CONTROLAB: Integration of Intelligent Systems for the Control of a Robot Arm

Eliana P. L. Aude*        Júlio T. C. Silveira*       Fabricio A. B. Silva*
Mario F. Martins*        Henrique Serdeira*     Ernesto P. Lopes**

*NCE/UFRJ  **IM/UFRJ
P.O. Box 2324 - Rio de Janeiro - RJ 20001-970 - Brazil
e-mail: elaude@nce.ufrj.br

ABSTRACT

CONTROLAB integrates intelligent systems and control algorithms aiming at applications in the area of robotics. This paper focuses on the analysis of the word recognition and the trajectory definition systems considering an application in which a robot arm is commanded by voice to pick up a specific tool placed on a table among other tools and obstacles. Neural network architectures based on the backpropagation and the recursive models are proposed for the implementation of a speaker-independent word recognition system. The robustness of the system using the backpropagation network has been verified in totally uncontrolled environments such as large public halls for the exhibition of new technology products. Experimental results with the recursive network show that a carefully designed network structure is able to overcome the false alarm problem faced by the backpropagation network. The trajectory to be followed by the robot arm is determined through the analysis of image information and the use of the VGRAPH algorithm to avoid obstacles. The algorithm performance is analysed and compared with that achieved by the PFIELD algorithm.

1. INTRODUCTION

CONTROLAB [1, 2] is an environment which integrates intelligent systems and control algorithms aiming at applications in the area of robotics. It consists of four sub-systems: word recognition, computer vision, trajectory definition with obstacle avoidance and trajectory control. These sub-systems are currently used in the control of a SCARA robot arm.

Two different neural network structures are proposed to deal with the problem of speaker independent word recognition. These structures follow the backpropagation and the recursive models. Very encouraging results have been achieved with the proposed network structures and the robustness of the word recognition system based on the backpropagation network has been verified several times in uncontrolled and noisy environments such as large public halls for the exhibition of new technology products. A recursive network structure with four input vectors and two feed-back vectors has shown to be able to overcome false alarm problems.

The computer vision sub-system works on digitalized images captured by a CCD camera. It is able to automatically set the image threshold level under different lighting conditions. The object recognition within the image is performed through the extraction of geometric attributes and the comparison of these attributes with those stored in a library of known objects.

The trajectory control problem is tackled with the use of a multivariable self-tuning control algorithm which is able to cope with the non-linearities of the robot arm dynamics and to properly model the interaction forces among the robot arm joints.

When searching for a desired object to be picked, the robot arm performs a rotational movement over its work area in several steps. This movement is performed by the self-tuning controller under the guidance of the computer vision sub-system. At each step, the captured image is analysed. If the object is not present in it, the robot arm moves to the next position on its search path. On the other hand, if the object is found, a trajectory is defined to pick the object. In both situations, obstacles detected by the vision sub-system are avoided by the robot arm with the use of the VGRAPH algorithm [3]. The performance of this algorithm is analysed and compared with results achieved with the PFIELD [4,5] algorithm.

The practical results presented in this paper have been achieved for an application which allows any operator to use voice commands to select an object to be picked by a SCARA robot arm driven by 5 DC motors and with a CCD camera attached to it. Within this application, the word recognition sub-system identifies the object name spoken by the operator. Then, the robot arm searches for the object with the assistance of the computer vision and the trajectory definition sub-systems. When the computer vision sub-system locates the selected object and finds its rotation angle, the trajectory definition sub-system determines the free trajectory to be followed by the robot arm under the action of the multivariable self-tuning control algorithm.

A recursive network structure with four input vectors and two feed-back vectors has shown to be able to overcome false alarm problems.

The computer vision sub-system works on digitalized images captured by a CCD camera. It is able to automatically set the image threshold level under different lighting conditions. The object recognition within the image is performed through the extraction of geometric attributes and the comparison of these attributes with those stored in a library of known objects.

The trajectory control problem is tackled with the use of a multivariable self-tuning control algorithm which is able to cope with the non-linearities of the robot arm dynamics and to properly model the interaction forces among the robot arm joints.

When searching for a desired object to be picked, the robot arm performs a rotational movement over its work area in several steps. This movement is performed by the self-tuning controller under the guidance of the computer vision sub-system. At each step, the captured image is analysed. If the object is not present in it, the robot arm moves to the next position on its search path. On the other hand, if the object is found, a trajectory is defined to pick the object. In both situations, obstacles detected by the vision sub-system are avoided by the robot arm with the use of the VGRAPH algorithm [3]. The performance of this algorithm is analysed and compared with results achieved with the PFIELD [4,5] algorithm.

The practical results presented in this paper have been achieved for an application which allows any operator to use voice commands to select an object to be picked by a SCARA robot arm driven by 5 DC motors and with a CCD camera attached to it. Within this application, the word recognition sub-system identifies the object name spoken by the operator. Then, the robot arm searches for the object with the assistance of the computer vision and the trajectory definition sub-systems. When the computer vision sub-system locates the selected object and finds its rotation angle, the trajectory definition sub-system determines the free trajectory to be followed by the robot arm under the action of the multivariable self-tuning control algorithm.
Previous papers have described the computer vision and the trajectory control sub-systems [1, 2]. This paper focuses on the analysis of the word recognition and the trajectory definition sub-systems. Section 2 describes the main features of the two neural network architectures that have been proposed to perform word recognition. In Section 3, the procedure for searching a specific object placed within the robot arm work area is presented. In addition, the use of the VGRAPH algorithm to define the robot arm trajectory in the presence of obstacles is discussed. The use of the PFIELD algorithm to solve this same problem is briefly commented. Section 4 illustrates the object search procedure and presents experimental results which show the performance of the VGRAPH algorithm. A comparison with results achieved with the PFIELD algorithm is also presented. In addition, Section 4 shows practical results which demonstrate the ability of the proposed neural network structures to perform speaker independent word recognition. Finally, Section 5 summarizes the main conclusions of the paper and points out future developments for the CONTROLAB system.

2. THE WORD RECOGNITION SUB-SYSTEM

The CONTROLAB speech recognition sub-system is word oriented and speaker independent. It pre-processes the voice signal and uses neural network structures to perform the spoken word recognition.

2.1 Voice signal pre-processing

The voice signal is divided into 256 samples Hamming windows and an FFT analysis is performed. The FFT magnitude is applied to a set of 20 filters distributed according to the “Mel” frequency scale [6]. The energy of the outputs of these filters is then calculated and used by the recognition module to identify the spoken word.

The overlap between two consecutive windows varies depending on the type of neural network to be used in the word recognition phase. For the recursive neural network, 192 samples are always shared between two consecutive windows. For the backpropagation neural network, 40 windows need to be processed and, therefore, the number of samples shared by two consecutive windows is calculated considering this restriction.

2.2 Word recognition

Two neural network architectures are proposed for performing the speaker independent word recognition task: backpropagation and recursive.

The proposed architecture for the backpropagation neural network is an extension of the one presented by Krause and Hackbarth [7].

The first hidden layer consists of a 20 x 7 array of neurons. The neurons on each row are directly connected to the energy values produced by a single filter for the 40 frames. Therefore the compression of the number of energy values adopted by Krause and Hackbarth [7] has not been used. The network itself does the modelling of this compression.

Another novel aspect of the proposed architecture is the introduction of a second hidden layer consisting of 10 neurons which performs a compression in the frequency domain. Each of these neurons is connected to the outputs of the 14 neurons which make up two consecutive rows within the first layer array.

All the neurons are biased and the standard backpropagation training algorithm has been adopted. The non-linear function which generates the neuron outputs is the hyperbolic tangent.

For each word to be recognized a separate network has been used. With this approach, the network has shown to be able to distinguish minimal pairs. A word is recognized when its network produces an output value above a pre-determined threshold and all the other networks produce output values below another pre-determined threshold.

Although the proposed backpropagation network has shown to be very good at recognizing trained words spoken by different people, it sometimes lacks the ability to properly detect as “unknown” non-previous trained words. It frequently recognizes as one of the trained words a non-trained spoken word. To solve this problem, known as the generation of “false alarms”, the use of recursive neural networks has been investigated.

The basic recursive neural network architecture proposed by Morgan and Scofield [8] has a pair of vectors containing energy values produced by filters for two consecutive frames and a feed-back vector corresponding to the first-layer outputs in the previous training iteration. It was observed that by increasing from two to four the number of input vectors, the network becomes more robust for classification purposes. However, it becomes slower and is unable to detect small differences between similar words. On the other hand, by inserting an extra feed-back vector representing the outputs with a delay of 2, the network is still robust for classification purposes and improves considerably its sensitivity to small differences between two similar words. Therefore, it becomes more effective to avoid the generation of “false alarms”. A network structure as shown in Figure 2 has been used for each word to be recognized.

The network is totally interconnected and a backpropagation training algorithm is used. The training procedure is iterative and at each iteration a new frame is used. The network inputs consist of 4 vectors containing the 20 energy values produced by the filters for 4 consecutive frames and a pair of vectors with 20 feed-back values from the outputs of the first layer neurons in the previous training iterations. In the training procedure, the network output error results from a comparison between the network output and a value which smoothly increases at each iteration. The function which generates this value is the one proposed by Morgan and Scofield [8].
3. THE TRAJECTORY DEFINITION SUB-SYSTEM

The solution of the trajectory definition problem consists of finding a path which the robot arm grip should follow to reach a specific goal starting from a known rest position. Considering a SCARA robot arm, the trajectory is defined by the necessary rotation angles at the “elbow” and “shoulder” joints for the arm to reach the goal.

In fact, the robot arm scans its work area in a sequence of steps by performing a rotation of the robot arm “shoulder”. At each step, the captured image is analysed. If the desired object is not identified within it, the robot arm performs another rotation step to the next scanning position. When obstacles are detected in the captured image, the VGRAPH algorithm is used to determine a free trajectory and the controller leads the robot arm grip to the next rotational scanning position avoiding the obstacles. This position is carefully determined in order to ensure that at least once in the overall scanning process the desired object will be totally contained within the rectangular captured image. Finally, when the desired object is found, a trajectory which avoids possible obstacles is determined by the VGRAPH algorithm to take the grip to the correct position to pick the object. Then, the grip is commanded to rotate by the necessary angle and to go down in order to pick the object.

The exact x-y coordinates to be reached by the grip at each step are determined in relation to the center of the robot view area bounding rectangle. With this information, an inverse kinematics problem is solved to calculate the values of the “elbow” and “shoulder” rotation angles. As the current implementation of the controller performs an angular velocity control, the rotation angles are transformed into time-sequences of angular velocities. These time-sequences describe rectangular pulse waveforms in which the area is equal to the desired rotation angle and the velocity amplitude is proportional to the rotation angle.

Therefore, the trajectory definition problem can be divided into two distinct procedures. The first one finds the correct rotation angle to be applied to the robot arm shoulder at each step until the desired object is found within the captured image. The second procedure defines the trajectory to be followed by the robot arm grip either to go to the next search position or to pick the object if it was already found. In both cases, the trajectory is defined using the VGRAPH algorithm to avoid obstacles.

3.1 The Search for the Object Procedure

The determination of the incremental rotation angle that should be applied to the robot arm shoulder at each step of the scanning process is dependent on the existence of objects that are cut by the top or by the left border of the image rectangle. The maximum angle corresponds to a situation in which no object is cut. The angle is such that the extension of the bottom edge of the image rectangle after the rotation goes pass the right top corner of the rectangle before the rotation. This limited advance of the shoulder angle ensures that no object, which might be immediately above the top edge of the image rectangle, will be cut by the bottom edge of the rectangle after the rotation.

When objects are cut either by the top or by the left border of the image rectangle, the bottom right corner of each cut object bounding box is considered. For each object an angle is found. The overall minimum value is taken as the final rotation angle. If a rotation of the shoulder takes the robot arm grip to be coincident with an obstacle, the angle is further reduced until the grip bounding box does not intersect the obstacle.

The coordinates of all the obstacles detected during the scanning process are stored and recovered at each new move of the robot arm. The recovered coordinates are transformed in relation to the new x-y axis of the image rectangle. Therefore, the system always knows the position of all the previously detected obstacles when it is deciding the new trajectory.

3.2 The Trajectory Definition Procedure

Given the final position to be reached, the trajectory definition sub-system aims at finding the minimum-length path to be followed by the grip, avoiding obstacles that may be present in the environment. The following hypothesis are considered:

- the obstacles are convex polyhedras having all their faces either parallel or orthogonal to the plane where the objects to be picked are placed;
- the height of the obstacles is big enough to block the movement of the grip;
Under these hypothesis, a 2-dimensional trajectory definition problem can be considered: given a start position \( S \), a final goal position \( G \) and a set of obstacles \( O \), find the shortest path for a circular object \( A \) to move between \( S \) and \( G \), avoiding obstacles. Two algorithms, \( \text{VGRAPH} \) and \( \text{PFIELD} \), have been used to solve this problem. With the \( \text{PFIELD} \) algorithm, a velocity vector field is generated over the robot work area. This field is analogous to the water velocity field. The flow is incompressible and the velocity field is irrotational. Therefore, the field is the gradient of a Harmonic scalar function called potential, which satisfies the Laplacian equation. The potential function is defined to have high values near the obstacles and a global minimum in \( G \). Thus, the approach to be followed consists of moving the robot in the inverse direction of the potential function gradient aiming at reaching its global minimum.

The \( \text{VGRAPH} \) algorithm [3] searches for the minimum-length path between specific vertices of a weighted graph, called \( \text{VGRAPH} - \text{Visibility Graph} \), which is generated from the obstacle configuration and from the starting and final robot positions.

Let \( V \) be the set of vertices belonging to all the polygons in \( O \). Let us define \( \text{VGRAPH} \) as the weighted and undirected graph \( \text{V}(N,L) \), where \( N=\{V,\{S,G\}\} \) and \( L=\{(n_1,n_2), \forall n_1,n_2 \in N \mid \text{the line segment joining the vertices } n_1 \text{ and } n_2 \text{ does not intersect any polygon}\).

The weight associated with every graph edge is given by the distance between the vertices it connects. Therefore, the edges indicate the visibility conditions and the distances between vertices in the graph. The minimum-length path between vertices \( S \) and \( G \) is the solution to the trajectory definition problem. This solution, however, assumes that the robot is a point object. To consider a circular robot with radius \( r_A \), each obstacle has to be expanded by a distance \( r_A \). The \( \text{VGRAPH} \) is generated from the vertices of these expanded polygons and the robot may be considered as a point object placed on \( S \).

The proposed technique, however, has two major drawbacks. Firstly, it can only cope with circular robots. Secondly, when the original obstacle polygons have very sharp angles, the expansion operation produces unnecessary losses of free areas. This problem can be minimized if a sharp polygon vertex is cut before expansion in such a way that the new polygon side is perpendicular to the line which bisects the vertex angle and is distant \( r_A \) from the original vertex. Then, the remaining small losses will only matter in very complex configurations, crowded with obstacles having distances between them comparable to the robot radius.

4. EXPERIMENTAL RESULTS

4.1 Word Recognition

The neural networks have been implemented on a TMS 32C030 DSP PC accelerator board running at 33 MHz. They have been trained to recognize 10 Portuguese words. The training procedure has been performed with 30 speakers chosen among those the network was unable at first to recognize at least one of the spoken words. Each word was repeated twice by each speaker.

The training of the backpropagation network took around 26 hours with a training step equal to 0.03. The network convergence was assumed to be achieved when the error for all outputs and all word samples was less than 0.05. The network has been around 100% successful in recognizing trained words spoken by the speakers used during the training phase. In several public technology exhibitions in Brazil, over 1000 speakers, among children and adults with different accents, have tested the system and the average success rate achieved for each word is in most cases above 85% as shown by Table 1.

<table>
<thead>
<tr>
<th>word</th>
<th>rate</th>
<th>word</th>
<th>rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>chave</td>
<td>84%</td>
<td>phillips</td>
<td>80%</td>
</tr>
<tr>
<td>alicate</td>
<td>90%</td>
<td>sai</td>
<td>88%</td>
</tr>
<tr>
<td>martelo</td>
<td>98%</td>
<td>boca</td>
<td>92%</td>
</tr>
<tr>
<td>lima</td>
<td>76%</td>
<td>bico</td>
<td>90%</td>
</tr>
<tr>
<td>allen</td>
<td>84%</td>
<td>pinça</td>
<td>86%</td>
</tr>
</tbody>
</table>

Table 1: Backpropagation Recognition Rate

The backpropagation network has often generated false alarms. However, when properly trained, it has shown to be able to distinguish between similar words (minimal pairs) very effectively.

The recursive network has achieved a similar rate of success for trained words as the backpropagation network. The training time, however, is much longer. The advantage of the recursive network lies on its ability to distinguish a non-trained word from a similar trained one. As the training procedure is performed along the evolution in time of the spoken word, the network is able to “see” inside the word and to display a low value in its output as soon as a syllable which is different from the trained one is detected in the spoken word. Therefore, it is useful to avoid the generation of false alarms. Figure 3 illustrates the behavior of a three-output recursive network when a non-trained word, “martirio”, and a trained word, “martelo”, with identical initial syllable are spoken. When the word “martirio” is spoken, the network third output becomes positive for the frames corresponding to the initial syllable and negative for the final ones.
4.2 Obstacle Avoidance

4.2.1 VGRAPH Operation

Figures 4 to 8 show the VGRAPH algorithm assisting the trajectory definition sub-system in the search for a hammer in the robot arm work area. The obstacles are represented by circular dark shapes.

In the first camera snapshot shown in Figure 4, the hammer is not found. Therefore, the robot arm moves to the next scanning position in the search procedure, where two obstacles are found, but not the hammer. With the use of the VGRAPH algorithm, the robot arm moves again to the next scanning position after storing the information on the positions of the two obstacles. Figure 5 shows the picture taken camera snapshot at this new position where three new obstacles are present. As the hammer could not be found because it is only partially shown, the robot arm moves to the next scanning position using the VGRAPH algorithm.

![Figure 4: First Snapshot](image)

![Figure 5: Second Snapshot](image)

Figure 6 shows the algorithm in operation taking into consideration the information stored on the coordinates of previously detected obstacles. The original grip position is labeled with an S and the circle is the grip desired final position. The expanded polygons correspond to the five obstacles. The visibility graph is shown with the shortest path represented by the highlighted edge.

![Figure 6: VGRAPH Operation](image)

The captured image at the next robot arm position is shown in Figure 7, where the hammer can finally be found. The optimum trajectory to reach the hammer is determined by the VGRAPH algorithm taking into consideration previously found obstacles. This trajectory is shown in Figure 8, where four intermediate positions are represented by dark circles and the final position by a light circle. The vertices marked with small circles are those which cannot be reached by the robot arm either because a collision would happen or because the arm cannot physically reach them. Therefore, the path on the right of the obstacles, which seemed to be the problem solution, cannot be considered.

![Figure 7: Third Snapshot](image)

![Figure 8: VGRAPH: Path to the Hammer](image)

4.2.2 VGRAPH x PFIELD

The analysis of the potential fields generated by the PFIELD algorithm when its control parameters change is performed through the observation of the movement of several particles initially placed along a regular grid in an environment where an obstacle is present. Figure 9 shows the trajectory followed by the robot from the source S to the goal G with the use of the PFIELD algorithm after a careful tuning of its control parameters. The obstacle edges had to be divided into 58 panels, the strength of the uniform flow (U) used in the definition of the potential field has been set to 1.5, the attraction coefficient of the goal \( \lambda_g \) has been set to 1000 and the normal velocity to the panels (V) has been set to 0.5.

The processing of the PFIELD algorithm consists of two phases. The first one solves an \( m \times m \) linear system to evaluate the panels strengths. In the second phase, a velocity vector is calculated iteractively as the particle moves within the environment. If a big step length for the particle movement is allowed, the quality of the solution gets worse. However if a small step length is used, the execution time of the second phase increases considerably.

For an implementation on a 90 Mhz Pentium microcomputer, Table 2 shows the achieved path length and the running time of the PFIELD algorithm considering different numbers of panels and values for the step length. The remaining parameters have been kept constant as \( U = 1.5, \lambda_g = 1000, V = 0.5 \). Only with 58 panels a valid trajectory to the goal is found. Even then, if a step length equal to 30 is used, the robot is carried away by the uniform flow. When a shorter step length is used, the overall path length is shorter but the algorithm running time gets much bigger in the second phase.

With the use of the VGRAPH algorithm to solve the same problem, the path length is 498.5 mm and the total running time is 10.28 ms.
Figure 9: PFIELD: U=1.5; λ_g=1000; V_i = 0.5; 58 panels

<table>
<thead>
<tr>
<th># of panels</th>
<th>Step length (mm)</th>
<th>Path length (mm)</th>
<th>1st phase (ms)</th>
<th>2nd phase (ms)</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>1</td>
<td>∞</td>
<td>1.98</td>
<td>∞</td>
<td>Follows the Flow</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>∞</td>
<td>1.98</td>
<td>∞</td>
<td>Collision with obstacles</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>∞</td>
<td>1.98</td>
<td>∞</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>300.0</td>
<td>9.84</td>
<td>198.3</td>
<td>Goal is reached</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>300.0</td>
<td>9.84</td>
<td>38.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>300.0</td>
<td>9.84</td>
<td>9.89</td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>1</td>
<td>619.3</td>
<td>160.2</td>
<td>1304.3</td>
<td>Follows the flow</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>739.0</td>
<td>160.2</td>
<td>309.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>1270.4</td>
<td>160.2</td>
<td>133.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>∞</td>
<td>160.2</td>
<td>∞</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: PFIELD Performance

With 8 or 17 panels, PFIELD is faster than VGRAPH but is unable to find a path to the goal. When it succeeds, the best running time achieved is nearly 30 times bigger (58 panels and step length equal to 20 mm) and the path length produced is about 2.5 times longer. The best path found by PFIELD is still worse than the one produced by VGRAPH and, in this case (58 panels and step length equal to 1 mm), the running time is about 142 times bigger. In addition, the PFIELD algorithm is very sensitive to its control parameters depending on the problem configuration. Small changes in some parameters produced completely different behaviors in the algorithm.

5. CONCLUSIONS

CONTROLAB integrates complex control algorithms with intelligent systems aiming at applications in robotics. This paper discussed and presented practical results on the use of neural networks for word recognition and on the use of the VGRAPH or PFIELD algorithms for trajectory definition of a robot arm in the presence of obstacles.

Experimental results have shown that the proposed backpropagation network architecture can perform speaker independent word recognition even in uncontrolled and noisy environments. It is able to distinguish minimal pairs, but frequent false alarms are generated. Practical experiments with a recursive neural network with two feed-back vectors and four input vectors have shown that it is able to adequately report as “unknown” previously non-trained spoken words, overcoming the false alarm problem. The recursive network architecture drawback is its longer training time.

For the trajectory definition in the presence of obstacles, the VGRAPH algorithm is simpler to use and able to produce good quality results faster than the PFIELD algorithm. Its major drawback is the need for the robot arm to be considered as having a circular shape. The PFIELD algorithm behavior is very sensitive to the definition of its control parameters. A correct choice of the number of panels, step length, uniform flow strength, goal attraction strength and velocity values is essential for the algorithm to produce useful results.

The evolution of the CONTROLAB environment aims at having the integrated utilization of the current intelligent systems in the control of an autonomous vehicle which will understand spoken commands and will be able to move within a known environment in the presence of unknown obstacles.

6. ACKNOWLEDGEMENTS

The authors would like to thank CNPq and FINEP, Brazil, for the support given to this research work.

7. REFERENCES