulator consists of a mobile base carrying a standard manipulator. This robotic system combines the navigation of a mobile base and the control of a static manipulator. The most important feature of such robots is the flexible operational workspace compared to the limited workspace of fixed manipulators [1]. This feature endows a mobile manipulator with the ability to operate in a large scale of operation [2].

Mobile manipulators have applications in many areas such as handling and transporting parts from one place to another, mining, forestry, construction, etc. Recently, target environment for activity of such robots has been shifting from factory environment to human environment [3] (office buildings, hospitals, homes, etc.) because they are particularly well suited for human-like tasks [4].

In recent years, a number of researchers studied path planning, trajectory following and control for mobile manipulators. Erden and colleagues [5] described a multi-agent control system of 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. Colle and *al*. [6] proposed a multi-agent system for controlling their mobile manipulator *ARPH*. To each articulation is assigned a reactive agent that realizes, in parallel, a local task without a priori knowledge on the actions of the other agents. Each agent calculates the current position of the end-effector and attempts by tiny local movements to match that position with the desired one. Yamamoto and Yun [7] presented a work to plan a path coordinating motion and manipulation through a control algorithm for the mobile base so that the manipulator is always positioned at the preferred configuration. The mobile base uses the measured joint position information of the manipulator for its own motion planning. The authors in [8] proposed a platform independent

approach for mobile manipulation and coordinated trajectory following. Given a path for the end-effector to follow, another path is planned for the mobile base in such a way that it is feasible. The mobile base and the end-effector follow their respective reference trajectories according to control algorithms.

This paper focuses on the problem of following operational trajectories for non-holonomic differentially-driven wheeled 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 has to avoid the obstacles present in the environment.

The outline of the paper is as follows. The next section of the paper describes the configuration of the experimental robotic system and presents the kinematic models of the mobile manipulator. Section three describes the multi-agent architecture proposed to control mobile manipulators. Section four exposes the protocol diagram of the mission of operational trajectory following by the end-effector of the robot. This section presents also the different parameters of the special case studied in this paper (the straight-line following). Section five presents experiments and discusses the main obtained results. Finally, a conclusion and future works are presented.

2. Architecture of the experimental mobile manipulator

The experimental robotic system, *RobuTER/ULM*, given by Fig. 1, consists of a rectangular differentially-driven wheeled mobile base on which is installed a manipulator.

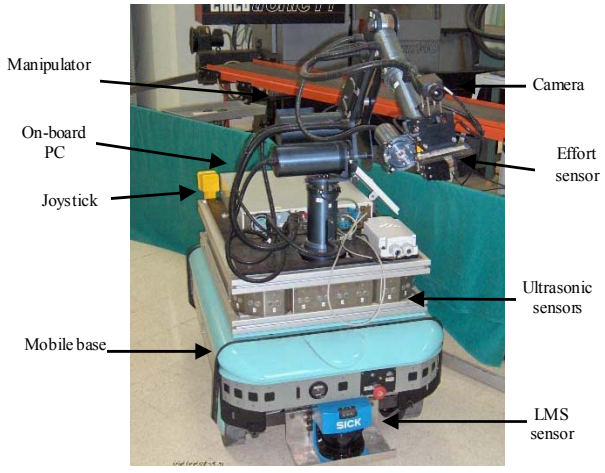


Fig. 1: *RobuTER/ULM* mobile manipulator

The non-holonomic mobile base, *RobuTER*, 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 in the front of the robot and an odometer sensor on each driven wheel.

The manipulator, *ULM*, is a six-dof (degree of freedom) ultra-light manipulator with two-fingered electrical gripper. All the joints of the manipulator are rotations. The manipulator is equipped with incremental position sensor on 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 with an acquisition card. Images are transmitted to an off-board PC (for processing) via a wireless video transmission system.

2.1. Main reference frames

For the kinematic analysis of the robot, the following assumptions are adopted:

- The mobile base moves on the plan.
- The wheels of the mobile base roll on the ground without sliding.
- The manipulator *ULM* is rigidly fixed to the mobile base *RobuTER*.

The kinematic analysis of the robot needs to focus on the following reference frames and the transformation matrices (Fig. 2) [9]:

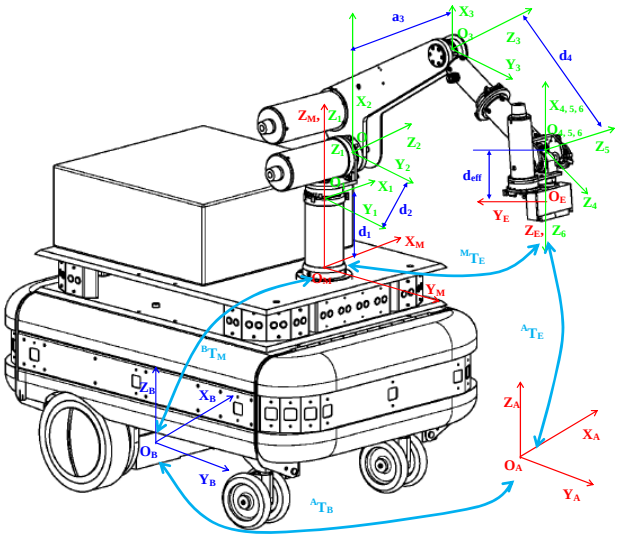


Fig. 2: Reference frames of *RobuTER/ULM* and the corresponding transformation matrices

- $R_A = (O_A, \overrightarrow{x_A}, \overrightarrow{y_A}, \overrightarrow{z_A})$: The absolute frame.

- $R_B = (O_B, \overrightarrow{x_B}, \overrightarrow{y_B}, \overrightarrow{z_B})$: The mobile base frame.
- $R_M = (O_M, \overrightarrow{x_M}, \overrightarrow{y_M}, \overrightarrow{z_M})$: The manipulator frame.
- $R_E = (O_E, \overrightarrow{x_E}, \overrightarrow{y_E}, \overrightarrow{z_E})$: The end-effector frame.
- $R_k = (O_k, \overrightarrow{x_k}, \overrightarrow{y_k}, \overrightarrow{z_k})$: It is attached to the joint k.
- ${}^{A}T_B$: This matrix defines R_B in R_A.
- ${}^{B}T_M$: This transformation matrix defines R_M in R_B.
- ${}^{M}T_E$: The transformation matrix defining R_E in R_M.
- ${}^{A}T_E$: The matrix defining R_E in R_A.
- ${}^{M}T_E$: The transformation matrix defining R_E in R_M.
- ${}^{k-1}T_k$: It defines R_k in R_{k-1}.
- ${}^{M}T_1$: This matrix defines R_1 in R_M.
- ${}^{6}T_E$: The transformation matrix defining R_E in R_6.

2.2. Kinematic analysis of the ULM manipulator

The position coordinates and orientation angles of the end-effector are computed in R_M by (1) following the *Modified Denavit-Hartenberg* (*MDH*) representation [10].

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

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

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

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 [9].

Table 1
MDH parameters and joints limits of *ULM*

k	α_k (°)	d_k (mm)	θ_k	a_k(mm)	Q_{Min}(°)	Q_{Max}(°)
1	0	d_1=290	θ_1	0	-95	96
2	90	d_2=108.49	θ_2	0	-24	88
3	-90	d_3=113	0	a_3=402	–	–
4	90	0	θ_3	0	-2	160
5	90	d_4=389	θ_4	0	-50	107
6	-90	0	θ_5	0	-73	40
E	90	d_{eff}=220	θ_6	0	-91	91

2.3. Differential kinematic analysis of the mobile base

The kinematic model of a non-holonomic differentially-driven mobile base can be described, in R_A, by three parameters: X_B, Y_B and θ_B (the Cartesian coordinates and the orientation angle). During its motion, the position coordinates and the orientation angle of the mobile base is given, by odometry, in real time, as shown in Fig. 3 where [9]:

- P_k, P_{k+1}: Current and next position of the mobile base.
- ϕ_k: Yaw angle at position P_k.
- ω_{Lk}, ω_{Rk}: Left and right angular velocity at position P_k.
- R_k: Steering radius of the mobile base at P_k.
- L: Half distance between the two driven wheels.
- r: Diameter of the driven wheels.
- ΔD_{Lk}, ΔD_{Rk}: Left and right elementary advances of the mobile base.
- (X_{bk}, Y_{bk}), (X_{bk+1}, Y_{bk+1}): Current and next position coordinates of the mobile base.
- θ_k, θ_{k+1}: Current and next orientation angle of the mobile base.
- $\Delta\theta_k$: Elementary rotation of the mobile base.
- ΔD_k: Elementary displacement of the mobile base.

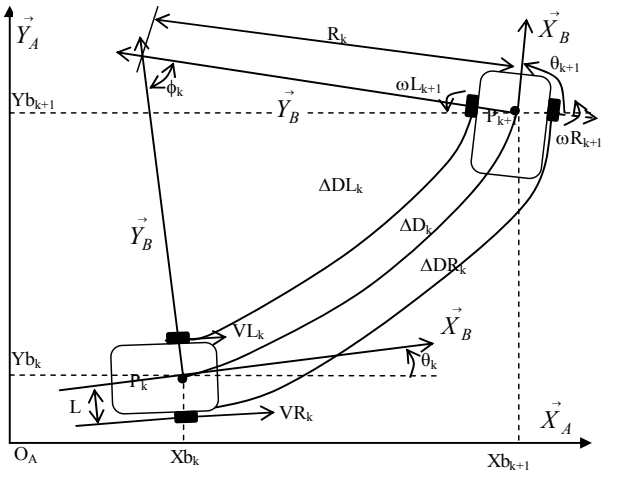


Fig. 3: Different parameters of the motion of a differentially-driven mobile base

$\Delta\theta_k$ is given by (3):

$$ \Delta\theta_k = \frac{\Delta D_{RK} - \Delta D_{LK}}{2L} \qquad (3) $$

ΔD_k is given by (4):

$$ \Delta D_k = \frac{\Delta D_{Rk} + \Delta D_{Lk}}{2} \qquad (4) $$

θ_{k+1} is given by (5):

$$ \theta_{k+1} = \theta_k + \Delta\theta_k \qquad (5) $$

The next position of the mobile base is given by (6):

$$ \begin{pmatrix} Xb_{k+1} \\ Yb_{k+1} \end{pmatrix} = \begin{pmatrix} Xb_k \\ Yb_k \end{pmatrix} + \Delta D_k . \begin{pmatrix} \cos(\theta_{k+1}) \\ \sin(\theta_{k+1}) \end{pmatrix} \qquad (6) $$

For the differential kinematic model of the mobile base:

$$V_{Rk} = -r.\omega_{Rk}=(R_k - L).\omega_k \quad (7)$$
$$V_{Lk} = r.\omega_{Lk}=(R_k + L).\omega_k \quad (8)$$

From (7) and (8), ω_k and V_k are given as follows:

$$w_k = r.\frac{w_{Lk} + w_{Rk}}{2.L} \quad (9)$$
$$V_k = \frac{V_{Rk} + V_{Lk}}{2} = r.\frac{w_{Lk} - w_{Rk}}{2} \quad (10)$$

The differential kinematic model of the mobile base is given by (11):

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta_k & 0 \\ \sin\theta_k & 0 \\ 0 & 1 \end{pmatrix} . \begin{pmatrix} V_k \\ w_k \end{pmatrix} \quad (11)$$

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 (12):

$${}^{A}T_E = {}^{A}T_B * {}^{B}T_M * {}^{M}T_E \quad (12)$$

${}^{A}T_B$ is given by (13) where (X_B, Y_B, Z_B) are the Cartesian coordinates of O_B in R_A.

$${}^{A}T_B = \begin{pmatrix} \cos\theta_B & -\sin\theta_B & 0 & X_B \\ \sin\theta_B & \cos\theta_B & 0 & Y_B \\ 0 & 0 & 1 & Z_B \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (13)$$

${}^{B}T_M$ is given (14) where (X_M, Y_M, Z_M) are the Cartesian coordinates of O_M in R_B.

$${}^{B}T_M = \begin{pmatrix} 1 & 0 & 0 & X_M \\ 0 & 1 & 0 & Y_M \\ 0 & 0 & 1 & Z_M \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (14)$$

For *RobuTER/ULM*, Z_B=120mm, X_M=30mm, Y_M=0mm and Z_M=520mm [9].

3. Multi-agent control architecture

Fig. 4 shows the multi-agent architecture proposed in [11] to control mobile manipulators.

The architecture consists of six agents (*i*) *Supervisory* agent (*SA*), (*ii*) *Local Mobile Robot* agent (*LMRA*), (*iii*) *Local Manipulator Robot* agent (*LARA*), (*iv*) *Vision System* agent (*VSA*), (*v*) *Remote Mobile Robot* agent (*RMRA*) and (*vi*) *Remote Manipulator Robot* agent (*RARA*). Each agent models a principal function of the mobile manipulator and manages a different sub-system. In addition, a mechanism connecting the four capacities (*i*) *Supervision*, (*ii*) *Perception*, (*iii*) *Decision* and (*iv*) *Action* corresponds to each agent of the architecture. More details are given in [11].

The first four agents (*SA*, *LMRA*, *LARA* and *VSA*) are installed on an off-board PC. The other agents (*RMRA* and *RARA*) are installed on the on-board PC of the robot. In addition, the agents of the architecture are implemented as a set of concurrent threads in order to be able to respond asynchronous and external events, and to deal with requests, as soon as possible, according to the dynamics of the robot. Furthermore, the agents of the architecture communicate via sockets using the TCP/IP protocol [9].

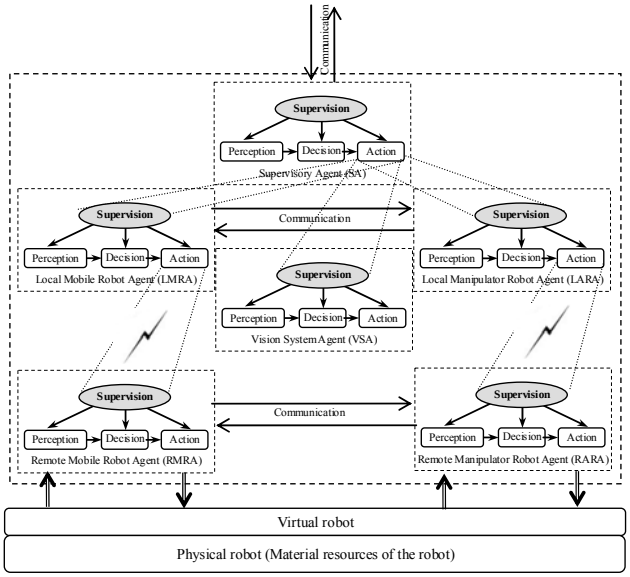


Fig. 4: Multi-agent control architecture of mobile manipulators

The *Supervision* capacity is a virtual entity whose role is to select modules which result in the necessary set of operations facing a given situation. The other capacities are explained in what follows:

3.1. SA

SA receives the mission to be carried out by the robot, then, decides on its feasibility according to the status and the availability of the required resources (*Perception* + *Decision*). If the mission is accepted,

SA distributes it on the local agents (*LMRA*, *LARA*, *VSA*) in order to establish a coordinated operations plan for execution (*Action*).

3.2. LMRA/LARA

LMRA/*LARA* receives the information on the environment of the mobile base/manipulator and analyzes reports from *RMRA/RARA* on the execution of the sent operations (*Perception*). In addition, *LMRA*/*LARA* cooperates with the other local agents (*LARA/LMRA*, *VSA*) in order to elaborate a coordinated operations plan for the execution of the assigned mission according to the received information and the status of the other agents (*Decision*). Finally, this agent sends operationsto *RMRA/RARA* for execution (*Action*).

3.3. VSA

VSA observes the environment of the robot by the vision system (*Perception*), extracts useful and required information for the execution of the assigned mission from captured images (*Decision*) and, sends the extracted information to the corresponding agent (*Action*).

3.4. RMRA

RMRA scans the proprioceptive and exteroceptive sensors equipping the mobile base. After that, it sends useful information to *LMRA* in order to maintain an up-to-date representation on the environment of the mobile base (*Perception*). In addition, *RMRA* ensures the control of the mobile base by sending instructions to its actuators (*Decision* + *Action*) according to the principle detailed in [9].

3.5. RARA

RARA collects information of sensors equipping the manipulator and sends it to *LARA* for processing (*Perception*). Furthermore, this agent ensures the control of the manipulator by sending instructions to its actuators (*Decision* + *Action*) according to the principle detailed in [9].

4. Protocol diagram of the operational trajectory following

The core thinking of using a multi-agent system to control mobile manipulators is that of realizing cooperation between the manipulator, the mobile base and the sensors system in order to accomplish the assigned mission.

The cooperative mission studied in this section is to follow a given operational trajectory by the end-effector of the robot.

4.1. Straight-line following

As shown in Fig. 5, 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 $P_f(X_f, Y_f, Z_f)$.

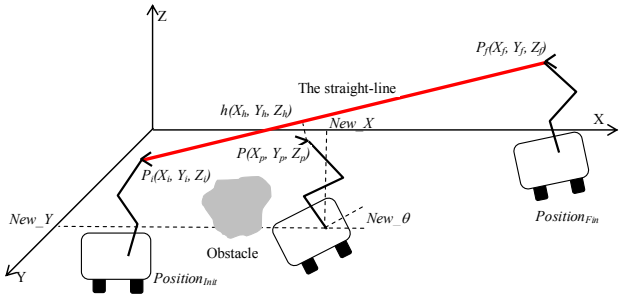


Fig. 5: The straight-line following mission and its different parameters

In order to compute the imposed positions (*Targets*) to be reached by the end-effector of the robot [12]:

- $[P_i, P_f]$: the segment connecting P_i to P_f.
- $p(X_p, Y_p, Z_p)$: the current position coordinates of the end-effector of the mobile manipulator.
- $h(X_h, Y_h, Z_h)$: the projection of p on $[P_i, P_f]$ (the next *Target* to be reached by the end-effector of the robot).
- (*Odo_X, Odo_Y*): current odometer sensors data of the mobile base.
- *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.
- *IKM*: the *Inverse Kinematic Model* of the manipulator computed as shown in [13].
- $m(X_m, Y_m, Z_m)$: a given point in the space.

$$m \in [P_i, P_f] \Leftrightarrow \begin{cases} X_m = t * (X_f - X_i) + X_i \\ Y_m = t * (Y_f - Y_i) + Y_i \\ Z_m = t * (Z_f - Z_i) + Z_i \\ t \in [0, 1] \end{cases} \quad (15)$$

$\vec{hp}$ is orthogonal to $\vec{P_iP_f}$ so:

$$(X_h-X_p)*(X_f-X_i)+(Y_h-Y_p)*(Y_f-Y_i)+(Z_h-Z_p)*(Z_f-Z_i)=0 \quad (16)$$

From (15) and (16), the coordinates of h is given by (17):

$$\begin{aligned} X_h &= t*(X_f - X_i)+X_i \\ Y_h &= t*(Y_f - Y_i)+Y_i \\ Z_h &= t*(Z_f - Z_i)+Z_i \\ t &= \frac{(X_f - X_i)*(X_p - X_i)+(Y_f - Y_i)*(Y_p - Y_i)+(Z_f - Z_i)*(Z_p - Z_i)}{(X_f - X_i)^2+(Y_f - Y_i)^2+(Z_f - Z_f)^2} \end{aligned} \quad (17)$$

The positioning error of the end-effector is computed by (18):

$$Err = \sqrt{(X_h - X_p)^2 + (Y_h - Y_p)^2 + (Z_h - Z_p)^2} \quad (18)$$

4.2. Protocol diagram

First of all, *SA* tests if all the positions (*Targets*) of the imposed operational trajectory could be reached by the end-effector of the robot. If one of these *Targets* can not be reached, *SA* displays an error message (*Mission impossible*). Otherwise, a request (*Move Manipulator (Target)*) is sent to *LARA* which tests if *Target* belongs of its current workspace [12]:

- *Target belongs of the current workspace of the manipulator*: *LARA* computes the different Q_i(i=1...dof) of *Target* using the *IKM* of the manipulator and sends a request (*Move Manipulator* ($Q_1 \ldots Q_{dof}$)) to *RARA*. Receiving this request, *RARA* executes the movement and, upon arrival, replies by *Position OK*. At the end, *LARA* informs *SA* that *Target is reached successfully*.
- *Target does not belong of the current workspace of the manipulator*: *LARA* sends a request (*Move Base (Target)*) to *LMRA* which computes *Position* for the mobile base (*Position*(X_B, Y_B, θ_B) is computed so that *Target* belongs of the new workspace of the manipulator) and sends a request (*Move Base(Position)*) to *RMRA*. *RMRA* moves the mobile base towards the desired position while avoiding obstacles present in the environment. During the motion of the mobile base, *RMRA* sends (*Odo_X, Odo_Y*) to *LMRA*. This latter agent computes (*New_X, New_Y, New_θ*) and sends them to *LARA*. Receiving this information, the first case is obtained.

In both cases, *LARA* and *RARA* attempt to position the end-effector of the robot, as close as possible, at the desired configuration in order to minimize the positioning error. *LARA* computes $p(X_p, Y_p, Z_p)$ and $h(X_h, Y_h, Z_h)$ in R_A. After that, *LARA* computes the different $Q_i(i=1\ldots dof)$ corresponding to $h(X_h, Y_h, Z_h)$ and sends a request (*Move Manipulator* $Q_i(i=1\ldots dof)$) to *RARA*. At the end of the movement, *RARA* replies by *Position OK*. These previous operations continues until the arrival to *Position*(X_B, Y_B, θ_B).

More details are given in the protocol diagram of Fig. 6.

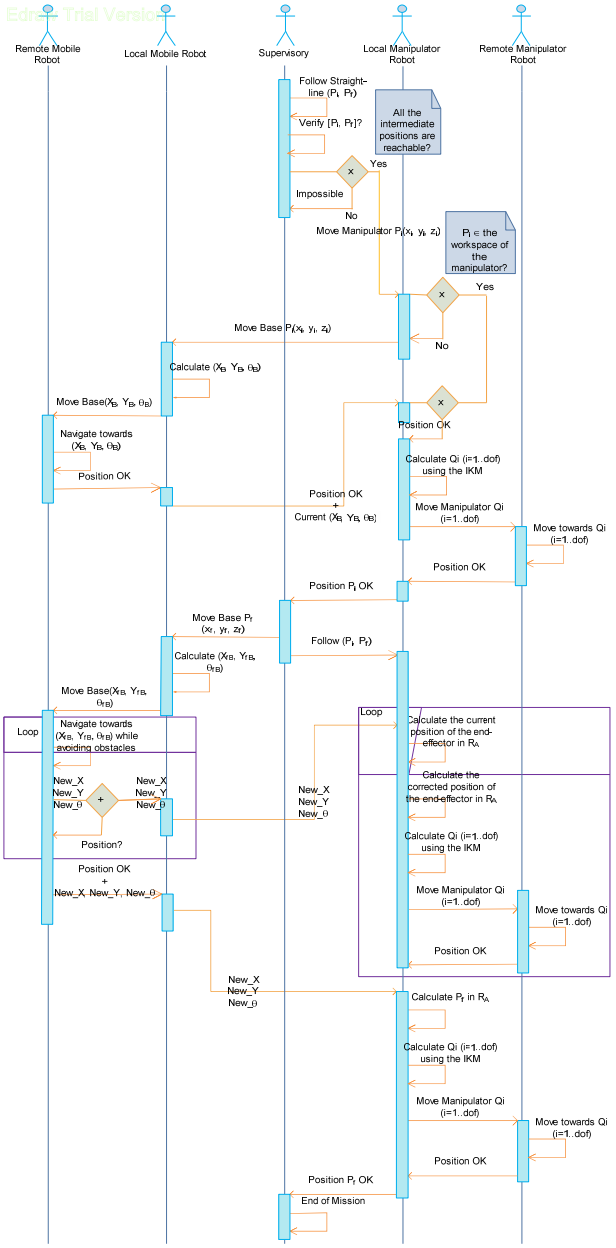


Fig. 6: Protocol diagram of the mission of following an imposed operational trajectory by the end-effector of a mobile manipulator

5. Experimental results

The straight-line following mission is implemented on *RobuTER/ULM*. (17) is used to generate the *Target* positions so that the end-effector of the mobile manipulator follows the desired straight-line.

All the positions are given in R_A. $P_i(X_i, Y_i, Z_i)$=(-691.72mm, -108.49mm, 1128.62mm) and $P_f(X_f, Y_f, Z_f)$=(-2408.17mm, -108.49mm, 1472.30mm). Therefore, the imposed operational trajectory consists of a straight-line of about 2000mm (1716.45mm) with a slope of about 350mm (343.68mm) [12].

The initial position coordinates and orientation angle 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 the initial position, the initial joints angles $(Q_{1Init}, Q_{2Init}, Q_{3Init}, Q_{4Init}, Q_{5Init}, Q_{6Init})$ = (0°, 60°, 0°, 0°, 32°, 0°) [12].

The final position and orientation 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°). Two cases are considered [12]:

5.1. The environment of the robot is free

Here, no obstacles are considered in the environment. The real operational trajectory followed by the end-effector and the imposed one are shown on Fig. 7. The red line represents the trajectory imposed to the end-effector of the robot. The blue one represents the real trajectory followed by the end-effector.

The real joints variations and the desired trajectories for some joints are shown in Fig. 8. The motion of the mobile base, given by Fig 9, consists of a straight-line connecting $Position_{Init}$ to $Position_{Fin}$ [12].

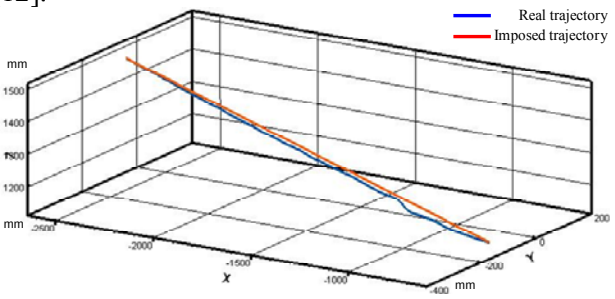


Fig. 7: Operational trajectory followed by the end-effector of the robot and the real one

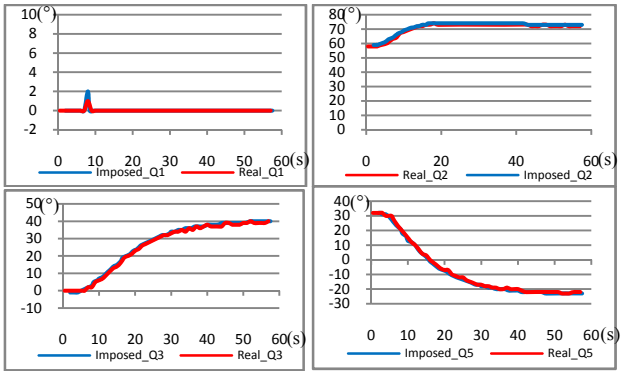


Fig. 8: Joints variations and the desired trajectories of some joints

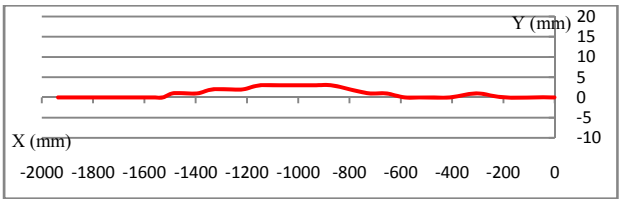


Fig. 9: Real trajectory of the mobile base in case of free environment

The execution of this mission took about 58 seconds. The average positionning error is about 6.67mm. The maximum positionning error is about 23.75 mm at $X_p \approx$-1000 mm [12].

5.2. The environment of the robot is cluttered

The second case is more difficult than the previous one. The mobile base has to avoid an obstacle present in the environment (at position (x, y)=(-1100mm, 0mm)) while the end-effector has to be always at the desired configuration (on the straight-line) or, at least, be as near as possible to that configuration [12].

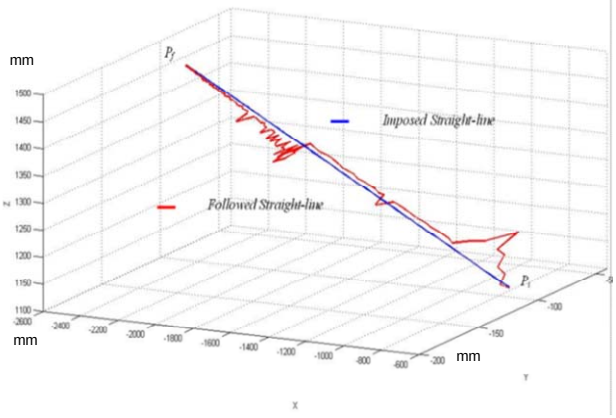


Fig. 10: Real trajectory and Imposed trajectory

The real operational trajectory followed by the end-effector and the imposed trajectory for the end-

effector are shown on Fig. 10. The blue line represents the imposed trajectory while the red one represents its real trajectory. The real joints variations (1 … 6) are shown on Fig. 11 [12].

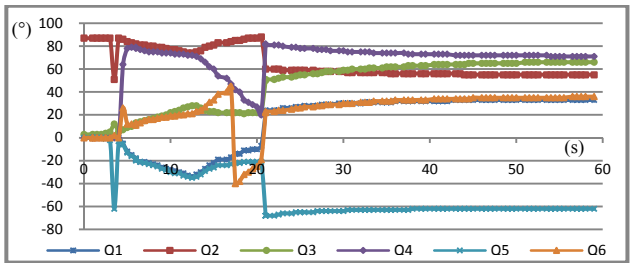


Fig. 11: Real joints variations (1…6)

For this case, the execution of the mission took about 60 seconds. The maximum positionning error is 141.83 mm observed at X_p≈-1200 mm. The average error is 43.64 mm. These errors are caused by the slow movement of the manipulator compared to the velocity of the mobile base [12].

The obtained results in this case are not acceptable. Another experiment is performed under the same conditions with limited velocities for the mobile base.

The operational trajectory followed by the end-effector and the imposed one are shown on Fig. 12. The real joints variations and the desired trajectories for some joints (1, 2, 3 and 5 respectively) are shown on Fig. 13. Fig. 14 shows the real trajectory of the mobile base [12].

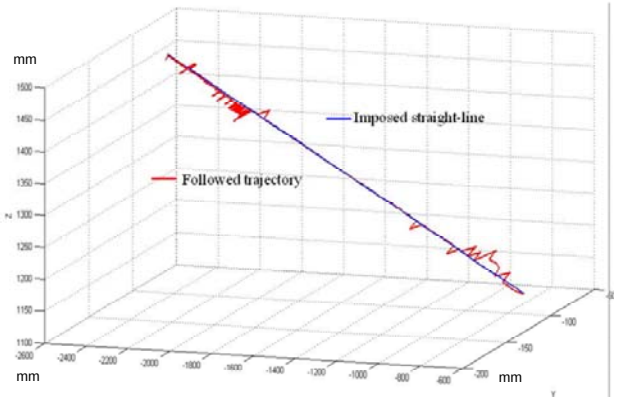
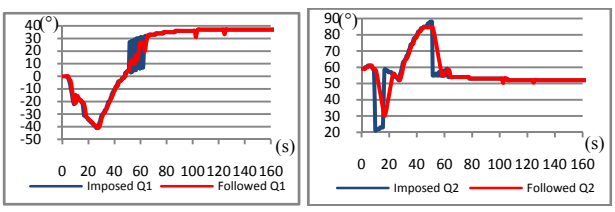


Fig. 12: Operational trajectory followed by the end-effector of the robot



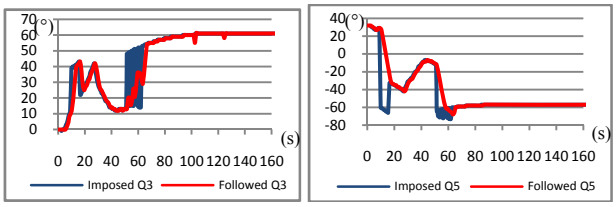


Fig. 13: Joints variations and desired trajectories of some joints

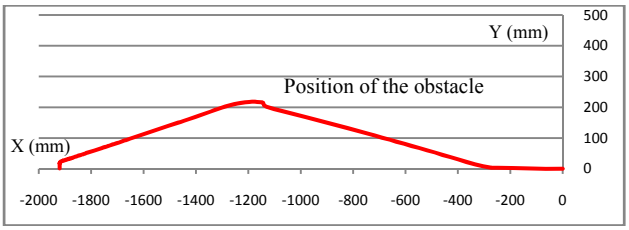


Fig. 14: Real trajectory of the mobile base (an obstacle present in the environment of the robot)

The maximum positioning error is 24.43 mm. It corresponds to X_p≈-1100 mm. The average error is about 3.41 mm. The execution time of this mission is about 160 seconds. This is not due to the avoidance of the obstacle present in the environment only, but also to the limited velocity of the mobile base [12].

After the beginning of the motion of the robot and at x≈-900mm (that corresponds to the obstacle avoidance) and during the recovery phase, the real trajectory coincides with that of the imposed one. The final joints angles (Q_{1Fin}, Q_{2Fin}, Q_{3Fin}, Q_{4Fin}, Q_{5Fin}, Q_{6Fin}) = (37°, 52°, 61°, 73°, -57°, 28°) [12].

5.3. Discussion of results

The errors observed during the experiments show that it is difficult to perfectly follow the desired straight-line. These errors are caused by various reasons [12]:

- The first reason 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 quick motion of its manipulator.
- The second reason is that the error on the estimated position coordinates and orientation angle of the mobile base during its motion, computed by odometry (*New_X, New_Y, New_θ*), affects the tip position of the end-effector directly. To absorb this error, the manipulator should quickly move to adjust itself when the error is detected. A localization approach is also envisaged in order to minimize that error.

- The final reason for this error is the difference between the velocity of the motion of the mobile base and that of the manipulator. The low velocity of the motion of the manipulator during the motion of the mobile base causes a delay in the positioning of the end-effector. This problem can be solved by increasing the velocity of the manipulator according to that of the mobile base.

6. Conclusion

This paper has proposed a strategy to effectively deal with the problem of following imposed operational trajectories by the end-effector of a non-holonomic differentially-driven wheeled mobile manipulator.

The robot is controlled by a multi-agent architecture that consists of six agents (*i*) *Supervisory*, (*ii*) *Local Mobile Robot*, (*iii*) *Local Manipulator Robot*, (*iv*) *Vision System*, (*v*) *Remote Mobile Robot* and (*vi*) *Remote Manipulator Robot*. The first four agents are installed on an off-board PC while the two last agents are installed on the on-board PC of the robot.

The controller was applied successfully to the *RobuTER/ULM* mobile manipulator in order to follow a straight-line connecting an initial position $P_i(X_i, Y_i, Z_i)$ to a final position $P_f(X_f, Y_f, Z_f)$. Two cases were studied (*i*) free environment and (*ii*) cluttered environment. In both cases, many experiments have been carried out and a result could be obtained for each one. The average positioning errors are acceptable.

To realize the operational trajectory following, one of the biggest problems is the accumulated error of the estimated position of the mobile base. This error affects the position accuracy of the end-effector of the robot. Therefore, the manipulator should have a capability to quickly adjust its position when the mobile base detects positioning errors. However, the controller was shown to be relatively effective when the robot moves with small velocities.

In future works, the performances and the robustness of the proposed approach should be shown and discussed through examples of other types of trajectories (zigzag, sinusoidal, rectangular, ellipsoidal, … trajectories).

References

1. Xu, D., Hu, H., Calderon, C. A. A., Tan, M.: *Motion Planning for a Mobile Manipulator with Redundant DOFs*, The International Journal of Information Technology, Vol. 11 N° 11, 2005, pp. 01-10.
2. Sugar, T., Kumar, V.: *Decentralized Control of Cooperating Mobile Manipulators*, The International Conference on Robotics and Automation (ICRA'98), May 16-20, 1998, Leuven, Belgium, pp. 2916-2921.
3. Nagatani, K., Hirayama, T., Gofuku, A., Tanaka, Y.: *Motion planning for mobile manipulator with keeping manipulability*, The International conference on Intelligent Robots and Systems (IROS2002), October 2002, Lausanne, Switzerland, pp. 1663 - 1668.
4. Alfaro, C., Ribeiro, M. I., Lima, P.: Smooth Local Path Planning for a Mobile Manipulator, Proceedings of ROBOTICA 2004, *The 4th Portuguese Robotics Festival*, April 2004, Porto, Protugal, pp. 127-134.
5. Erden, M. S., Leblebicioglu, K., Halici, U.: *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, Vol. 39 N° 3, 2004, pp. 287-306.
6. Colle, E., Nait Chabane, K., Delarue, S., Hoppenot, P.: Comparaison d'une méthode classique et d'une méthode utilisant la coopération homme-machine pour exploiter la redondance de l'assistant robotisé, *The 4th HANDICAP 2006 Conference*, 7-9 June 2006, Paris, France.
7. Yamamoto, Y., Yun, X.: *Coordinating locomotion and manipulation of a mobile manipulator*, IEEE Transaction on Automatic Control, Vol. 39 N° 6, 1994, pp. 1326-1332.
8. Egerstedt, M., Hu, X.: Coordinated Trajectory Following for Mobile Manipulation, *The International Conference on Robotics and Automation (ICRA'00)*, April 2000, California, USA, pp. 3479-3484.
9. Hentout, A., Bouzouia, B., Toumi, R., Toukal, Z.: *Multi-agent Architecture for Telerobotics of Mobile Robot Manipulators*, Journal of Automation and System Engineering, Vol. 3 N° 4, 2009.
10. Khalil, W., Kleinfinger, J. F.: A new geometric notation for open and closed-loop robots, *The International Conference on Robotics and Automation (ICRA'86)*, April 1986, San Francisco, USA, pp. 1175-1180.
11. Hentout, A., Bouzouia, B., Toukal, Z.: *Multi-agent Architecture Model for Driving Mobile Manipulator Robots*, The International Journal of Advanced Robotic Systems, Vol. 5 N° 3, 2008, pp. 257-269.
12. Hentout, A., Akli, I., Bouzouia, B., Benbouali, R., Ouzzane, E., Bouskia, M. A.: Multi-agent Control Architecture of Mobile Manipulators: Following an Operational Trajectory, *The 3rd IEEE International Conference on Signals, Circuits and Systems (SCS'09)*, November 05-09, 2009, Jerba, Tunisia.
13. Gorla, B., Renault, M.: *Modèles des robots manipulateurs : application à leurs Commandes*, Editions Cepadues, 1984.