Multi-agent Control Approach for Autonomous Mobile Manipulators
Determination of the Best Footsteps Combination

Messous Mohamed Ayoub, Hentout Abdelfetah and Bouzouia Brahim
Centre for Development of Advanced Technologies (CDTA),
Division of Computer-Integrated Manufacturing and Robotics (DPR),
BP 17, Baba Hassen, Algiers 16303, Algeria

Keywords: Multi-agent Control, Mobile Manipulators, Best Footsteps Combination, *RobuTER/ULM*, Simulation, *JADE*.

Abstract: This paper presents our ongoing efforts toward the development of a distributed multi-agent framework for autonomous control of mobile manipulators. The proposed scheme assigns a reactive agent (*Joint agent*) to control each articulation of the manipulator, a hybrid agent (*Mobile base agent*) for the control of the mobile base, and a *Supervisory agent* to coordinate and synchronize the work of all the precedent agents. Each *Control agent* implements a *Simulation-verification* technique, in order to optimize, locally and independently from the other agents, the value of a predefined *Objective function* (f_{Obj}).
The paper illustrates, also, the methodology we have followed to determine the best combination among possible footsteps for the *Control agents* of the system. This combination will be the input, among others, of the procedure seeking the optimal solution bringing the position of the end-effector of the mobile manipulator as close as possible to the imposed *Target position*. For this aim, different simulation scenarios are described and carried out, with and without considering breakdowns of some articulations of the manipulator or the mobile base. For the evaluation of the obtained solutions and the selection of the best footsteps combination, we have considered two criteria (*i*) f_{Obj} values and (*ii*) the number of iterations.

1 INTRODUCTION

An intelligent mobile manipulator is a kind of intelligent system, which can autonomously perform scheduled tasks in complex, unknown and changing environments with sensing, perceptive, knowledge acquisition, learning and inference capabilities, decision making and acting ability (Nebot et al., 2004), by using only its limited physical and computational resources with a reduced human intervention (Medeiros, 1998).

The intelligence of a mobile manipulator is constructed as an integrated system of many special software subsystems with different functions such as manipulation, locomotion, vision, planning, etc. To realize the global task by the robot, the different subsystems need to cooperate with each other and compete for limited resources. Thus, it is very important to build a high performance autonomous robot control system to make these subsystems harmoniously work together to achieve this goals.

The control systems are mainly divided into two different approaches (*i*) in the single-agent system, the only agent has to perform all the actions (sensing, planning, control, etc.) (*ii*) in contrast, the multi-agent system (*MAS*) decomposes the large system into many small and distinct agents (Duhaut, 1999) (Erden et al.,2004) (Delarue et al., 2007).

MAS offer simple solutions and benefit of all the advantages of distributed problem solving. This perspective made it possible to consider the system as composed of simple modules, which gave an easier way to design the whole system. In addition, the need for massy mathematical models of the robot (*Inverse kinematics model* and *differential-equation-solvers*) is overcome (Duhaut, 1999). Therefore, there is a considerable decrease in design effort and computation time compared to single-agent system. Finally, with such a usage of *MAS*, the control is more flexible to be applied to any robot (mobile, manipulator or mobile manipulators). Toward modularity, complexity and advanced adaptive behaviors of intelligent robot system (Nebot et al., 2004), we have adopted the multi-agent paradigm.

Mohamed Ayoub M., Abdelfetah H. and Brahim B..
Multi-agent Control Approach for Autonomous Mobile Manipulators - Determination of the Best Footsteps Combination.
DOI: 10.5220/0005064403400347
In *Proceedings of the 11th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2014), pages 340-347
ISBN: 978-989-758-039-0

Our ongoing efforts concern the development of a generic distributed multi-agent framework for autonomous control of mobile manipulators. The proposed scheme assigns a reactive agent (*Joint agent*) to control each articulation of the manipulator, a hybrid agent (*Mobile base agent*) to control the mobile base, and a *Supervisory agent* to coordinate and synchronize the work of all the precedent agents. Each *Control agent* (*Joint* agent, *Mobile base* agent) makes a virtual movement, in all the possible directions with different footsteps (*Joint footstep, Base Translation footstep, Base Rotation footstep*), locally and independently from the other agents. After that, the agent computes an *Objective function* (f_{Obj}) between the current position of the end-effector and the imposed *Target* position. The retained movement, for each agent, is that optimizing f_{Obj}.

The purpose of the current paper is to determine the best combination of footsteps, for all the *Control agents*, in order to reach the optimal solution bringing the end-effector position as close as possible to the *Target* position. For this aim, different simulation scenarios are described, with and without considering breakdowns of some articulations of the manipulator or the mobile base.

The rest of the paper is organized as follows. Section two describes the proposed multi-agent scheme for autonomous control of mobile manipulators. The interaction between the agents of the system is detailed, also, in this section. Section three shows the implementation of the proposed approach on *RobuTER/ULM* mobile manipulator. It determines, moreover, the best combination of footsteps via the obtained simulation results. Section four concludes the paper and draws-up future works.

2 CONTROL APPROACH

Mobile manipulators are composed of two heterogeneous very unlike sub-systems (a mobile base and a manipulator). Consequently, different controlling entities are solicited to ensure a modular, yet robust control scheme. The interactions among these entities assure the required cognitive behaviors for the robot to accomplish different tasks. Through a class diagram, Figure 1 shows a generalized view for the proposed scheme, in which two kinds of agents can be distinguished:

2.1 *System* Agents

This kind of agents is intended for the treatment of data issued from the different sensors equipping the robot. It can assure the main functionalities related to or used in the control process such as (*i*) vision module, (*ii*) robot localization module, (*iii*) target localization module, etc. However, this list is not exhaustive; other modules would be thought and implemented progressively.

2.2 *Robot* Agents

We focus mainly, in our study, on the second type of agents, which are dedicated to the control process itself. We do have though two sub-classes:

(i) *Control agents*: they are involved in the computation of the commands to be sent to the robot (*Joints agents* and *Mobile base agent*).

(ii) *Supervisory agent*: it is responsible for the synchronization and the coordination between the *Control agents*, and for the selection of the most fitted choices.

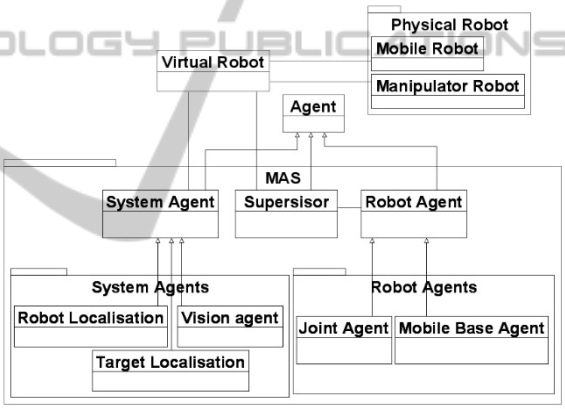

Figure 1: Class diagram of the proposed system.

Each *Control agent* receives, from the *Supervisory agent*, the initial situations of the robot ($Configuration_{Init}(q_1, \ldots, q_{dof})_{Init}$ and $Base_{Init}(x_B, y_B, \theta_B)_{Init}$) and the imposed coordinates of the $Target(x_T, y_T, z_T)$ to be reached. The *Control agent* drives its corresponding mechanical subset independently from the other agents, trying to bring the end-effector as close as possible to *Target*. Throughout this process, the current position of the end-effector $Effector(x_E, y_E, z_E)$ is computed using the *Direct Kinematic Model* (*DKM*) of the robot. This operation is rather straightforward and doesn't require any complex matrix inversion calculus in contrast of classical robotic approaches (Hentout et al., 2013). The computation of the *DKM* is given by (1) where:

- $Base(x_B, y_B, \theta_B)$: it represents the current situation of the mobile base.

- *Configuration*(q_1, ..., q_{dof}): it is the current configuration of the manipulator joints (*dof*: *degrees of freedom*).

$$Effector=DKM(Base, Configuration) \quad (1)$$

In the following, we describe the elementary movements for each *Control agent*, along with a short description of the corresponding behaviors. More details are given in (Hentout et al., 2014).

2.2.1 *Joint* Agents

Each articulation is controlled by a *Joint agent*, which allows two possible elementary movements. The following process, illustrated in Figure 2, is implemented to figure out the most fitted choice:

- A *Joint agent* makes a virtual rotation (*MoveUp*) in the positive direction ($Configuration_{Up}$) with a *Joint footstep*.
- The agent computes the objective function value f_{Obj} ($Distance_{Up}$) as shown in (2):

$$Distance_{Up}=F\_Objective(Base, Configuration_{Up}, Target) \quad (2)$$

- The agent repeats these two previous actions while changing the direction of the rotation (*MoveDown*, $Configuration_{Down}$, $Distance_{Down}$).

After comparing the two values ($Distance_{Up}$, $Distance_{Down}$) with the previous value of f_{Obj}, the *Joint agent* chooses the action to be made. For some cases, the best choice would be to stay still because neither of the two virtual movements would improve f_{Obj}. Subsequently, the selected movement will be sent, as a proposal (*Distance_Joint*, *New_Configuration*), to the *Supervisory agent.*

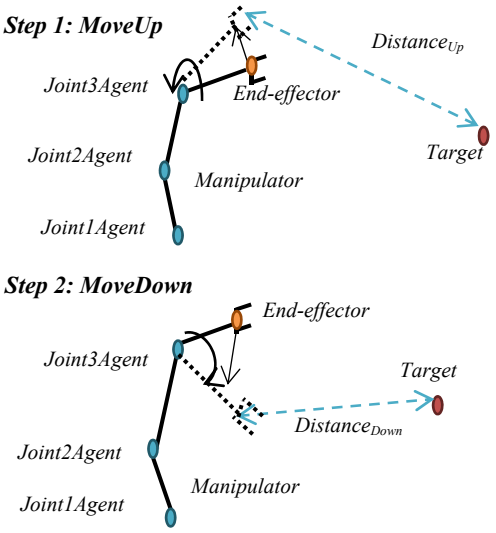

Figure 2: Elementary movements for each *Joint agent.*

2.2.2 *Mobile Base* Agent

For the *Mobile base agent*, we have defined four elementary movements as presented in Figure 3:

- The *Mobile base agent* makes a virtual forward movement (*MoveForward*) with a *Base Translation footstep* ($Base_{FW}$).
- The agent computes the new objective function value ($Distance_{FW}$).
- The *mobile base agent* repeats the previous actions for the other elementary (*MoveForward*, *TurnRight*, *TurnLeft*) movements (Table 1).

Table 1: Elementary movements for the *Mobile base agent.*

Elementary movements	New situation of the mobile base	Footsteps	f_{Obj} Values
MoveForward	$Base_{FW}$	Translation footstep	$Distance_{FW}$
MoveBackward	$Base_{BW}$		$Distance_{BW}$
TurnRight	$Base_{TR}$	Rotation footstep	$Distance_{TR}$
TurnLeft	$Base_{TL}$		$Distance_{TL}$

The Mobile base agent will choose, afterward, its local best choice, which will be the one optimizing f_{Obj} value amongst the four elementary movements and the actual position. The selected choice will be, finally, sent as a proposal (*Distance_Base*, *New_Base*), to the Supervisory agent.

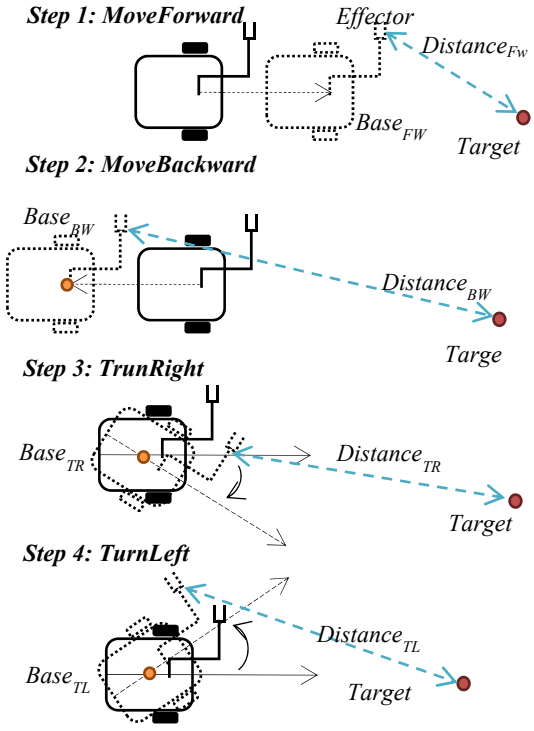

Figure 3: Elementary movements of the *Mobile base. agent.*

2.2.3 *Supervisory* Agent

This hybrid agent is responsible for the coordination and synchronization between the *Control agents*. After receiving the *Target* position from the human operator, the *Supervisory agent* verifies its reachability ($z_T \in [z_{Min}, z_{Max}]$ where z_{Min} and z_{Max} are respectively the minimum and maximum reachable height in the workspace of the robot). If not reachable, the agent displays a *Target unreachable error message* and terminates the process. Otherwise, it computes the initial situation of the end-effector ($Effector_{Init}$) and the initial value of the *Objective function* ($fObj\_{Init}$). After that, the *Supervisory agent* sends this information along with the initial situation of the mobile base ($Base_{Init}$) to the *Control agents*, and waits, then, for their proposals.

Once all the proposals are received, the *Supervisory agent* chooses the best one (best f_{Obj} value). The holder of this proposal will be expecting a *Contract* message to confirm the selection and to execute, thus, the proposed movement.

When f_{Obj} reaches a predefined optimum value ($f_{Obj}<\varepsilon$), the *Supervisory agent* terminates the process and sends a *Target reached successfully* message. Otherwise, the *Supervisory agent* reiterates the precedent steps and continues the process until reaching the goal (f_{Obj} is optimal).

2.3 Message Exchange Scheme

The interaction between the agents is implemented via a messages exchange protocol based upon the famous *Contract-net protocol* (Davis and Smith, 1983). The sequence diagram of Figure 4 gives a global vision of the whole interactions among the *Control agents* and the *Supervisory agent*. The different messages are defined as follows:

- *INFORM*: it is sent by the *Supervisory agent* to all the active *Control agents* at the beginning of each iteration. This message contains the current situation of the mobile base (*Base*), the current configuration of the manipulator (*Configuration*) and the current value of the objective function (f_{Obj}).
- *CFP (Call For Proposal)*: this message, sent after the precedent message by the *Supervisory agent*, contains the position of the *Target*. If we are dealing with a stationary *Target*, it is sent just once at the beginning of the process.

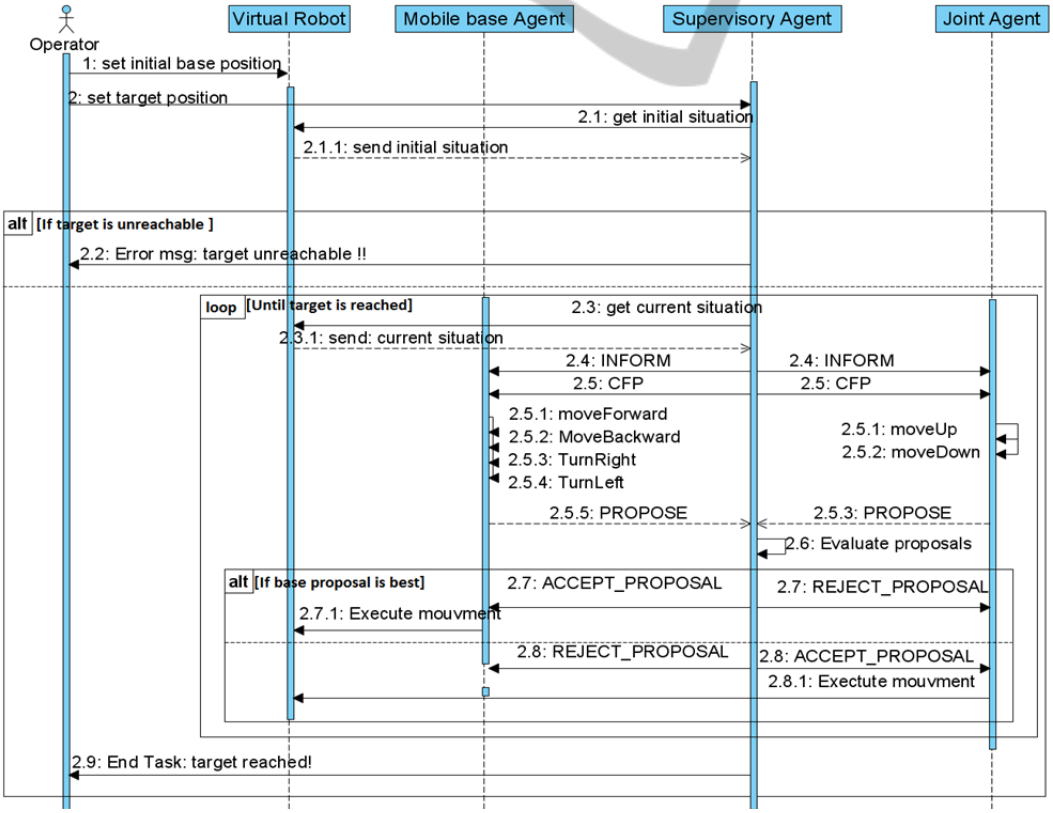

Figure 4: Sequence diagram for the whole system.

- *PROPOSE*: following the reception of a *CFP*, this message, sent by each *Control agent* to the *Supervisory agent*, comprises the best local proposition of the *Control agent* (*Distance_Joint*, *New_Configuration* / *Distance_Base*, *New_Base*).
- *ACCEPT_PROPOSAL/REJECT_PROPOSAL*: after receiving all the propositions from the *Control agents* (*PROPOSE*), the *Supervisory agent* selects the best choice and sends an *ACCEPT_PROPOSAL* message to the agent holding this proposition. All the other *Control agents* will receive a *REJECT_PROPOSAL* message.
- *ACK*: following the reception and the execution of the best selected movement by the chosen *Control agent*, an acknowledgment message is sent to the *Supervisory agent*.

3 EXPERIMENTAL WORK

We have adapted the proposed control system to the characteristics of our experimental robotic platform. Figure 5 presents a global view for the control structure of *RobuTER/ULM* mobile manipulator. The architecture involves a set of eight agents (Hentout et al., 2014):

- Six reactive *Joint agents* are assigned to control the six-dof *ULM* manipulator.
- One hybrid *Mobile base agent* to control the mobile base *RobuTER* of the robot.
- One hybrid *Supervisory agent* to coordinate and synchronize the precedent agents.

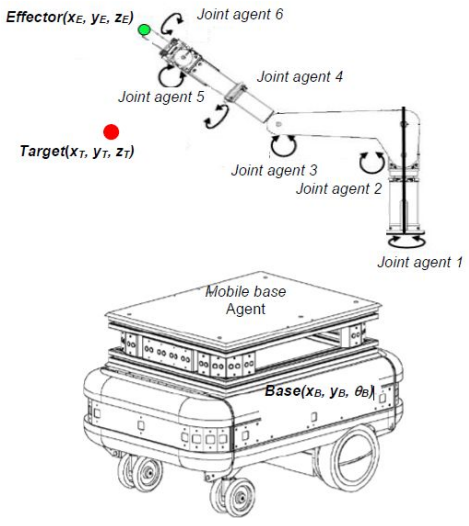

Figure 5: Control scheme of *RobuTER/ULM*.

In this work, the local objective of each agent is to reduce the quadratic distance between the current situation of the end-effector *Effector*(x_E, y_E, z_E), and the imposed position of the *Target*(x_T, y_T, z_T). This distance is computed as follows:

$$f_{Obj} = \sqrt{(x_T - x_E)^2 + (y_T - y_E)^2 + (z_T - z_E)^2} \quad (3)$$

JADE (*Java Agent DEvelopment Framework*) has been used as an implementation tool of the proposed approach. The main reason was the fact that *JADE* is one of the best modern agent open source platforms (Iñigo-Blasco et al., 2012). This framework provides basic middleware-layer functionalities which simplify the implementation of distributed applications using a software agent abstraction (Bellifemine et al., 2008) (Floroian and Moldoveanu, 2010).

3.1 Simulation Scenarios

Different simulation scenarios have been considered. They are intended to tune the performances of our approach facing different *Target* positions with different initial situations for the robot. To achieve this goal, we have implemented the graphical interface, shown in Figure 6, to be able to determine the finest combination of footsteps for the *Control agents*, which is the main contribution of this paper.

The considered tasks consist of bringing the end-effector of *RobuTER/ULM* to the final operational position *Target*. The parameters that define each task are (*i*) the initial situation of the mobile base $Base_{Init}$ (*ii*) the initial configuration of the manipulator $Configuration_{Init}$ and (*iii*) the dictated *Target*. For validation purposes, we have chosen the five tasks presented in Table 2. The distances are given in millimeters (mm) and the angles in degrees (°).

Table 2: Details of the considered validation tasks.

Task	$Configuration_{Init}$	$Base_{Init}$	*Target* (mm)	$f_{Obj\ Init}$(mm)
1	(0, 0, 0, 0, 0, 0)	(0, 0, 0)	(-330, -630, 1080)	1126.9129
2	(0, 0, 0, 0, 0, 0)	(0, 0, 0)	(-4260, 0, 665)	4698.9355
3	(0, 60, 0, 0, 32, 0)	(0, 0, 0)	(-2408, -108, 1472)	3114.8048
4	(0, 87, 0, 0, 5, 0)	(0, 0, 0)	(-2400, -63, 1325)	2946.8779
5	(0, 87, 0, 0, 5, 0)	(0, 0, 0)	(-2400, -67, 1320)	2946.8226

The following sub-sections illustrate the followed methodology in order to determine the best combination among possible footsteps, for the *Control agents* of the system. This combination will be the input, among others, of the procedure seeking the optimal solution bringing the position of the end-effector of the robot as close as possible to the imposed *Target position*. In this paper, we have

considered a static selection of footsteps. So, for each *Joint agent*, we have used an elementary rotation footstep (*Joint footstep*), whereas we identified two elementary footsteps for the *Mobile base agent* (*i*) *Base Translation footstep* and (*ii*) *Base Rotation footstep*.

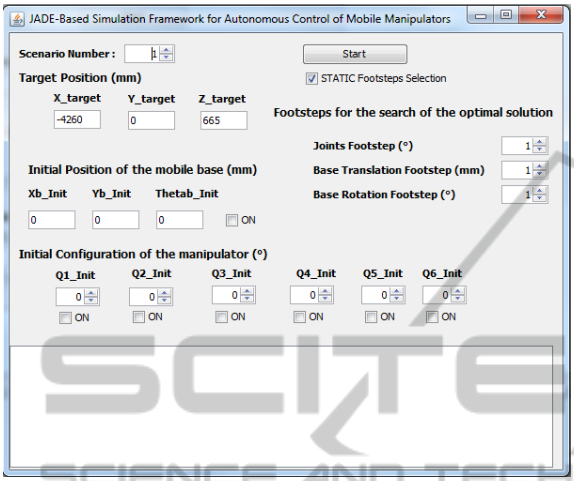

Figure 6: Screenshot of the developed simulation system.

3.2 Best Combination of Footsteps

A combination of footsteps is composed of the previous three elementary footsteps, which will affect greatly the quality of returned results. Therefore, to find the best combination, the previous tasks have been tested using various combinations.

After many tests, we have fixed the interval [1, 10] for the values of the footsteps. This will leave us with thousands of possibilities. Because just one combination is admissible, we need to make sure to select the best one. Therefore, and to overlay the maximum of values, we have selected the enumerable set {1, 5, 10} for all the footsteps. This produced 27 (3x3x3) combinations, having the following structure *{Joint, Base Translation, Base Rotation} footstep*, and varying from {1, 1, 1}, {1, 1, 5}, {1, 1, 10} to {10, 10, 10}.

A criterion needed to be defined for the evaluation of each footsteps combination. Thus, we considered two criteria (*i*) the final value of f_{Obj}, along with (*ii*) the number of iterations. These two criteria will be evaluated for each of the five previous tasks. Figure 7 presents a summary histogram for all the obtained simulation results of each task with the 27 different footstep combinations.

The prior tasks are evaluated without considering breakdowns (first case). For enhancing further the selection of the best combination of footsteps, two other cases are considered (*i*) breakdown of joints 3 and 4 of the manipulator (second case), and (*ii*) failure of the mobile base (third case).

For the following study, we have overlooked the results obtained from the last case. This is justified because all the scenarios gave almost identical solutions. Therefore, considering these results in the selection of the best footstep combination is insubstantial. The obtained results for the second case are given in Figure 8.

Depending on f_{Obj}, we have ranked the five best results for the first two cases (without breakdown and with joints breakdown) for all the tasks. The collected data allowed us to draw different tables to determine the best footsteps combination. The best combination is the one generating the maximum of best solutions for all the considered tasks.

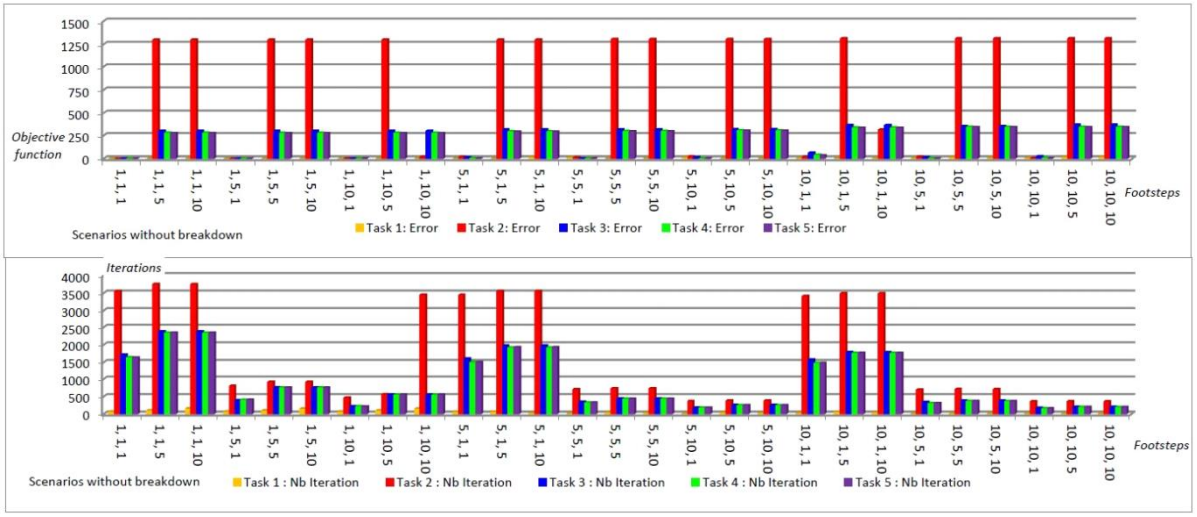

Figure 7: Obtained results without breakdown.

3.2.1 *Joint* Footstep

According to Table 3, the best *Joint footstep*=1° ({1, x, x}).

Table 3: Selection of the best *Joint footstep.*

Best solution: Without breakdown + Breakdown axes 3 and 4								
Footsteps		Task 1	Task 2	Task 3	Task 4	Task 5	Sub-total	Total
1, x, x	1, 1, x	6	2	2	2	1	13	34
	1, 5, x	2	2	2	2	3	11	
	1, 10, x	2	2	2	2	2	10	
5, x, x	5, 1, x	2	1	2	0	1	6	17
	5, 5, x	1	2	2	1	2	8	
	5, 10, x	1	0	1	1	0	3	
10, x, x	10, 1, x	0	0	1	1	0	2	04
	10, 5, x	0	0	0	1	0	1	
	10, 10, x	0	1	0	0	0	1	

3.2.2 *Base Translation* Footstep

Table 4 gives the best footstep for the translation movement of the mobile base (*Base Translation footstep*). In this case, we selected two different footsteps. The first *Base Translation footstep=1mm* ({x, 1, x}) and the second *Base Translation footstep=5mm* ({x, 5, x}).

Table 4: Selection of the best *Base Translation footstep.*

Best solution: Without breakdown + Breakdown axes 3 and 4								
Footsteps		Task 1	Task 2	Task 3	Task 4	Task 5	Sub-total	Total
x, 1, x	1, 1, x	6	2	2	2	2	14	21
	5, 1, x	2	1	2	0	1	5	
	10, 1, x	0	0	1	1	0	2	
x, 5, x	1, 5, x	2	2	2	2	2	10	20
	5, 5, x	1	2	2	1	2	8	
	10, 5, x	0	0	0	1	1	2	
x, 10, x	1, 10, x	2	2	2	2	2	10	14
	5, 10, x	1	0	1	1	0	3	
	10, 10, x	0	1	0	0	0	1	

3.2.3 *Base Rotation* Footstep

Table 5 summarizes the best footsteps for the rotation movement of the mobile base (*Base Rotation footstep*). The trivial choice is *Base Rotation footstep=1°* ({x, x, 1}).

Table 5: Selection of the best *Base Rotation footstep.*

Best solution: Without breakdown + Breakdown axes 3 and 4								
Footsteps		Task 1	Task 2	Task 3	Task 4	Task 5	Sub-total	Total
x, x, 1	x, 1, 1	3	3	5	3	3	17	47
	x, 5, 1	1	4	4	4	5	18	
	x, 10, 1	1	3	3	3	2	12	
x, x, 5	x, 1, 5	3	0	0	0	0	3	5
	x, 5, 5	1	0	0	0	0	1	
	x, 10, 5	1	0	0	0	0	1	
x, x, 10	x, 1, 10	2	0	0	0	0	2	4
	x, 5, 10	1	0	0	0	0	1	
	x, 10, 10	1	0	0	0	0	1	

3.2.4 Recapitulation

Four possibilities have been selected by using the first evaluation criterion (f_{Obj}) {1, x, x}, {x, 1, x} and {x, 5, x}, and {x, x, 1}. The crossing between these latters generates two different combinations:

- {1, 1, 1}.
- {1, 5, 1}.

For the selection of one final combination, it is necessary to adopt another criterion, which consists of comparing their respective iterations numbers. From Table 6, it can be noted that the search of the best solutions by using the first combination, {1, 1, 1}, requires a very high number of iterations and, consequently, an important execution time compared with the other combination {1, 5, 1}. This latter is selected as the best combination of footsteps for

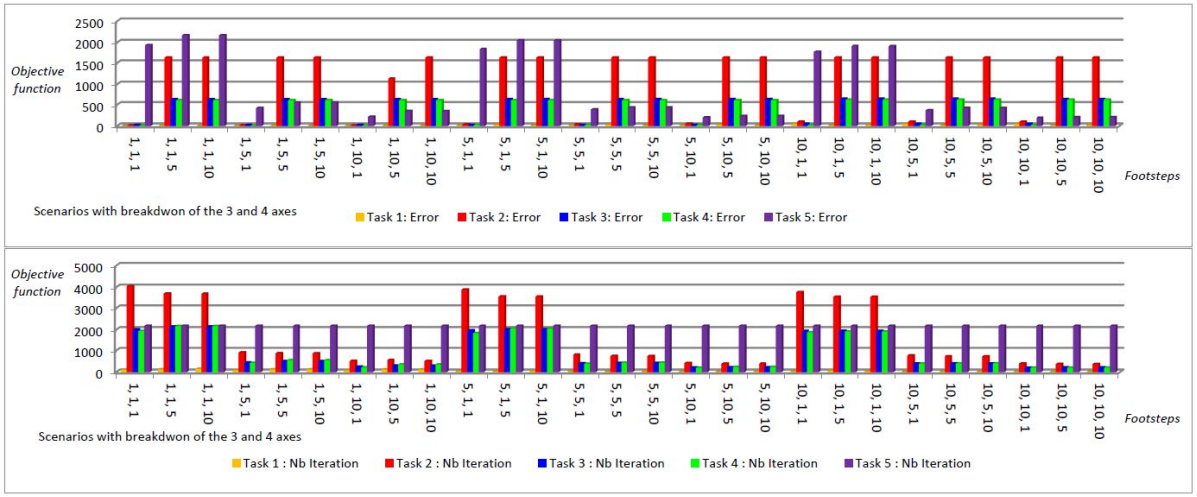

Figure 8: Obtained results with breakdowns of joints 3 and 4 of the manipulator.

searching the best solution.

Table 6: Final selection of the best footstep between (1, 1, 1) and (1, 5, 1).

Iterations: Without breakdown + Breakdown axes 3 and 4						
Footsteps		Task 1	Task 2	Task 3	Task 4	Task 5
1, 1, 1	Without breakdown	75	3555	1719	1647	1638
	With breakdown	110	4039	2004	1932	1933
1, 5, 1	Without breakdown	75	810	395	410	411
	With breakdown	85	905	444	422	420

4 CONCLUSION

Throughout this paper, we deal with the development of a novel generic approach to control autonomous mobile manipulators. The proposed approach is centered upon an agent-based framework. For the implementation, we have used the *JADE* platform which is one of the most interesting multi-agent development frameworks. A graphical interface was developed in order to perform the various validation scenarios. Finally, simulation results have been presented and discussed.

The control approach assigns a hybrid agent to control the mobile base (*Mobile base agent*), a reactive agent to control each articulation of the manipulator (*Joint agent*), and a hybrid *Supervisory agent* to coordinate and to synchronize between the previous agents. Each *Control agent* has its own local goal to be reached independently from the other agents. It consists of bringing the end-effector as close as possible from the imposed *Target* position. In its current version, the proposed approach considers a default static selection of footsteps for each *Control agent*. Consequently, finding the most fitted combination of footsteps was an important quest. Details of the methodology and the measures we have followed to search the best combination, were presented in this paper.

As an improvement of the proposed approach, we intend to implement a heuristic-based technique for dynamically tuning the footsteps. A fuzzy-logic inference system seems to be a realistic choice. Future perspective consists, also, of implementing and testing the proposed approach on the physical experimental mobile manipulator robotic platform (*RobuTER/ULM*) while accomplishing real positioning tasks.

REFERENCES

Bellifemine F. L., Caire G., Poggi A., Rimassa G., (2008), "JADE: A software framework for developing multi-agent applications. Lessons learned", *Information and Software Technology*, 50, pp. 10-21.

Davis R., Smith R. G., (1983), "Negotiation as a metaphor for distributed problem solving". *Artificial Intelligence, 20(1)*, pp.63-109.

Delarue S., Hoppenot Ph., Colle E., (2007), "A Multi Agent Controller for a Mobile Arm Manipulator". *The International Conference on Informatics in Control, Automation and Robotics (ICINCO2007)*, France.

Duhaut D., (1999), "Distributed Algorithm for High Control Robotics Structures". *International Conference on Artificial Intelligence, Vol. 1*, pp 45-50.

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.

Floroian D., Moldoveanu F. (2010), "Using Robosmith for Multi-agent Robotic System". *Bulletin of the Transilvania University of Brasov, Series I: Engineering Sciences, 3(52)*.

Hentout A., Messous M. A., Oukid S., Bouzouia B., (2013), "Multi-agent Fuzzy-based Control Architecture for Autonomous Mobile Manipulators: Traditional Approaches and Multi-agent Fuzzy-based Approaches". *The 6th International Conference on Intelligent Robotics and Applications (ICIRA2013), pp. 679-692*, Busan, Korea, 25-28 September.

Hentout A., Messous M. A., Bouzouia B., (2014), "Multi-agent Control Approach for Autonomous Mobile Manipulators: Simulation Results on *RobuTER/ULM*". *The 19th World Congress of the International Federation of Automatic Control (IFAC WC2014)*, Cape Town, South Africa, 24-29 August.

Iñigo-Blasco P., Diaz-del-Rio F., Romero-Ternero C., Cagigas-Muñiz D., Vicente-Diaz S., (2012), "Robotics software frameworks for multi-agent robotic systems development". *Robotics and Autonomous Systems, 60(1)*, pp. 803-821.

Medeiros A. A. D., (1998), "A Survey of Control Architectures for Autonomous Mobile Robots". *Journal of the Brazilian Computer Society, 4(3)*, Campinas.

Nebot P., Saintluc G., Berton B., Cervera E., (2004), "Agent-based Software Integration for a Mobile Manipulator". *The International Conference on Systems, Man and Cybernetics (SMC'04)*, pp. 6167-6172, The Hague, The Netherlands, 10-13 October.