Feedback and Imprecise Information Processing in a Voice Interface to a Robotic System

Ramón P. Ñeco, Nicolás García, José M. Azorín,
M. Asunción Vicente, Luis M. Jiménez, Rafael Puerto

Abstract—**In this paper two methods are presented so that it is possible to process the feedback of the errors produced by a teleoperated robot, using a voice interface with natural language processing capabilities. The solutions proposed in this paper had its initial motivations on the following questions: (1) To find a method to process the feedback produced by the user during the execution of a teleoperated command, expressed using a voice interface, and (2) how can the system use the information provided by the feedback process so that the robot behaves in the way desired by the operator in successive executions of the task.**

Keywords— **Artificial Intelligence, Fuzzy Logic, Human-Computer Interaction/Interface, Learning Systems, Teleoperation.**

I. Introduction

IN this paper a module of feedback processing and error correction in a real-time robotic teleoperated system is presented. The global system consists of the following modules: speech recognition module, bilateral control module, vision module, learning module, and the robot controller and interface. The speech recognition module includes a natural processing system that admits expressions of high level commands in natural language, and provides a low level program, directly executable by the robot.

The speech processing module provides the operator the possibility to use his/her natural language to correct in real time the errors that the robot can make during the execution of the requested command, including feedback information toward the robot. This information received by the robot can be used to adapt its behavior in some aspects, in such a way that it improves the future execution of the same command. Two very related approaches have been used: a probabilistic approach (changing parameters of a density probability function) and a possibilistic approach (changing the parameters of a membership function of a fuzzy set).

The objective of the work presented in this paper is to obtain methods so that the teleoperated robot improves the execution of tasks that has already learned how to carry out in a satisfactory way, or whose execution steps are specified in the database, but with missing details (basically parameters) that can still be refined or to be adjusted using feedback with natural language. Also, it is possible that these parameters have acquired standard values during the learning process of the task using learning techniques [1], [4], and they should be adapted to the necessities of each operator or to the specific context in which the task is being developed. These improvements can be made during the execution of the task by the robot, using qualitative expressions (and possibly also quantitative) in natural language.

Some authors have developed works partially related with the topic described in this paper. In the work presented in [3], the authors developed a first adaptation technique applied to differentiate the different executions of the same verb to different execution circumstances or to different users, focusing on the specific application of control devices for the disables. In the work described in [7] the authors show a method for the pragmatic inference as a necessary complement for command languages. The authors develop a method to model and to recognize the intention of the human operators that relates sequences of domain actions ("plans") with changes in some pattern of the environment of the task. In the work described in [2] a formal model was designed in order to represent computationally the intentions of the user in dialogue systems.

II. Experimental Environment

As in all teleoperated systems, the experimental system considered here consists of a remote environment, as well as a local environment that controls and supervises the remote environment. The devices that interact with the task, as well as the sensors that send to the operator the information of what is going on in the remote environment have been located in the remote area (figures 1 and 2). The elements of the remote environment are the following: A robotic arm (Mitsubishi PA-10) of 7 degrees of freedom, which executes the commands emitted by the operator; a computer that acts as the robot controller; a computer for image processing that captures the images from the cameras located in the remote environment and recognizes the objects that are present in the scene; wide range area cameras that are used for three-dimensional recognition of the scene, as well as to provide to the operator visual information about the remote environment; a camera located at the end of the robotic arm, to obtain more precise visual information in the manipulation of the objects on the part of the robot; a force sensor that allows to know the force exerted by the robot in the execution of the task, information that is transmitted to the master manipulator so that the operator has the information about the force and in this way the teleoperation degree can be increased.

The authors are with the Automatic and System Engineering Department, Miguel Hernández University (UMH), Elche, Spain. E-mail: ramon.neco@umh.es.

This work has been supported by the Spanish Comisión Interministerial de Ciencia y Tecnología (CICYT) through project TAP98-1083-C02-01.

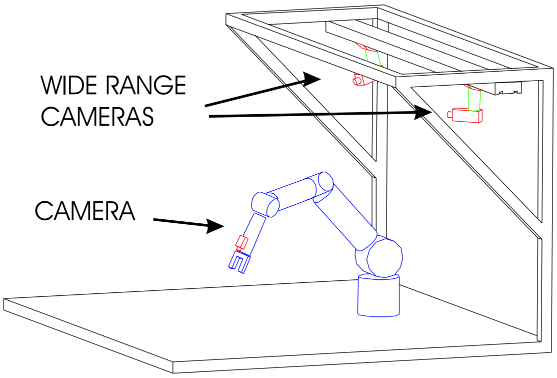

Fig. 1. Configuration of the elements in the remote environment.

Fig. 2. Teleoperated system used for the experiments on natural language.

In the local environment all the elements such that the operator can interact to send and to receive the commands to the remote environment can be found. These elements are the following: graphic computation system, by means of which the operator knows in real time the state of the task and can control in a complete way the remote system; master device; a computer for speech recognition that performs the speech recognition together with the natural language processing so that the operator can command the teleoperated system using voice commands [6]. Also, this computer implements the bilateral control of the system: from the movements that the operator carries out in the master device, the computer sends them to the remote environment so that the robot manipulator reproduces the movements. Likewise, the computer transmits the received information of the torques and forces exerted by the robot to the master device in order to allow the operator to perceive this information.

III. Error Feedback Using Natural Language

The feedback will be used to modify continuous output signals (variables) generated by the robot as a part of the response to a command in natural language. For example, let us imagine that the robot is requested to place a piece on a region of a table, and that the system has stored in the learning database the steps to carry out this simple task. It is possible that the space in which the robot could place the piece in this command is very wide, and that a range exists along the longitude of the region of the table where the placement of the piece is considered "correct" or "acceptable" for the user. The commands given in natural language can be something as "*put the screwdriver at the other end of the table*", without specifying concrete coordinates. However, the operator has usually a very clear idea of a range at the other end of the table where it is acceptable to place the piece, and another range where it is not. After the execution of the command, a very natural way to allow the feedback from the operator to the robot's action is to emit such correction commands as "*a little more to the left*", "*more slowly*", "*farther*", "*much more to the right*", etc.

The desired effect of these commands is the correction of the current execution of the command in real time by the robot and that as a consequence it also corrects the future executions. In this way it is avoided that the operator should specify details and parameters in a "non natural" way, indicating the exact values, as in the following command: "*put the screwdriver four cm away from the right side of the shelf, parallel to the front edge and three cm out of the same one*" that are not natural for operators or users non specialists in Robotics. Instead of this, the present work is focused on a feedback process: the robot executes the emitted command and the operator interacts with the robot during the execution of the task. The extracted information from the interaction with the user is also stored in the database of tasks, so that after the possible future request of the same task by the same user, the system can adjust the robot's behavior using natural language.

In a similar way to the use of other classical devices to guide a robot (master and joystick), two types of feedback using expressions in natural language can be distinguished:

1. *Position feedback*, which corresponds with commands to the robot in cases in which some position (final or partial) of the robot has not been completely satisfactory. Typical commands that fall in this category are: "*more to the left*", "*much more to the left*", "*a little more to the right*", etc.

2. *Accuracy feedback*, which corresponds with commands that refer to the value of some magnitude (forces, speed, ...) and they can be expressed using natural language commands like "*press with more care*", "*not so slowly*", "*much more quickly*", etc.

From the observation of the practical execution of the command by the robot, and using expressions in natural language as the previous ones, the operator can express the desired effect of the command, also retaining the same effect in future executions of the same command. The objectives of the feedback subsystem using natural language described in this paper are, therefore, two:

1. To correct in real time the robot's current action.

2. To store the derived information obtained from the correction so that this information can be used in later executions.

In this section it will be assumed that the system has already a set of tasks that the robot has learned how to execute, or that in some way it has a list of sequential actions associated to each one of the tasks of this set. This list of actions will be stored in the database of tasks of the system, and the robot's controller can access to it to execute the actions. It is also assumed that most of these tasks depends on a set of parameters, that is, their execution depends on an n-dimensional vector $\mathbf{p}$ of parameters. These parameters can be positions at each step of the execution of the task, speeds, forces, etc. The value of these parameters is what will be adapted or learned through the feedback types that will be described in this section.

A. Possibilistic Representation Techniques

Depending on the dimension n of the vector of parameters $\mathbf{p}$, it can be distinguished among one-dimensional tasks ($n = 1$), two-dimensional ($n = 2$),..., n-dimensional tasks. Since the specification of the "acceptable" values of the parameters won't be made using concrete values, but emitting feedback commands, the first proposed option uses the ideas of fuzzy logic to represent the possible values of the parameters [8]. Each parameter will belong to a fuzzy set P that will represent the acceptable fuzzy values so that the robot carries out the task.

A very simple example for a one-dimensional task will be considered. Let us assume that the robot has learned how to place a tool of the environment in its storage place. This task will be executed after a command like "*store the screwdriver in the shelf*" or "*put the screwdriver in the shelf*". The robot already knows the steps of this task to carry out it in an autonomous way that will be basically the following ones: (1) If the gripper is closed, open it; (2) Move the end of the arm to the adequate grasping point; (3) Close the gripper; (4) Move the end of the arm to the selected point to store the tool; (5) Open the gripper; and (6) Move away from the tool.

In the surface of the destination location of the tool a range can exist where it is valid to place it, from the user's point of view. In this simple example, the parameter that is sought to adjust using feedback is the appropriate interval along the surface of the shelf where to store the tool. This parameter is denoted as $\mathbf{p}$, and will be one of the three coordinates of the destination point in the three-dimensional space. The rest of steps can have also parameters susceptibles of being optimized. For example, the grasping point of the piece is defined in a generic way, parametrized in order to allow the adaptation to the specific execution. This parameter will be used in the step (2).

The variable $\mathbf{p}$, which is the objective of the feedback belongs to a fuzzy set P that represents qualitatively a certain way of placing the tool. For example, to place it preferably at the center of the destination, a fuzzy set can be used as the one that is shown in figure 3, where in the X and Y axis the value of the parameter $\mathbf{p}$ and the degree of membership to the set are represented, respectively. Similarly, the fuzzy concept "*preferably to the left or in the middle*" can be given by the fuzzy set whose membership function is shown in figure 4.

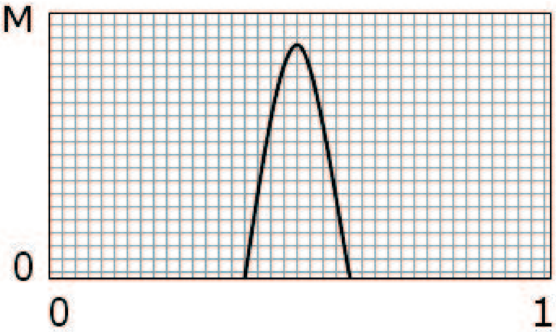

Fig. 3. Fuzzy concept "*preferably in the middle*".

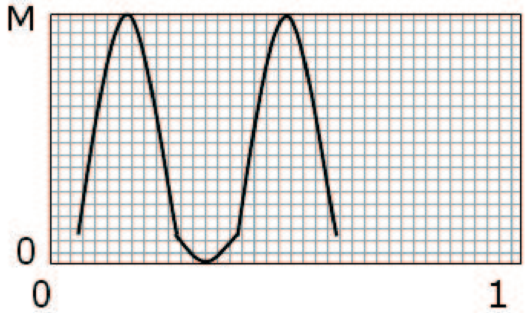

Fig. 4. Fuzzy concept "*preferably to the left or in the middle*".

Following these ideas, the parameters that are part of the vector p are not considered to be numeric parameters (real, integer,...), but rather they are *fuzzy* numbers, which allows their adaptation or their generation in a flexible way. The fuzzy numbers constitute a special subset of the fuzzy sets that are defined on the set of the real numbers [5].

B. Stochastic Representation Techniques

Another considered possibility to represent the imprecise information in the feedback to the robot using the speech interface is the direct use of probability density functions. For one-dimensional tasks, simple probability distributions of one variable can be used, $f(x)$, where x is the value of the parameter, and f is the probability density function. Using these ideas, a stochastic approach can be designed as an alternative representation of the parameters that are desired to adjust. The robot will use as working parameter values of $\mathbf{p}$ that belong to an interval around the mean of the distribution, with a dispersion that is given by the variance.

The choice of the function f depends on the robot, the environment and the specific task. In the work described in this paper a well-known function has been used, specially in learning experiments [3], the *beta* function or distribution whose probability density is given by:

$$f(x) = \frac{g(\alpha+\beta)}{g(\alpha)g(\beta)} x^{\alpha-1}(1-x)^{\beta-1}, \quad \alpha > 0,\ \beta > 0,\ 0 < x < 1 \tag{1}$$

where g is the *gamma* distribution, which is defined as:

$$g(x) = \int_0^\infty e^{-t} t^{x-1} dt \quad (2)$$

The computation of g for integer numbers is simple, since it interpolates the factorial function ($n!$), in such a way that $g(n+1) = n!$. The mean and the variance of the distribution f can be easily obtained from the following expressions:

$$\bar{f} = \frac{\alpha}{\alpha + \beta} \quad (3)$$

$$\sigma_f^2 = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} \quad (4)$$

The beta distribution is frequently an appropriate model to describe the random behavior of the percentages. This takes advantage in the work presented in this paper to represent probabilistically behaviors that depend on one or several parameters. Depending on the values of α and β , the function will represent different behaviors. If $\alpha = \beta = 1$, then all the values in the interval $(0, 1)$ are equally probable, representing the idea that the user doesn't care too much the value of the parameter while it belongs to a specific interval. In the simple example described in the previous section, this is translated to the fact that it is not relevant the exact point in which the tool is placed in the shelf, in the range of its longitude (that is, in the range of the function f). For the values $\alpha = \beta = 1$ an uniform distribution in the interval $(0, 1)$ is obtained, that is usually the initial distribution for the task.

In the extreme case that both parameters α and β are equal to zero, a so called rectangular distribution is obtained, since the probability density function is reduced to the constant function $f = 1$, and its form is similar to a rectangle whose ends are excluded. Another singular case occurs when one of the parameters is zero and the other one is the unit. The curve of the distribution corresponds then with two straight lines of opposed slopes.

C. Behavior Adaptation

Before feedback takes place on the robot's behavior, the operator should emit a command. When the robot have to decide the value of the parameter vector $\mathbf{p}$ associated with this command, from the membership function of a fuzzy set or from the probability density function, a sampling algorithm is used, based on these functions. There exist several algorithms to carry out this sampling. In the work presented in this paper several sampling techniques have been used, being the most elaborated the one that is described next. First, an interval $[x_{min}, x_{max}]$ for the generation of the values and a superior limit M of f in this interval are defined. The algorithm consists of generating pairs of random numbers and presenting the first one as the result only if the second satisfies a condition. If the condition is not satisfied, a new pair is randomly chosen. The steps of the algorithm are the following:

1. Generate a pair of random numbers (a, b).
2. $x = x_{min} + (x_{max} - x_{min})a$.
3. If $Mb \leq f(x)$, then the output is x, end.
4. Go to step 1.

Once the robot has chosen the value of the parameter from the current function that represents the knowledge, the user can emit correcting commands associated to that parameter. As reply to the correcting command, the robot executes an immediate action and it adapts the internal representation of the parameter depending on the corrective command. Therefore, depending on the nature of the feedback command emitted by the operator, the adaptation of the function will be different.

C.1 Position feedback

The position feedback in natural language should have an immediate effect on the robot's position, besides modifying the membership function associated to the position parameters for that task. For example, after a feedback like "*... a little more to the left ...* ", the robot will move a longitude toward the left and it will also adjust the coordinates for the next time it executes the task.

The magnitude of the displacement as a consequence of the position feedback depends on the specific feedback command that the operator has emitted. A simple way of modelling these displacements is to define n different constants $c_1, c_2, ..., c_n$ for each position feedback category, such that:

$$c_1 < c_2 < \cdots < c_n \quad (5)$$

The commands that generate relatively very small movements will come defined with the constant c_1, those that generate relatively small movements with the constant c_2, those that generate mean movements with the constant c_3, those that generate big movements with the constant c_4, those that generate very big movements with the constant c_5, and so on. The decision about the value that should have the variable n and the concrete values that should have the n constants c_i depends on the nature of the parameter and on the specific task and they are determined by means of a specific design or in the learning phase of the tasks.

In the case of the stochastic learning, the specific value of the robot's displacement is multiplied by the dispersion of the density distribution function. The dispersion can be computed using the standard deviation σ_f. In this way, when the robot is given an order of position feedback, the constant c_i corresponding to the corrective command will be obtained and it is multiplied by the standard deviation, σ_f, of the distribution function f associated to the parameter. For example, if the following feedback command is emitted "*... much more to the right ...* ", and the constant associated to this command is c_i, then the displacement will be given by the product $c_i\sigma_f$.

In the case of the possibilistic technique used in the experiments, the constant is multiplied by the dispersion of the membership function. Triangular, trapezoidal and a special type of membership function associated to the denominated fuzzy numbers have been used in the experiments. These special type of membership functions are centered around a number, with arbitrary functions to both sides of the center, and they are called *Left-Right fuzzy numbers* (*L-R*), whose membership functions are defined as:

$$f_n(x) = f(x; a, \alpha_L, \alpha_R)_{LR} = \begin{cases} L\left(\frac{a-x}{\alpha_L}\right) & \text{if} \quad x \leq a \\ R\left(\frac{x-a}{\alpha_R}\right) & \text{if} \quad x > a \end{cases} \tag{6}$$

where a is the center of the fuzzy set, α_L and α_R are positive real numbers which represent the dispersion of the function, and R and L are two functions that satisfy the following conditions:

- $R(0) = L(0) = 1$;
- R and L are non-increasing in the interval $[0, \infty[$.

To decide the change in the robot's position as a consequence of the feedback with voice, as it has been indicated before, the constant c_i is multiplied by the dispersion corresponding to the membership function, given directly by the constants α_L and α_R in the case of *L-R* fuzzy numbers. If triangular or trapezoidal membership functions are used, the dispersion is given by the distance between the center and the extremes of the triangle or trapeze.

Besides causing a change in the robot's position, this feedback type should cause a change in the generation of the parameter for future executions of the same command.

In the case of the stochastic learning, the method used to change the robot's future answer is to modify the mean of the distribution of the defined probability to make it equal to the current value of the parameter (the value obtained after the feedback). Also, after each corrective action, the system modifies the distribution in such a way that decreases the variance. This is made assuming that after several executions and corrections on the part of the operator of the specific parameter, the deviation of the distribution should tend to decrease, since the more corrections carried out, the bigger "reinforce" of the learning of the parameter. A way of getting this is making the new variance equal to the square root of the previous variance:

$$\sigma_f(i+1) = \sqrt{\sigma_f(i)} \tag{7}$$

In the case of the adaptation in the possibilistic approach, the functions are modified in a similar way. The center as well as the dispersion of the membership function of the fuzzy number are adequately modified. Therefore, in the representation of *L-R* fuzzy numbers, the dispersion will be given by the parameters α_L and α_R. The variation of these parameters after a position feedback command is computed in a similar way to the case of the standard deviation:

$$\alpha_R(i+1) = \sqrt{\alpha_R(i)} \tag{8}$$

$$\alpha_L(i+1) = \sqrt{\alpha_L(i)} \tag{9}$$

In this way it is possible to decrease the dispersion in the random selection of the parameter whose value has been corrected.

C.2 Accuracy Feedback

As it has already been mentioned, an accuracy feedback is the emition of interaction commands that refer to the value of some magnitude (forces, speed,...). It is expressed with commands in natural language as "*press with more care*", "*... not so slowly...*", etc. This feedback doesn't cause an immediate change in the position and the robot's current state, but rather it causes a change in the function that is used to generate the magnitude to which refers the command. As in the previous case, the change can be reflected changing the variance of the distribution, or the dispersion of the membership function of the fuzzy number directly. If the command refers to the necessity of increasing the value of a magnitude, then the variance or the dispersion should be increased (for expressions like "*more quickly*").

On the other hand, if the command refers to the necessity of decreasing the value of a magnitude, then the system should decrease the variance or the dispersion explicitly (for example, for expressions like "*... with less force...*"). The change that is made to the variance or the dispersion should be proportional to the current values of variance and dispersion, so that the change takes place in a controlled way. A form of getting this is to increase or to decrease the dispersion according to the absolute value of the first derivative of the dispersion with respect to one of the parameters of the distribution (for example, with respect to α , in the case of the beta distribution, or with respect to the center, in the case of the possibilistic techniques), so that an increase or decrease following the direction of the gradient takes place:

$$\sigma_f(i+1) = k\frac{\partial \sigma_f}{\partial \alpha}, \quad k > 0 \tag{10}$$

If approval expressions are allowed in the language after the execution of a task (expressions as "*it's OK*", "*well*", "*that's right*", etc.), which can be understood as positive feedback commands, then the mean of the distribution can be changed in such a way that it is equal to the value of the parameter given by the robot, also diminishing the variance (for example, to their square root). In the system implemented in this paper, this modification of the parameters was not considered, because it would overload the real-time system excessively. If the user doesn't make any correction on the parameter after the execution of the command, the internal representation of this parameter is not changed.

IV. Validation of the Methods and Conclusions

The group of techniques described in section 3 has been experimented in an environment of a teleoperated robot like the one described in section 2. In Table I a sequence of values of the parameter **p** is shown. This parameter represents the distance to the origin of coordinates considered in the environment for a command of placement of a piece with position feedback, using possibilistic techniques with *L-R* fuzzy numbers.

In Table II the same previous experiment is shown, using the beta probability distribution.

TABLE I

Position feedback with possibilistic techniques, using *L-R* fuzzy numbers.

p	position feedback
0.500	*more to the right*
0.725	*much more to the right*
0.911	*a little more to the left*
0.886	*much more*
0.780	*OK*

TABLE II

Position feedback with stochastic techniques, using the *beta* distribution.

p	position feedback
0.500	*much more to the left*
0.300	*a little more to the right*
0.401	*a little more to the left*
0.381	*a little more*
0.751	*OK*

In figure 5 a sequence of functions of probability distribution is shown after the processing of position feedback commands, indicating successive displacements toward the right. It can be observed that, besides moving the mean of the distribution, the process also diminishes the standard deviation in order to reinforce the fact that feedback has taken place on the associate parameter, reducing in consequence the effective range in the practice in which the robot obtains values using the sampling algorithm.

The initial values for the parameters for the beta distribution are always $\alpha = \beta = 10$ (except for the cases in which a mean different from 0.5 is explicitly indicated). Later on, as a consequence of each feedback command, new integer values are computed for *alpha* and β such that the mean of the distribution approaches to the new value obtained after the feedback, and such that the standard deviation diminishes approximately according to the constant $1/k$, where $k = 2$ in the reported experiments (figure 5). Note that since the values of α and β are integer numbers, the real-time method can only obtain an *approximate* solution to the equation

$$\sigma_f(i+1) = \frac{1}{k}\sigma_f(i) \tag{11}$$

which has to be solved to obtain the new standard deviation, and to the equation

$$\bar{f} = x \tag{12}$$

to obtain the new mean in each feedback iteration.

The main conclusion of the work described in this paper is that both representation methods allow the implementation of a feedback system that is very natural to an human operator, specially indicated when programming a robot to perform high-level tasks. The feedback and correction system is adequate for the application to a real-time teleoperated system, which allows the operator to "train" the robot in order to adapt in a very natural way the "intuitively" desired values for some parameters that define the robot behavior.

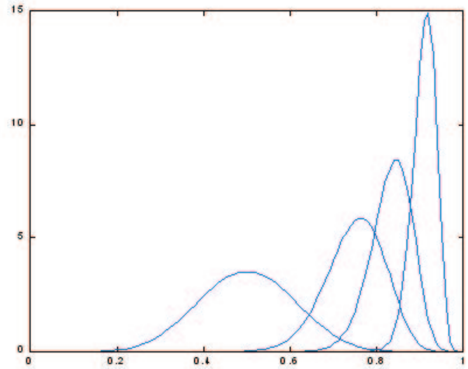

Fig. 5. Successive displacements towards the right of the mean of the distribution and decrease of the standard deviation as a consequence of the corresponding feedback commands.

The two described techniques allow the processing of the feedback following the objectives that have been described in the introduction. However, the possibilistic technique presents the advantage of being more *intuitive* in the definition and interpretation of the obtained representations of the parameters, besides being the most appropriate for a real-time robotic system, since the computational complexity required is smaller.

The probabilistic technique presents the advantage of allowing a more *precise* feedback of the parameters. Only in the cases in which the specific robotic task carried out by the operator requires high numerical precision, the use of the stochastic technique is justified, keeping in mind the required computational cost.

References

[1] Andreae, J.H., *Associative Learning for a Robot Intelligence*, Imperial College Press, London, 1998.

[2] Carberry, S., J. Chu-Carroll, L. Lambert, *Modeling Intention: Issues for Spoken Language Dialogue Systems*, Proceedings of the International Symposium on Spoken Dialogue, pp. 13-24, 1996.

[3] Crangle, C., P. Suppes, *Language and learning for robots*, Center for the Study of Language and Information, Stanford, 1994.

[4] Fernández, C., Puerto, R., Reinoso, O., Aracil, R., *Learning Strategies for Teleoperation*, 7th International Symposium on Intelligent Robotic Systems, Portugal, 1999.

[5] Matía, F., *Análisis y diseño de sistemas de control de procesos basados en lógica borrosa*, PhD. Thesis (in Spanish), E.T.S.I.I. Universidad Politécnica de Madrid, Madrid, 1994.

[6] Ñeco, R.P., Reinoso, O., García, N., Aracil, R., *A Structure for Natural Language Programming in Teleoperation*. In: Proc. of the 6th International Conference on Control, Automation, Robotics and Vision, Singapur, December 2000.

[7] Yared, W.I., T.B. Sheridan, *Plan Recognition and Generalization in Command Languages with Application to Telerobotics*, IEEE Transactions. on Systems, Man and Cybernetics, 21(2), 1991.

[8] Zadeh, L.A., *Fuzzy Logic = Computing with Words*. IEEE Transactions on Fuzzy Systems, 4, (2), 1996.