An original approach for a better remote control of an assistive robot

Sébastien Delarue, Paul Nadrag, Antonio Andriatrimoson,
Etienne Colle and Philippe Hoppenot
IBISC Laboratory - University of Evry Val d'Essonne - France

Abstract

Many researches have been done in the field of assistive robotics in the last few years. The first application field was helping with the disabled people's assistance. Different works have been performed on robotic arms in three kinds of situations. In the first case, static arm, the arm was principally dedicated to office tasks like telephone, fax… Several autonomous modes exist which need to know the precise position of objects. In the second configuration, the arm is mounted on a wheelchair. It follows the person who can employ it in more use cases. But if the person must stay in her/his bed, the arm is no more useful. In a third configuration, the arm is mounted on a separate platform. This configuration allows the largest number of use cases but also poses more difficulties for piloting the robot.

The second application field of assistive robotics deals with the assistance at home of people losing their autonomy, for example a person with cognitive impairment. In this case, the assistance deals with two main points: security and cognitive stimulation. In order to ensure the safety of the person at home, different kinds of sensors can be used to detect alarming situations (falls, low cardiac pulse rate…). For assisting a distant operator in alarm detection, the idea is to give him the possibility to have complementary information from a mobile robot about the person's activity at home and to be in contact with the person. Cognitive stimulation is one of the therapeutic means used to maintain as long as possible the maximum of the cognitive capacities of the person. In this case, the robot can be used to bring to the person cognitive stimulation exercises and stimulate the person to perform them.

To perform these tasks, it is very difficult to have a totally autonomous robot. In the case of disabled people assistance, it is even not the will of the persons who want to act by themselves. The idea is to develop a semi-autonomous robot that a remote operator can manually pilot with some driving assistances. This is a realistic and somehow desired solution. To achieve that, several scientific problems have to be studied. The first one is human-machine-cooperation. How a remote human operator can control a robot to perform a desired task? One of the key points is to permit the user to understand clearly the way the robot works. Our original approach is to analyse this understanding through appropriation concept introduced by Piaget in 1936. As the robot must have capacities of perception

decision and action, the second scientific point to address is the robot capacities of autonomy (obstacle avoidance, localisation, path planning…). These two points lead to propose different control modes of the robot by a remote operator, from a totally manual mode to a totally autonomous mode. The most interesting modes are the shared control modes in which the control of the degrees of freedom is shared between the human operator and the robot. The third point is to deal with delay. Indeed, the distance between the remote operator and the robot induces communication delays that must be taken into account in terms of feedback information to the user. We will conclude this study with several evaluations to validate our approach.

1. Introduction

Many researches have been done in the field of assistive robotics in the last few years. The first application field was helping with the disabled people's assistance. Different works have been performed on robotic arms in three kinds of situation. In the first case, static arm, the arm was principally dedicated to office tasks like telephone, fax… Several autonomous modes exist which need to know the precise position of objects. In the second configuration, the arm is mounted on a wheelchair. It follows the person who can employ it in more use cases. But if the person must stay in her/his bed, the arm is no more useful. In a third configuration, the arm is mounted on a separate platform. This configuration allows the largest number of use cases but also poses more difficulties for piloting the robot.

The second application field of assistive robotics deals with the assistance at home of people losing their autonomy, for example a person with cognitive impairment. In this case, the assistance deals with two main points: security and cognitive stimulation. In order to ensure the safety of the person at home, different kinds of sensors can be used to detect alarming situations (falls, low cardiac pulse rate…). For assisting a distant operator in alarm detection, the idea is to give him the possibility to have complementary information from a mobile robot about the person's activity at home and to be in contact with the person. Cognitive stimulation is one of the therapeutic means used to maintain as long as possible the maximum of the cognitive capacities of the person. In this case, the robot can be used to bring to the person cognitive stimulation exercises and stimulate the person to perform them.

Different works deal with autonomous robotics. They have several drawbacks in these kinds of application. Concerning disabled people assistance, persons want to act by herself/himself on the environment. In the case of people loosing their autonomy, one important point is to permit human-human interaction, through a robot seen as intermediary communication. One can also notice that autonomous robotics can not yet propose robots with several days of autonomy (except for spatial missions but at very expensive costs and with limited action capabilities). Our option is to develop remote control robots. That has two main advantages. As human being is in the control loop, it is possible to use her/his capacities, especially decision ones, which are the most difficult to get from a robot. That permits to assure total autonomy. The second point is that the remote operator is involved in the performed task, which is for example a clear demand from disabled people using technical assistance. This choice implies that a mission is realised by close Human Machine Cooperation between the robot and the human operator. It is also clear that the robot has some kinds of autonomous capacities, which can be used by the remote operator if needed.

The first part of this paper deals with autonomy capacities of the robot. It is essential to know what the robot can do to think about Human Machine Cooperation, which is the main point of the second part. We also propose different evaluations results in the last part of the paper.

2. Robot capacity of autonomy

Displacement in an environment is realised in two steps. The first one is the description of the trajectory to perform to reach the given goal. The second one consists in following the previous trajectory, avoiding unexpected obstacles. To make these two steps possible, the system needs to have information on its environment and the capacity to localise itself in this environment. We do not address the case in which the system has no information on its environment and builds itself a representation of it, using SLAM techniques. We suppose the system has a sufficient precise knowledge of the environment, which is the case in our application field. In short, displacement in an environment requires tow kinds of capacities: path planning and obstacle avoidance. A combination of these two capacities gives the robot navigation capacities. Localisation is also needed to achieve displacements toward a goal.

2.1 Trajectory planning

This is the first step of autonomy and permits to define the trajectory the robot has to follow. [Latombe91] presents the three main method families for planning: road maps, cell decomposition and potential fields. Before presenting them briefly, it is useful to define what free space is.

Free space describes all the positions the robot can reach taking into account environment information. In 2D mobile robotics, that represents all the (x, y, θ) positions where the robot can arrive. To simplify computation algorithms, a classical solution is to describe a kind of "growing" obstacle obtained by extension of the workspace by the dimensions of the robot. In that case, each orientation of the robot generates a workspace in which the robot can be considered as a point. In the case of a circular robot, only one workspace is needed. We introduced imaginary obstacles around the door to make the planned trajectory easier to follow.

The study of the connectivity of the robot's free space enable to determine a network of 1 dimension curves called a roadmap, which describe all possible trajectories from an initial point to a goal point. Among all possible paths, only one is chosen. A^* algorithm is well adapted for this work. Given a cost function, it determines the optimal path. It is possible to optimise the distance, but the function can also take into account the amount of energy, perception, capacities of the robot. In the case of cell decomposition, the workspace of the robot is split into several parts called cells. They are built to assure that all couples of points inside the same cell are linkable by a straight line. A graph linking all the adjacent cells is also built, which is called the connectivity graph. The idea of potential fields is to determine an artificial potential field representing the constraints given by the environment. Obstacles create a repulsive force while the goal creates an attractive force. This method has a well-known major drawback: the function has local minima, which are not the goal.

In our project, we use a visibility graph with the A* algorithm. In our application (the robot evolves indoors), the environment is sufficiently well-known so that the built graph is nearly complete. Moreover, cost function is interesting to use because we can choose several parameters to optimise the trajectory. All these aspects are detailed in [Hoppenot96] and [Benreguieg97].

2.2 Obstacle avoidance

A robot can follow the trajectory planned as above only if the environment is totally known. That is why local navigation systems have been developed based on robot sensors acquisition. It is now usual to combine path planning and local navigation. In that case, path planning is in charge of long terms goal while local navigation only deals with obstacle avoidance. Some qualitative reasoning theories have been developed. We propose a solution based on fuzzy logic to process obstacle avoidance ([Zadeh65], [Kanal88], [Lee90]).

When the vehicle is moving towards the target and the front sensors detect an obstacle located on the path, an avoiding strategy is necessary. The selected method consists in reaching the middle of a collision-free space. The used navigator is built with a fuzzy controller based on a set of rules as follows:

If Rn is x_i **and** Ln is y_i **Then** ω is t_i

and if Fn is z_i **then**

v is u_i "

else…

x_i, y_i, z_i, t_i and u_i are a linguistic labels of a fuzzy partition of the universes of discourse of the inputs Rn, Ln and Fn and the outputs ω and v, respectively. The inputs variables are the normalised measured distances on the right R, on the left L and in front F such as:

$$R_n = \frac{R}{R+L},\ L_n = \frac{L}{R+L} \text{ and } F_n = \frac{F}{\inf}$$

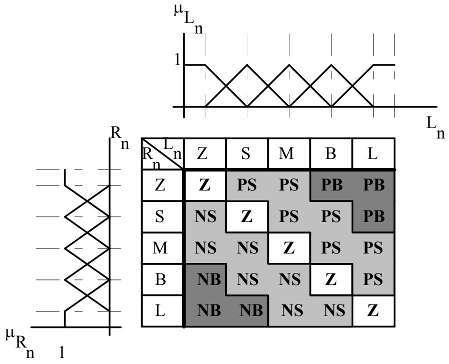
where inf is the sensor maximum range.

The output variables are the angular and the linear speeds. On simplicity grounds, the shape of the membership functions is triangular and the sum of the membership degrees for each variable is always equal to 1. The universes of discourse are normalised between -1 and 1, for ω, and 0 and 1 for the other ones.

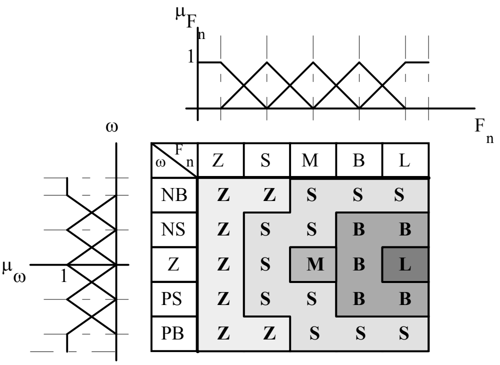
Each universe of discourse is shared in five fuzzy subsets. The linguistic labels are defined as follows:

Z	: Zero	NB	: Negative Big
S	: Small	NS	: Negative Small
M	: Medium	Z	: Zero
B	: Big	PS	: Positive Small
L	: Large	PB	: Positive Big

The whole control rules deduced from a human driver's intuitive experience is represented by fifty rules shown in the two following decision tables (Table1 and Table2): 25 rules allow to determine the angular velocity w and 25 others determine the linear speed v.



R_n \ L_n	Z	S	M	B	L
Z	Z	PS	PS	PB	PB
S	NS	Z	PS	PS	PB
M	NS	NS	Z	PS	PS
B	NB	NS	NS	Z	PS
L	NB	NB	NS	NS	Z



ω \ F_n	Z	S	M	B	L
NB	Z	Z	S	S	S
NS	Z	S	S	B	B
Z	Z	S	M	B	L
PS	Z	S	S	B	B
PB	Z	Z	S	S	S

Table 1. Angular velocity rules

Table 2. Linear speed rules

An example of such fuzzy rules is:

If (R_n is *Large*) **and** (L_n is *Large*) **then** (ω is *Zero*) **and if** (f is *Large*) **then** (v is *Large*).

Complete results are detailed in [Hoppenot96] and [Benreguieg97].

2.3 Navigation behaviour

The ω and v control actions produced by the above fuzzy controller handle the robot to avoid the obstacles when it is attracted by the immediate nearest subgoal (SG^k). This latter exercises an attractive force which guides the robot to its destination. The actions (ω_a and v_a) generated by this force are modulated by the inverse of the distance $\left(\left\|R, SG^k\right\|\right)$ between the centre of the robot and the k^{th} subgoal.

$$\begin{array}{lll}
\text{If } \left(\left(L_n \notin [0.2;0.4]\right) \text{ or } \left(R_n \notin [0.2;0.4]\right) \text{ or } \left(F_n < 0.2\right)\right) & & \\
\quad \text{then:} \quad \omega_a = 0 & \text{and} & v_a = 1 \\
\text{else if } \left(\left\|R, SG^k\right\| > D\right) & & \\
\quad \text{then:} \quad \omega_a = \dfrac{C_{a\,t}}{\left\|R, SG^k\right\|}\theta_a & \text{and} & v_a = 1 - \omega_a \\
\quad \text{else:} \quad \omega_a = 0,5 \,.\, \theta_a & \text{and} & v_a = 1 - \omega_a
\end{array}$$

The setpoints V and Ω applied to the robot result of a linear combination between the obstacles avoidance and the subgoal attraction.

$$\begin{aligned}
&\text{If } \left(\left\| R, SG^{k} \right\| < D\right) \text{ or } \left(\left\| R, SG^{k-1} \right\| < D\right) \\
&\qquad\qquad \text{then } V = Min(v, v_a).V_{\min} \ \left(m/s\right) \\
&\text{else} \qquad\quad V = Min(v, v_a).V_{\max} \ \left(m/s\right) \\
&\Omega = \beta \ \left(\omega + \alpha\omega_a\right) \qquad \left(\text{rd/s}\right)
\end{aligned}$$

where Cat, α and β are coefficients adjusted by experimentation to get the best trajectory generation.

2.4 Localisation

Planning and navigation (in the sense of following the planned trajectory) are possible only if the robot has information about its localisation. Localisation methods are divided into two families. Relative localisation consists in computing present position taking into account the previous one and the robot displacement. This method is easy to implement in real-time, but its main drawback is that its error is not bounded and tends to grow with time. Absolute localisation is based on exteroceptive perception and knowledge of the environment. It is performed in 4 steps: (i) data acquisition (here with a camera), (ii) primitive extraction (here segments), (iii) 2D-3D Matching and (iv) position computing. The major part of our work was done on the last two points.

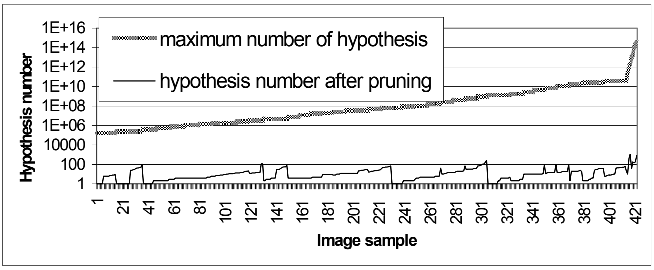


Fig. 1. Pruning performance with unary and binary constraints

Concerning matching methods, they can be classified in two groups: methods which search a solution in the “correspondence space” such as alignment ([Ayache86], [Lowe87]), geometric hashing ([Lamdan88]) or interpretation tree search ([Grimson90b], [Grimson87]) and those which search in the “transformation space” such as generalised Hough transform ([Grimson90a]). One of the most popular approaches is the interpretation tree search introduced by Grimson ([Grimson90b], [Grimson87]). We have proposed a two-stage method for mobile robot localisation based on a tree search approach and using straight line correspondences. The first stage serves to select a small set of matching hypothesis. Indeed, exploiting some particularities of the context, the sets of image lines and model segments are both divided in two subsets. Two smaller interpretation trees are then obtained. Two different geometric constraints (a unary constraint and a binary constraint) which can be applied directly on 2D-3D correspondences are derived and used to prune the interpretation trees. In the second stage, poses corresponding to retained matching hypothesis are calculated. An error function is used to select the optimal match if it does exist. Figure 1 shows the number of hypotheses to be tested regarding their total number. As the ordinate axis scale in a logarithm one, the method gives very interesting results. Considering that point (iv) is the time consumer, it is very important to reduce as much as possible the number of hypotheses to test. All results can be found in [AitAider05], [AitAider02b] and [AitAider01].

Position computing (step (iv)) is based on Lowe’s algorithm ([Lowe87]). We have proposed two main adaptations of this method. First of all, Lowe's method is based on point correspondences. In this application, the image is first segmented into contours. Contours generally correspond to physical elements in the work space, such as edges constituted by intersections between surfaces of the flat. These edges tend to be straight segments. Lines are easier to extract from contour images and their characterisation by polygonal approximation is reliable even in the presence of noise. Partial occlusion (due to the view angle or the presence of non-modelled objects) does not affect line representation parameters. Furthermore, the extremities of the edges that could possibly be considered as point features are not always seen in the image due to the dimension of the flat edges in comparison with their distance to the camera. These reasons make it more prudent to use straight line correspondences. Thus, the 3D model can simply comprise a set of straight segments whose extremities have known co-ordinates in the world frame. The second adaptation concerns the degrees of freedom of the system. Lowe's algorithm works in 6D (three positions and three orientations). In our context, we have only two positions and one orientation as the robot evolves in a 2D environment. That gives the possibility to reduce the number of parameters. The obtained system of equations is still non-linear and contains multiple unknowns. Convergence properties are highly dependent upon the quality of the initial estimate of the solution vector. Many situations unfortunately arise in which the robot is “completely lost” in its environment and has no perception of its actual location. An approach to reducing the effects of non-linearity is to find a way to uncouple some of the variables. The rotation and translation parameters have been uncoupled. An initial estimate of the solution can be found by analytically solving one of these equations. A numerical optimisation by means of least squares using Newton's method is then to be applied. Development can be found in [AitAider02a].

3. Human machine cooperation

3.1 Appropriation and human-like behaviour

According to Piaget, the intelligence is before all adaptation ([Piaget36], [Piaget52]). The functional organization of the living being emerges from the balanced relation which is established between the individual and the environment. This balance is made possible by transformations induced by the characteristics of the environment with which a person interacts. For Piaget, who analyzes the birth of the intelligence in its sensorimotor dimension, the adaptation can break up into two processes.

The first one is the process of assimilation. According to this author, this process is defined as the tendency to preserve a behavior. That is made possible thanks to a certain repetition of the behavior in question which thus is schematized. A scheme constitutes a structured set of the generalizable characteristics of the action which will allow the reproduction of the same action even if the scheme is applied to a new but close situation. These schemes constitute an active organization of the lived experiment which integrates the past. They thus include a structure which has a history and progressively changes with the variety of the situations met. The history of a scheme is that of its generalization but also of its adjustment to the situation to which it is applied. Generalization is conceptualized by the process of assimilation. Concretely, by their proximity of appearance or situation, the use of new objects can be assimilated to pre-existent schemes. The property of differentiation, as for it, refers to the second process responsible for the adaptation called by Piaget accommodation. When the complexity of the situation does not allow a direct assimilation, a mechanism of accommodation builds a new scheme by important modifications of pre-existent schemes. If one takes the example of the acquisition of the manipulation of a stick by the young child ([Piaget36], [Piaget52]), one completely understands the nature complementary of this process with that of the assimilation. In this experiment, a child is placed vis-a-vis a sofa on which a bottle is posed out of range. However within his hand range, there is a stick. Initially he tries to seize the bottle directly, then seizes the stick and strikes with the stick and by chance makes object fall. When the bottle is on the ground, he continues to strike by observing the movements obtained, then he ends up pushing the object with the stick to bring it back towards him. Later, in the absence of the stick, he seizes a book to bring closer the bottle. The child thus, first of all, implemented a scheme already made up - to strike with a stick - but this assimilation of the situation to the scheme does not make it possible to succeed each time. Consequently the scheme gradually will be adapted in order to manage the displacement of the object, until leading to a new scheme: to push with a stick. Lastly, this one will be generalized to other objects, here a book. The same mechanism is applied to the man-machine relation. The development of sensorimotor schemes in the young child is relatively transposable with the situation of the operator having to build schemes of action for controlling the robot. When the machine presents operating modes close to those known by the operator, he builds robot control schemes by an assimilation process based on preexistent schemes. On the contrary, if the machine operating is completely different, the person is obliged to accommodate. It is this principle of adaptation, at Piaget sense, applied to the man-machine relation which we describe as mechanism of appropriation.

3.2 Ergonomic design

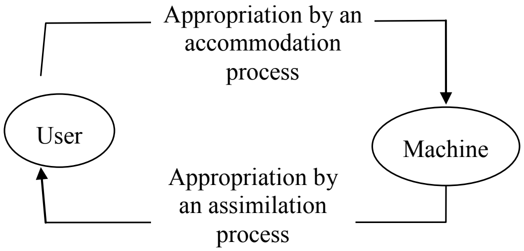

Appropriation by an accommodation process
User
Machine
Appropriation by an assimilation process

Fig. 2. Application of the Piaget model of adaptation to the Man-machine Co-operation

The preceding considerations expose that the nature of the adaptation of the human to the machine is strongly dependent on the operating difference between these two entities (Figure 2.). If the difference is weak, the adaptation is carried out by a process with a dominant of assimilation, i.e. by the generalization of the initial schemes relevant for the control of the machine. Conversely, if there is a high difference of operating, the operator is confronted with a situation which is completely unfamiliar for him. It is thus the process of accommodation which becomes, for a time, dominating. It leads to the transformation and reorganization of available schemes, which gradually produce new compositions of schemes allowing the reproducible management of the new class of situations. It comes out from these observations that the question of the difference existing between the schemes and initial representations of the operators and, the schemes and representations necessary to control the machine are crucial in the ergonomic design of the human-machine co-operation. Within this framework, two complementary approaches are conceivable.

The first option is to seek to reduce the difference between the existent schemes of the operator and those appropriated to the control of the machine. The approach consists in considering the machine as the prolongation of the motor functions of the user. Thus, when such an assumption is relevant, the user will tend to give his own characteristics and properties to the machine ([Gaillard93]). In this case, an anthropomorphic design seems to be well suited in order to build a directly appropriable tool by the assimilation process. However, in many situations, such a human-machine compatibility is not easily reachable. This is why the second option aims at taking note of the mismatch between schemes necessary to control the machine and those of the user, insofar as the difference is regarded as not significantly reducible. Consequently, the ergonomic design will seek to facilitate the conceptualization of the difference by an accommodation process based on learning.

3.3 Application context

Within the framework of maintenance in residence, a robot able to move, to handle usual objects and to perceive its environment can be an essential complement to the other technological means placed at the disposal of a person with reduced autonomy for her/his safety and in order to ensure other services like the tele-survey, the tele-health, and the social relation… If we take the principle that the robot is semi-autonomous, the user remote controls it (Fi3).

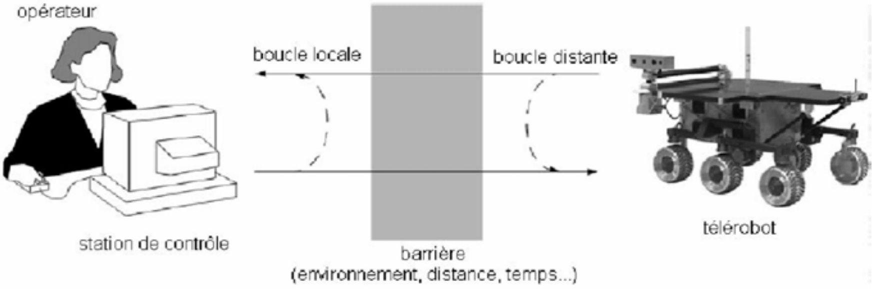


Fig. 3. Remote control situation (adapted from [Fong01])

According to the type of user and use, problems are quite different. The taxonomy defined by the community of the human-robot interaction is a useful tool which makes it possible to determine precisely the nature of the interaction between the robot and its user. Taxonomy gathers the criteria of classification of the interaction in three categories: structural, relational and operational. If one retains only the relevant criteria for our application, this system is composed of user, handicapped or not, and of a robot able to move, to seize and handle objects. In addition, at a given moment, the robot interacts with only one person. Finally the space-time criterion which belongs to the operational category of taxonomy makes it possible to distinguish between three types of tasks. If the robot is remote controlled by the person with reduced autonomy, the robot and the person share the same space i.e. are in the residence of the person. In this case, two situations are possible, the robot is remote controlled either in sight or out of sight. On the contrary if the user is a distant person who remote controls the robot via the Internet, the robot and the user are at different places.

3.4 Control modes

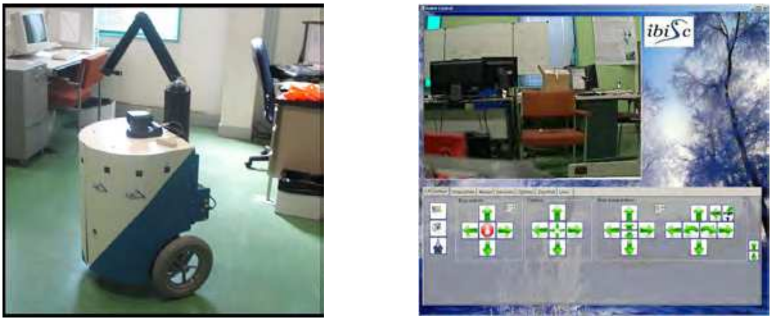


Fig. 4. ARPH system (a) and Human-Machine Interface (b)

The robot of the ARPH project (Figure 4a) is composed of a mobile platform with two driving powered wheels and a MANUS manipulator arm. An ultrasonic sensor ring makes possible to avoid obstacles. The robot is equipped with a pan-tilt video camera. The system – robot and video camera - is remotely controllable via a client/server architecture and a wireless Wi-Fi network.

The user controls the machine from a computer located at a distant place (Figure 4b) by using different control modes.

In the Manual mode the operator controls directly all the degrees of freedom of the robot.

In the Assisted mode, the control of the degrees of freedom of the robot is shared between the user and the machine. This type of mode is the most concerned by a design approach based on appropriation which aims at determining the most efficient way of sharing the control of the robot between the user and the system. As such a type of mode depends on the task and the robot used, it is possible to imagine a large variety of assisted modes.

The key idea is to facilitate an appropriation by the process with a dominant of assimilation (Figure 1) by giving the robot human-like behaviors. The assumption is that the user builds by analogy, more intuitively, a realistic mental representation of the robot. The implementation of behaviors of the human type on the robot rises from the multidisciplinary step made up of the four following stages:

1- Identification of relevant human behaviors, generally perception-action loops

2- Modeling of the behavior's candidates to extract from them the principal characteristics

3- Translation of the model resulting from the neurosciences to an implementable behavior on the robot

4- Evaluation of the mode

We present three examples of assisted-mode design.

3.4.1 Human behaviour candidates

Visio-motor anticipation seems to be a good behavioural solution to palliate the difficulties of space perception and representation. During a displacement, the axis of the gaze of a person systematically anticipates the future trajectory. Indeed, in curve trajectories, head orientation, more precisely gaze direction, of the person is deviated on the inside of the trajectory. This would guide the trajectory by a systematic anticipation of the trajectory direction with an interval of 200 milliseconds ([GRA96]).

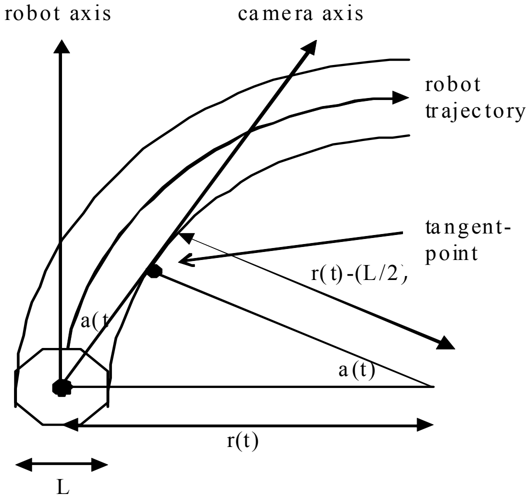
An analogy can be made between the direction of human glance and that of the pan-tilt video camera which equips the robot, so we sought to implement on the robot this type of behavior. The foreseen consequence was an improvement of the speed of execution of the movement and the fluidity of the trajectories of the robot. Taking into account the functional architecture of the robot, two implementations of the visual anticipation during a displacement are conceivable: (i) by automation of the anticipatory movement of the camera according to the orders of navigation which the operator sends to the robot or conversely (ii) by automation of the navigation of the robot starting from the orders which the operator sends to the video camera.

3.4.2 Platform mode

In the situation “I look at where I go” that we call in the following “platform mode”, the operator directly controls the displacement of the robot. The video camera is oriented only by a reflex action following the trajectory followed by the robot. From the analogy carried out between the human glance and the mobile video camera, the latter is automatically oriented in direction of the point of tangent to the internal trajectory of displacement, i.e. at the place where visual information is most relevant to guide the locomotion (Figure 5). The angle a(t) between the axes of the robot and the pan-tilt video camera must be conversely proportional to the curvature radius of the robot trajectory in order to move the camera towards the tangent point.

By using the trigonometrical properties, $a(t)= \arccos [1- (L/2)/r(t)]$, where $L/2$ is the half-width of the robot.

Due to the fact that the ARPH robot is speed controlled by the user, the radius r(t) is obtained by the ratio between the linear and rotation velocities of the robot.



$$\boxed{a(t) = \arccos(1-(L/2)/r(t))}$$

Fig. 5. Implementation of the behavior "I look at where I go"

3.4.3 Camera mode

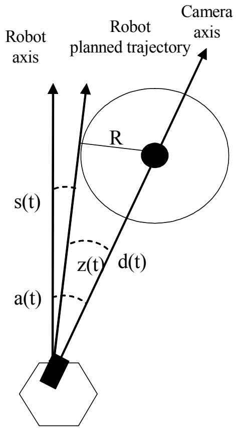


Fig. 6. Implementation of the behavior "I go where I look at"

In the situation “I go where I look at” that we call in the following “camera mode”, the operator controls the orientation of the camera directly. The orientation of the robot will be computed from the orientation of the camera. Contrary to the preceding model, the vision is now actively controlled by the user and the displacement of the robot is automated. The model is inspired from the behavior of visual anticipation which consists to fix a reference mark and to maintain it in its visual field in order to describe an ideal trajectory around it. The great advantage of this situation is that it makes it possible the operator to visualize continuously the obstacle nearest to the robot limiting the collision risk.

The angle of navigation of the robot s(t) is defined by the difference between the angle a(t), between the axes of the robot and the pan-tilt video camera, and angle z(t), between the axis of the camera and the tangent to the robot planned trajectory. By using the trigonometrical properties $z(t)= \arcsin[R/d(t)]$ where d(t) is the distance between the robot and the reference mark of navigation and R, the radius of a safety zone around the reference mark. $d(t) = [v(t)/da(t)/dt]\sin[a(t)]$ where v(t) is the linear speed of the robot.

A comparison between manual, platform and camera modes has been realized. The experimental protocol of the evaluation, the results and the data analysis are developed in section 4.

3.5 Robot Learning by the user

The control of the robot is a complex task. It is not enough to give human-like behaviors to the robot so that it is usable. It is necessary to reduce the difference between the mental representation of the instrument that is made by the user and its real use using an accommodation process (cf Figure 1). To assist user for the acquisition of a suitable mental representation of the robot operating, a serie of targeted trainings of progressive difficulty is proposed. The use of a robot simulator presents several well-known advantages to carry out this phase of training. The simulation makes it possible to reproduce, in full safety for the person, situations which can be very varied or not easily realizable in reality. Moreover, the simulator allows a strict control of the experimental conditions and provides exploitable data to judge the degree of acquisition of the skills to be learnt. In addition, it ensures time saving and a reduction of logistic costs of evaluation when the subjects are disabled people. One of the main question is the reproducibility of the results when the user passes from a virtual situation to a real situation, in other words if there is a transfer of skills or knowledge acquired in simulation to the real world. The transfer of training is a relatively important process from an adaptive point of view, “because it is rare that one finds in the life an activity which is exactly that which was learned at the school” ([Lieury04]). However, its implementation is often difficult. “Knowledge is often so related to the context of its acquisition that the individuals meet great difficulties of using them in different contexts” ([Roulin06]). We thus carried out an experiment intended to highlight a positive transfer effect between a situation of training on a simulator of the robot ARPH in its environment, and a situation of transfer consisting in controlling the real robot. One speaks about positive transfer when the effect of former knowledge acquired in a first situation improves the efficiency in a second situation. This efficiency can be translated, for example, by a reduction of the error number, a reduction of the execution time, or an increase of the number of good answers. We formulate the assumption that a preliminary training with the simulator of robot ARPH in a situation whose characteristics are similar to those of the real situation should involve a positive transfer. In short, a dedicated training in simulation should facilitate the use of the real robot.

The experimental protocol of the evaluation, the results and the data analysis are developed in section 4.

3.6 Delay treatment

3.6.1 Related work

In our case, the Internet will be used as the communication medium. That indicates another important problem by adding a transmission delay between the master and the slave computers. The natural reaction to this delay is to adopt a move-and-wait behavior: the operator issues a command and waits for the robot's feedback. This greatly slows the process of controlling the robot. The delay encountered on the Internet has two main sources: the physical distance that the signals have to travel and the network congestion. One can reasonably expect a continuous component (more or less constant, likely corresponding to the physical distance between the remote operator's and the robot's locations) and a variable component (spikes in the delay, only appearing from time to time, likely because of network congestion) [Moon00], [Garcia03].

When the delay's value is not above a certain threshold (usually, around 300 ms [Henderson01] for tasks involving only a video feedback, but it depends on the type of task and on the operator himself), the operator is not going to be disturbed by it. If it exceeds this value, one needs to think of ways to diminish its influence on the operator. One way of helping the operator overcome the influence of the delay is to adapt the video feedback to match the delay. Another way is to reinforce the human-robot cooperation by relying more on the robot's capacities to manage itself. The former category of aids for the remote control (adapting the video feedback), has the advantage of not requiring a very computational-powerful robot. One of the problems with it is that, for certain types of video feedback aids, it requires some a priori knowledge of the environment and for the robot to be the only thing moving (or for the environment to evolve very slowly).

An aid that seems to give very good results is to use a predictive window [Baldwin99]. In this case, the camera acquires a 360° image and only a portion of this image is presented to the operator. The purpose of this is to reduce the network traffic, as only the selected portion will be transmitted to the operator. It is also possible to think of the image as a texture, so that when the camera is moving, the texture that was represented by the initial image will be deformed as to match the new viewpoint ([Cobzas05]). If it isn't too complicated to create a model of the environment and if the environment isn't expected to change (or to do it slowly, so as to have time to update the model), we could think of using a representation in virtual reality [Bares97]. In this way, the robot will be controlled in the virtual world. If creating the whole model isn't possible, using a representation of the robot (or parts of it – such as its end effector) could be used to help alleviate the problems induced by an unreasonable delay. Such is the case of [Friz99], who uses a mechanical arm over a workbench, whose task is to create different structures using wooden blocks. In this thesis work, the operator can use three cameras located around the workbench and a camera that's attached to the arm's base. Visual cues concerning the future position and orientation of the robotic arm are given. They are represented in such a way that their interference with the real image is minimal (according to the “less is more” principle).

3.6.2 Proposed solution

What we aim for in our system is to either give the impression that the communication between the remote operator and the robot happens with virtually no delay (just the minimal time necessary to process the information) or to allow the operator to see the position where the robot should be if there was no delay.

As already mentioned, the communication between the remote operator and the robot will pass through the public Internet. This means that it is possible that we will have to deal with the influence of the delay that will exist between the robot and the operator’s computer.

The article that served as the basis of the moving window aid is [Baldwin99]. Because we don’t have a panoramic camera (60° field of view), if the robot will turn too much to the side, a black area will be used instead of the image that should have been there. An improvement on this kind of aid is that, when the robot advances, a portion of the image (central, if the robot moves straight ahead) will be zoomed in on the display interface. This will give the impression of the robot moving forward for the operator. If the robot is to move backwards, the selection window will enlarge (while, as in the previous case, the image will be scaled as to fit into the display interface). A scaling factor can be applied just before displaying the modified image, as to be able to control the sensation of turning/advancing. All these modifications will be realized on the operator’s side of the remote control system. For the deploying stage, we are currently investigating how to synchronize the clocks of the robot and of the remote operator’s computer using the network time protocol [Mills90]. According to [Elson02], the maximum difference between the two clocks will not be superior to 2 ms, which is good enough for our case, when the operator is disturbed only when the delay exceeds 300 ms [Henderson01].

A second type of aid is to use a simplified model of the robot, which will be presented on screen to the operator when, because of the delay, the image displayed is not the one that he should be seeing. This method has its roots in [Friz99]. For our case, we don’t always display the virtual robot (called a phantom from now on). The phantom will appear only when the distance between the position that the operator currently has for the robot and where the robot should be (in the absence of delays) goes above a certain value (0.2 m for the tests).

4. Evaluations

4.1 Comparison of human-like control modes

The evaluations concern the assisted modes presented in §III.4. The hypotheses of our work were the following ones. Firstly, a situation in which the pan-tilt video camera is mobile and points towards the future trajectory of the robot, should lead to better performances in terms of control of the trajectory than a situation in which the camera is fixed and always pointing along the axis of the machine. Secondly, by reference to works evoked above, a strategy of the type “I go where I look at” will lead to a more optimized trajectory than “I look at where I go”.

4.1.1 Experiment results

Three command modes have been tested, two with anticipation -camera mode and platform mode- one without anticipation. This last mode corresponds to a fixed camera, aligned with the robot axis, and a manual control of the displacement of the platform.

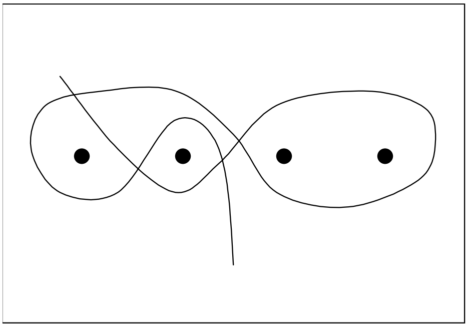


Fig. 7. Schematic representation of the trajectory.

The hypothesis is that in "camera" control will be better than in "platform" mode. The "fixed" mode is used as a reference. The operator has to realise with the robot the trajectory given in Figure 7. The criteria of comparison are performance parameters: trajectory execution time, collision number, stop number and a behavioural parameter: trajectory smoothness.

About collisions (Figure 8), anticipation conditions ("camera" mode plus "platform" mode) present significantly less collision than "fixed" mode (p<0.01). There is no significant difference between "camera" mode and "platform" mode.

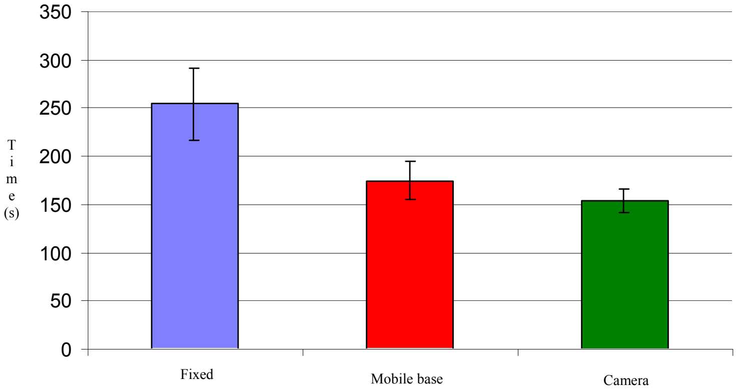


Fig. 8. Mean time of execution.

Concerning number of collisions (Figure 9.), anticipation condition have significantly less collisions than "fixed" condition (p<0.03). But there is no significant difference between "platform" mode and "fixed" mode. There is also no significant difference between "camera" mode and "platform" mode.

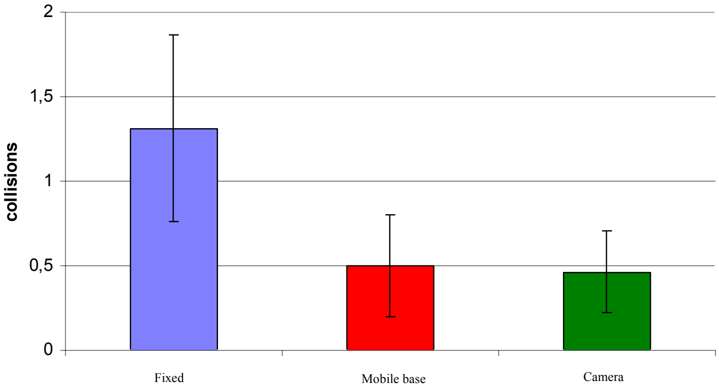


Fig. 9. Mean number of collisions.

About number of stops, both anticipation conditions are very significantly better than "fixed" condition.

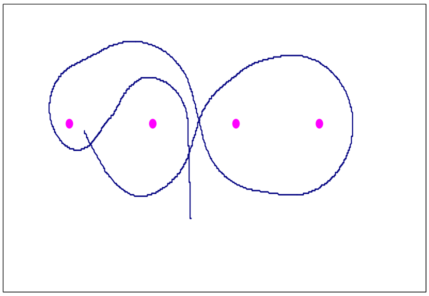


Fig. 10. Trajectory in anticipation mode.

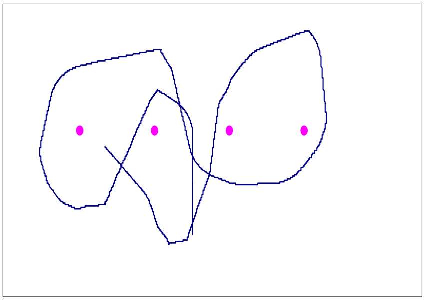


Fig. 11. Trajectory in "fixed" mode.

Let us now examine behavioural parameters. Figure 10 shows a trajectory in anticipation condition. Figure 11 shows a trajectory in "fixed" condition. The second one is more angular than the first one. [Peruch99] proposes a solution to quantify this. Number of occurrence of