A MULTI AGENT CONTROLLER FOR A MOBILE ARM MANIPULATOR

Sébatien Delarue, Philippe Hoppenot and Etienne Colle
IBISC, CNRS FRE 2873 - University of Evry, 40 rue du Pelvoux 91020 Evry cedex, France

Keywords: Multi-agent, arm manipulator, mobile platform.

Abstract: Assistive robotics especially mobile arm manipulator can be useful for restoring manipulation function of disabled people in everyday life tasks. However, those systems must be designed to be reliable, fault tolerant and easy to control. This article proposes a method based on multiagent system for controlling the robot. This kind of distributed architecture makes possible to be fault-tolerant without specifying additional management of faults what improves reliability. Moreover, it is also possible to add specific constraints, for example human like behaviors in order to facilitate the use of the system by the person. The multiagent method is easier to implement than classical robotics approaches.

1 INTRODUCTION

ARPH project (French acronym for Robotic Assistance to disabled People), developed in IBISC laboratory, deals with restoring handling function for handicapped person. It is a semi autonomous mobile manipulator arm. Three types of control mode are developed. In automatic one, the operator only gives the goal and the system achieves it automatically. However user would like to do by himself so in manual mode, the operator controls all the degrees of freedom of the system but the user's workload often is too important. The idea consists in developing shared modes in which operator and system share robot control. Project is divided into two research axes: robotics for building autonomous functions and human machine co-operation. The paper focuses on robot control while keeping in mind specificities due to the co-operation between human and machine. For example industrial robot performances such as accuracy are less necessary than easiness of control by the user and fault tolerance, the person being strongly tributary of the assistance.

The classical method to control a robot is to compute mathematical static and/or dynamic models of the robot (Yoshikawa, 1990). The approach provides good results in known environments for carrying out repetitive tasks. If the objective is known in Cartesian space ($p(x,y,z)^T$), those models provide speed or angular value of arm joints so that end effector performs the task. Generally models are computed off-line so they are not modifiable and cannot be adapted easily to quick changes of robot structure e.g. due to the dysfunction of one of robot joints. It requires addition of specific management for making the system fault-tolerant and taking into account on-line changes. In assistive robotics, it is an important criterion which can be solved by exploiting the system redundancy.

In literature, many works exploit redundancy for other objectives, for example to keep an optimal manipulability of the end effector (Yoshikawa, 1990), (Bayle, 2001). Some authors, (Chabane, 2005) (Yoshikawa, 1984), have proposed manipulability measures related to the task to be achieved. Our goal is to propose an only model which exploits robot redundancy and able to perform task even in case of joint default.

In addition, the model must take into account some aspects of human machine cooperation. For instance previous works demonstrated the interest of giving robot human like behaviors for facilitating the appropriation of the robot by the user (Rybarczyk, 2002) (Fuchs, 2001).

Distributed artificial intelligence makes complex problem resolution possible more easily than classical approaches. Our approach is based on a multi-agent architecture divided into two parts, a set of agents for the arm manipulator and only one for the mobile platform. After a bibliographical study, we present the multi-agent architecture able to

Delarue S., Hoppenot P. and Colle E. (2007).
A MULTI AGENT CONTROLLER FOR A MOBILE ARM MANIPULATOR.
In *Proceedings of the Fourth International Conference on Informatics in Control, Automation and Robotics*, pages 292-299
DOI: 10.5220/0001625902920299

control the arm manipulator. Then, the architecture is extended in order to control the mobile arm.

2 AGENTS AND MAS

Computer science evolves from a centralized architecture (sequential treatments) to a distributed one (parallel treatments). So, very quickly are appeared autonomous agents, able to achieve individual task with no external help. MAS for Multi Agents Systems try to solve complex problem which cannot be solved by a unique limited-mind entity (Ferber, 1995). An agent can be defined as a autonomous and flexible software entity (Weiss, 1999). An agent must be able to answer environment changes. An agent has to perceive its environment, to treat the data and then to act. This is a sensori-motor loop. Stimuli may come from the inside (agent itself) or the outside (environment). Action is performed on the agent (internal states) or on the environment. Agent behavior is the result of interaction between the agent and its environment.

It exists three ways to implement agent behaviors: cognitive, reactive, hybrid. Cognitive way divides the internal treatment into three parts perception, planning and action. Agent must have its own world knowledge. It is able to analyze the situation, to anticipate and then to plan an action. Issues of this approach are limited speed and limited flexibility. It is also inadequate in the case of unexpected events. Reactive agent locally perceives its environment (and possibly its internal states) and deduces immediately actions to be carried out only from this source of information. This principle is based on reflex action ((Zapata, 1992), (Wooldridge, 1999)). Hybrid approach merges the first two agents a basic reactive behavior with a high cognitive level. It aims at joining reactivity with thinking and organization abilities of cognitive agents ((Brooks, 1986), (Chaib-Draa, 2001)).

The objective of multi-agents systems (MAS) is to bring together a set of agents and to organize them to reach a high level goal. We can find several kinds of MAS. There are reactive systems which bring together agents and then try to get an emergent behaviour that can solve a higher level problem than each agent can do. Another type of MAS appeared with the need of making agents communicate in order to cooperate (Beer, 1998). Parker (Parker, 1999) uses a central machine which supervises messages. In this case, supervisor organizes groups of agents whose competences are different but necessary to succeed.

Reactive MAS are often used to solve problems with the help of limited-mind agent having poor world knowledge and poor action ability. The objective is to find a social emergent behaviour of an agent society able to solve a complex problem. For example, design of ants or bird behaviour uses this type of approach (Drogoul, 1993). Higher level MAS are frequently used in mobile robotics, especially in collective robotics (Lucidarme, 2003). Systems are based on criteria like gratification, altruism or cooperation (Lucidarme, 2002). A complex dialog is implemented between agents. A high hierarchical level entity is needed to oversee the task to achieve and to centralize decisions and communications. The objective is for example to coordinate several tens of mobile robots transporting containers on quays (Alami, 1998). Some others applications are developed for path planning, including obstacle avoidance (potential fields) associated with artificial life algorithms (Tournassoud, 1992), (Mitul). In robotics an application of MAS is the design of arm manipulator model. We can find few approaches (Duhaut, 1999) (Duhaut, 1993), which describe how to reach a Cartesian position without needing arm inverse kinematic model. The method seems to be well-adapted to our case. Each link is implemented like an agent. Task resolution starts with the link of end effector. It tries to align itself with the goal and to place end effector on the goal by uncoupling itself from previous link. Then, next link does the same with a new sub goal given previously.

Figure 3 shows the beginning of the recursive algorithm applied to a 3 DOF arm. Only three steps are shown in order to illustrate how it works. The goal is the cross. Initial situation is first step. On step 2, the end effector link rotates virtually and uncouples itself from the arm to reach the goal. On step 3, second link rotates and uncouples itself from the arm to join the end effector link. The main characteristic of the algorithm is that end effector link makes bigger rotation than second one and so on. Arm unfolding is not equally distributed between each joint.

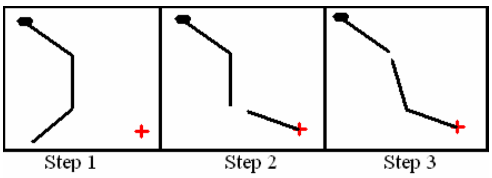

Figure 1: D. Duhaut approach.

Another use of MAS concerns high level complex system like fault detection (Guessoum, 1997) or data merging (vision, touch, sound ...). It often uses neuronal networks or fuzzy rules to find the more pertinent information. In this case the word "agent" is usually employed in the meaning of "expert system".

3 ALTERNATIVE MAS APPROACH

3.1 Presentation of the Approach

In our approach, we associate an agent to each arm joint. Each agent has the same behavior rules as the others. A joint agent computes the position of the end effector (Figure 2). The objective is to move the end effector as close as possible to the goal. The agent behavior is described by a very simple algorithm:

Agent rotates joint in one direction and computes the new position of the end effector. If this one is closer to the goal than the present one, it is the good direction and rotation continues in that direction. If the new position is not better then rotation follows the opposite direction.

By repeating this algorithm, the end effector moves as close as possible to the goal. In the basic algorithm it is possible that the agent rotates in the bad direction at the first step. We will see in the next paragraph that it can be avoided by using arm direct kinematic model.

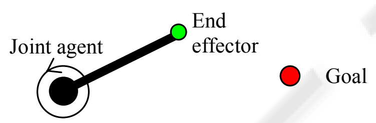

Figure 2: Initial Agent position.

This behavior is now extended to a n-joint arm. Each joint is controlled by an instantiation of the reactive agent described before. The set of arm reactive agents process data in parallel way. Agents are autonomous and each one only tries to minimize the distance between end effector and goal. The algorithm of each joint is implemented in separated processes. As said before we look for an emergent behaviour that satisfies the global mission: the arm end effector must reach the position of the goal.

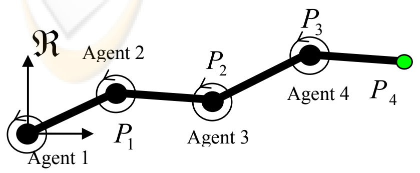

Figure 3: Four agent extent.

Figure 3 shows a four agent example. As previously explained in the one agent case, each agent must know the position of the end effector and the movement generated by its action. It can be done following two ways: first one, all joint values are recorded in real time on a blackboard readable by all agents. So, each joint agent can calculate the position of the end effector by using the simple direct kinematic model of the manipulator. This solution implies to know this model. Second way avoids the knowledge of any model by using an external system (video camera, GPS) that can detect the real position of the end effector. In the first case, as the manipulator model is known, each agent can pre-calculate the end effector position without really moving the joint it controls. So, after a simulation in one direction, it can decide the right move and rotates joint directly in the right direction. In the second case, we don't need models or any joint encoders. We don't need any communication between agents about joint position but we must accept that a joint may rotate in the wrong direction at some steps of the algorithm. What is interesting in this case is that the algorithm can be implemented directly on any manipulator without knowing any arm measurement.

Example of Figure 3 shows the case of a 2D robot. The method is also efficient for a 3D system.

3.2 Results

3.2.1 Comparison with (Duhaut, 1993)

Figure 4 shows the difference between two approaches. The first, on the left, uses our MAS algorithm called MSMA. The second, on the right and called MD, has been developed by Dominique Duhaut (Duhaut, 1993). Simulation presents a five-joint arm manipulator represented on a 2D plane. The arm unfolding is presented in three steps.

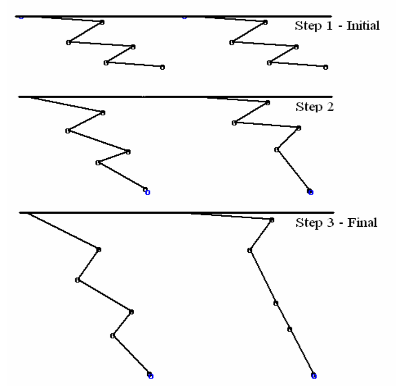

Figure 4: Comparison MSMA and MD.

We can see that during unfolding, the MD approach tries to unfold links starting with the end effector one. The MSMA system unfolds all links at the same time. So, each joint contributes to the movement, avoiding links alignment. Moreover, visual aspect is closer to human behavior. When a person wants to take an object, he does not stretch himself to the maximum.

Figure 5 shows an arm folding. Again, we can see a more distributed movement between joints in the case of the MSMA approach. There is no collision between links.

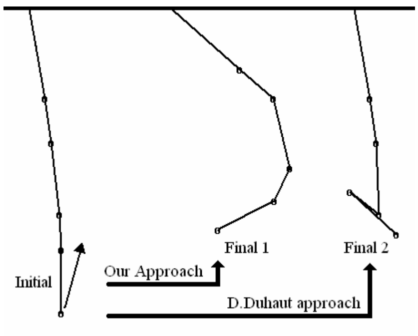

Figure 5: Arm folds.

3.2.2 Behavior in Case of Broken Joint

Duhaut's method (Duhaut, 1993) does not permit to simply take into account a joint fault so we only present results with our method. Figure 6 shows system behavior including two broken joints (dashed limb and squared joint). In this simulation we can use only the end effector position knowledge (no kinematic model, no encoder values). In that case, broken joint means the motor associated to it is out of order (joint encoder out of order or not). If we use a blackboard and the direct kinematic model, as explained in the presentation of the approach, then, the joint encoder must not be out of order to guarantee a correct operation.

Here, the arm works at only 60% of its capacities. Reachable domain is delimited by two half circles. Dotted area represents the reachable space taking into account the reduced capacities. A systematic test has been realized, covering all the reachable space: 100% of the 4592 tested positions have been reached. The figure also shows three sample positions. It is possible because of the redundancy of the system.

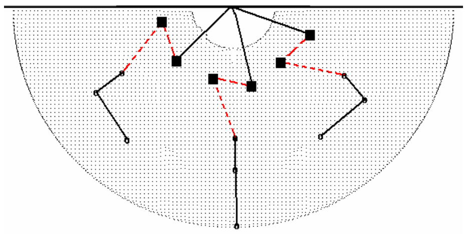

Figure 6: Two broken joints simulation with MSMA.

3.2.3 Discussion

As agents work independently, all joints participate to the movement inducing a natural visual aspect of the arm for the user. For us, this property is very important because it gives the system a human behavior and helps human-machine cooperation. The movement is distributed in all joints and the arm configuration stays far from singular positions. We can also see that there is no link collisions while the arm folds. In the third simulation, without integrating any fault treatment, we can see that MSMA using external localization system is fully fault tolerant if goal is reachable. If a blackboard is used to store joint values in addition to the direct kinematic model, joint encoders must not be out of order even if motor is. In the two cases, even if a motor is randomly moving because of noises or control system fault, the system will reach the goal. We can notice that MAS system does not need to be informed of joints faults. Each software agent does its work without taking care if it really controls or not its associated joint.

4 APPLICATION TO MOBILE ARM MANIPULATOR

The system to control is composed of a manipulator arm embarked on a mobile platform. The first objective deals with human-machine co-operation. The idea is to give to the system behaviors inspired from human being. For example, when a person wants to take a book, he/she tries to keep his/her arm not extended. If the book is too far to be taken by arm extension, the person walks in the direction of the book aligning the body on the hand movement direction. The second objective is to make the system more tolerant to some joint fault by exploiting redundancy without using specifics fault treatments. Indeed, in assistive field, it is important to maintain a good quality of service.

4.1 Mobile Platform Agent

The control of the mobile platform uses an agent which controls angular and linear speeds. The mobile platform agent is more complex than the arm ones. The agent used for the mobile platform is an hybrid one. Its cognitive capacities give the possibility to add some interesting behaviors.

Firstly, the mobile platform has to move forward because sensors for obstacle avoidance are located on the front side of the system.

The second implemented behavior is to align direction of displacement of the mobile platform with the manipulator arm. Indeed, when a person wishes to catch an object and must move to do it, he/she generally tights the arm forward in the direction of the movement of the body.

The third behavior concerns deadlock. In certain cases, both mobile platform and arm shoulder joint rotate at the same speed, but one on the right and the other on the left direction. In that case, the gripper does not converge to the objective. The agent of the platform is able to detect this kind of situation. It introduces a waiting cycle by leaving to the arm the priority for reaching the goal. Once this cycle ends, the platform agent tries again to align itself with the arm if it is possible.

Reactive behavior is the same one as for the arm agents. Mobile platform agent tries to minimize distance between the end effector and the goal. Mobile platform agent works independently from the arm ones.

4.2 Results

We now simulate the whole system algorithm. First we compare it with a classical mathematical method using manipulability constraints. Secondly, we simulate faults on some joints.

4.2.1 System and Protocol

The system is a 3D mobile arm. It is composed of a mobile platform equipped with two driving wheels and a free wheel to ensure stability. A 6 DOF manipulator arm is fixed on the mobile platform. In the following simulations, the end effector imposed task consists in following a straight line with a constant speed, in the upper-right direction of the mobile platform. Figure 7 describes the system and the associated mathematical frame. The displacement is perpendicular to the initial orientation of the platform. The displacement plan is formed by x and y axis. Initial conditions are showed in Table 1.

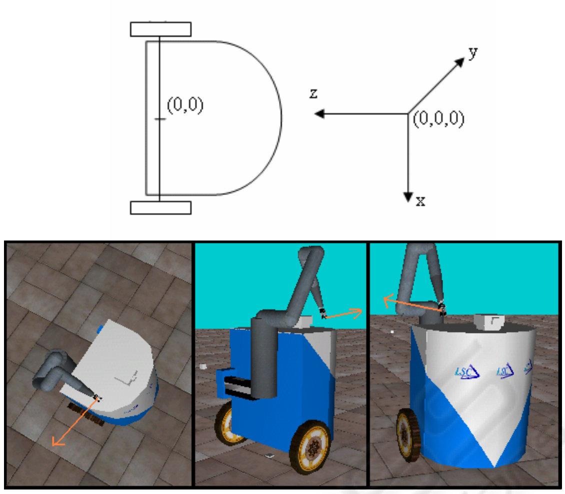

Figure 7: System description.

Table 1: Initial simulation conditions.

Object	**Initial value**
Platform position	(0,0,15) meters
Manus Joint 1	270 degrees
Manus Joint 2	120 degrees
Manus Joint 3	-125 degrees
End effector position	(18,79,-20) meters
Simulation steps	400
Sampling rate	60 ms
Shift wanted for each step	(0.42 , 0.12 , 0) cm
Total duration	24 s

On the first simulation, objectives are the following ones:

- End effector must follow the desired trajectory with good accuracy
- Mobile platform must move forward because of front side sensors belt
- Mobile platform has to align itself with the arm orientation
- Arm has to avoid its extended configurations.

We compare our approach (MSMA) using agents introduced above with the MIM one which uses manipulability criterion (Chabane, 2005).

Secondly, we check fault tolerance ability of MSMA approach. So, we simulate a breakdown of the shoulder and check if the whole system redundancy (generated by the mobile platform) permits to the system to reach the goal. We also simulate a breakdown of joint 2 to check the whole system behavior.

4.2.2 Results

Her, we do not show end effector trajectory. With both approaches, the move is correct with a good accuracy (less than 3mm of difference between the real path and the desired one). There is no notable difference between them. Accuracy is linked to simulation step duration. We choose a high one of 60 ms because our real system has a 60 ms command loop.

Figure 8 shows platform trajectory and orientation. We can see two very different performances. MSMA works well. It always goes forward keeping obstacle avoidance sensors in front. It first turns slowly on the right, and then goes straight until the end and aligns itself with the arm orientation. We see a graining point in MIM simulation and the platform ends the task moving backwards.

During the move and with both models, the arm is never bended. Angle between joint 2 and joint 3 stabilizes to a 70 degrees value which is far from the 0 degrees singular value of the arm.

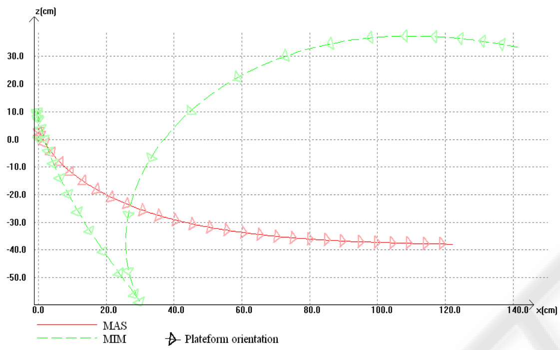

Figure 8: MIM and MSMA platform moves.

We now show how the system reacts in fault cases. MIM is not designed to be fault tolerant so its behavior is not interesting here. It is why next simulations show only MSMA model in three samples faults situations:

- Arm breakdown joint 1 (shoulder) at 60°
- Arm breakdown joint 1 (shoulder) at 30°
- Arm breakdown joint 2 at 120°

These values are chosen as examples to illustrate motor joint breakdown fault tolerance. We assume that if goal is reachable, fault tolerance is effective in all the cases. Breakdown occurs at time 0. Figure 9 shows end effector trajectory on x axis. We can see that MSMA satisfy objective even in fault situations. Results are the same on y and z axis. In the first two situations, redundancy between arm joint 1 and the mobile platform rotating movement is exploited. As end effector follows the wanted path, we can say that the platform agent works well when the arm shoulder is broken. In that case, mobile platform moves forward but does not align with the arm orientation. Indeed, this alignment is performed by the redundancy between arm joint 1 and the mobile platform rotating movement and, in that case, joint 1 is broken. Moreover, the arm is not tightened and the angle between joint 2 and 3 is stabilized to 70 degrees. In the third situation, redundancy between joint 1 and joint 3 is exploited. Both of them permit a vertical move (y axis). Once again the end effector follows the trajectory correctly. In that case, the mobile platform moves forward and aligns with the arm orientation. We can still notice that the arm is not tightened and that its posture remains far from singular position.

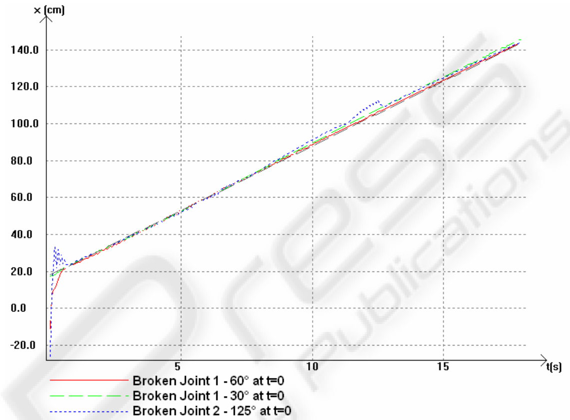

Figure 9: End effector trajectory on axis x.

4.2.3 Discussion

On the first simulation, we can see that the classical model used implies a more complex trajectory for the mobile platform on the whole system simulation. There is a graining point making the robot blind (it can not avoid obstacles anymore because ultrasonic belt is in front side). It’s due to the manipulability criterion used in this simulation. Our model shows a better behavior, closer to a human one. The platform goes forward and aligns itself on the arm orientation inducing a more assimilable move for the user. The use of independent agent helps to considerate directly human behavior in algorithm. Also, the system is fault tolerant as shown in simulation. The end effector desired trajectory is reached even if arm joint 1 or 2 are broken. This ability is due to multi agent architecture which is able to run even if a component is defective.

5 DISCUSSION

First, with our approach, we do not cut out the final objective in sub goals which each agent would have to reach. Each one has a local work to do independently from others. We do not organize any co-operation. Here, we speak about emergent behavior. Indeed, one agent can not reach the goal alone. It needs others to achieve the task but it does not know it. This kind of system provides very good result concerning fault-tolerance.

Secondly, this approach induces a goal for each agent. It is then possible to influence behavior of some agents. Then we can easily adjust or add behaviors to facilitate man machine co-operation. That is the case for example with platform alignment on the direction of the end effector in the same way than alignment of the human body on its hand direction. That leads to an easier assimilation of the system by the user. Indeed, with a classical model, to add a behavior requires the integration of constraints in the global model itself, which is not easy with this kind of system.

Our approach has also its limits. If we integrate many behaviors, it is possible for the system to loose its wanted emergent behavior. The added constraints could be in conflict with the initial objective which is to reach the goal. The system then enters in non convergence cases. At the same time, we can loose fault tolerance ability. To avoid this kind of errors, it is first necessary for agents to keep autonomy compared to the goal they have to achieve and thus to be as simple as possible. Secondly, we have to include priorities in the local objectives of each agent. Reaching the goal has a high priority, going forward has a smaller one. Aligning platform on arm orientation has a very small one. Thirdly, we have to implement deadlocking treatment by introducing delays in specific situations. In our system, these particular treatments are implemented in the agent of the mobile platform. We do not plan any problem resolution between agents. In our approach, we keep simple algorithm to avoid high hierarchical management. Indeed, high hierarchical management could then be compared with a system using mathematical models and including a fault treatment supervisor to switch between them.

6 CONCLUSION AND PERSPECTIVES

Our MAS system gives good results in relation to human behavior. Objects can be caught with an easily assimilable movement for the user (forward move, alignment of the platform with arm orientation, simple trajectory with no turnaround, no bended arm). Accuracy is similar to classical method. It is fault tolerant without integrating any specific treatment. It makes easier the integration of special human driving mode. There is no need to compute mathematical model and especially the inverse model. Our MAS system algorithm is easy to implant and need only some geometric formulas and thus very little computing power. It is a real time one. We now have to implement algorithm on our mobile platform and create scenarios of displacement to judge the relevance of our algorithm on a real robot. The first tests on the real system give encouraging results. Simulations shown in this article give an idea of the mobile manipulator behavior with straight lines trajectory. These kinds of trajectories are usually used by people driving ARPH with HMI actually developed. So we think that MAS system can directly replace the Cartesian internal manipulator mode actually used. Actual HMI allows user to control the platform and the arm manipulator separately (two sets of buttons) as shown on Figure 8.

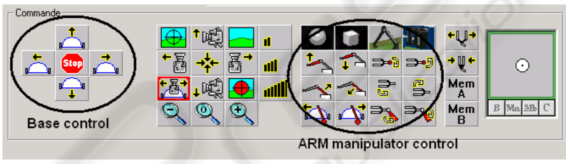

Figure 10: The actual HMI two sets of buttons.

With MAS, it's possible to control the whole system with only one set of buttons by driving only the grip, thus limiting the difficulty of seizure operation by the user.

We also want to improve object grasping. One possible way is to integrate a neuronal network that could help the system to have a better posture (eg: catching an object by the left if user is a left hand writer...). This improvement should lead us to manage with agents not only for the mobile arm but also for the orientation of the grip. Moreover, ARPH mobile platform has an ultrasonic sensors belt for obstacle avoidance when moving in a room that has to be integrated with MAS system to have a fully operational system.

REFERENCES

Yoshikawa, T.: Foundation of robotics: *"Analysis and control"*, The MIT Press, 1990

Bayle, B., Fourquet, J.Y. and Renaud, M.: *"Manipulability Analysis for Mobile Manipulators"*, ICRA'2001, Séoul, South Korea, May 2001, pp. 1251-1256.

Khiar Nait Chabane; Philippe Hoppenot; Etienne Colle, *"Mobile Arm for Disabled People Assistance Manipulability Analysis"*, International Computer Systems and Information Technology Conference, Algeria, July 2005

Yoshikawa, T.: *"Analysis and control of Robot manipulators with redundancy"*, In M. Brady & R.

Paul. Robotics Research: The First International Symposium, MIT Press, 1984, pp 735-747.

Yves Rybarczyk, Etienne Colle, Philippe Hoppenot: *"Contribution of neuroscience to the teleoperation of rehabilitation robot"*, SMC'2002, Hammanet, Tunisia, 6-9 october

Fuchs, J. Moreau, G et Papin, *“Le traité de la réalité virtuelle“*, Nantes : Les Presses de l’Ecole des Mines, 2001

Jacques Ferber, *"Les systèmes Multi agents Vers une intelligence collective"*, IIA Informatique Intelligence Artificielle InterEditions, september 1995

Weiss G., *"A Modern Approach to Distributed Artificial Intelligence"*, The MIT Press, Cambridge, Massachusetts , 1999

R. Zapata, P. Lepinay, C. Novalés and P. Déplanques, *"Reactive behaviors of fast mobile robots in unstructured environments : sensored based control and neural networks."* SAB, December 1992.

Wooldridge M., Weiss G., "Intelligent Agents" *Multiagent systems*, - Ed.: MIT Press - 1999.

R. A. Brooks, *"A robust layered control system for a mobile robot"*, IEEE Journal of Robotics and Automation, p. 14-23, 1986.

Chaib-Draa B., Jarras I., Moulin B., *"Systèmes multi-agents : principes généraux et applications ,Principes et architectures des systèmes multi-agents*, J.-P. Briot et Y.Demazeau (Eds.) Hermès - 2001.

Beer M., d’Inverno M., Luck M., Jennings N., Preist C., Schroeder M., *"Negotiation in Multi-Agent Systems"*, Workshop of the UK Special Interest Group on Multi-Agent Systems (UKMAS’98) - 1998

L. Parker, *"Adaptative heterogeneous multi-robot teams"*, Neurocomputing, special issue of NEURAP’98: Neural networks and their applications, 28, p.75-92, 1999.

Drogoul A, *"De la simulation multi-agent à la résolution collective de problèmes. Une étude de l’émergence de structures d’organisation dans les systèmes multi-agents."*, Thèse de doctorat, Université Paris 6. - 1993

P. Lucidarme and A. Liégeois, *"Apprentissage de comportements réactifs pour des ensembles de robots mobiles"*,JJCR17, Paris, pp. 33-39, IEEE best paper award - 2003

P. Lucidarme, O. Simonin et A. Liégeois, *"Implementation and evaluation of a satisfaction/altruisme-based architecture for multi-robot systems"*, proc. ICRA'02, Washington D.C., pp. 1007-1012 - 2002

R. Alami, S. Fleury, M. Herrb, F. Ingrand and F. Robert *"Multi-robot cooperation in the Martha project"*, IEEE Robotics and AutomationMagazine, 5(1), p.36-47, 1998.

Pierre Tournassoud, *"Planification et contrôle en robotique Application aux robots mobiles et manipulateurs"*, Edition Hermes Traité des nouvelles Technologies série robotique – 1992

MITUL SAHA, *"Manipulator Planning and Obstacle Avoidance Using Genetic Algorithm"*, http://members.rediff.com/mitulsaha/ga.html

Dominique Duhaut, *"Distributed Algorithm For High Control Robotics Structures"*, International Conference On Artificial Intelligence – 1999 – Volume 1 p 45 à 50

Dominique Duhaut, *"Using multi agent approach solve inverse kinematic problem"*, proceeding 1993

Z. Guessoum, *"A Hybrid Agent Model: a Reactive and Cognitive Behavior"*, ISAD97, IEEE, Berlin, Germany, pp. 25-32 - April 1997.