Using Neural Networks for Pattern Recognition in Robotic Tasks

I. Lopez-Juarez, J. Corona

CIATEQ A. C., Centro de Tecnología Avanzada,
Mechatronics and Intelligent Manufacturing
Systems Research Group (MIMSRG)
Manantiales 23A, Fracc. Ind. B.Q.,
El Marqués, Querétaro. CP 76246. México.
(ilopez, jcorona@ciateq.mx)

Abstract

Assembly tasks by robots have traditionally depended on simple sensing systems and the robot manufacturers programming language. However, this restricts the use of robots in complex manufacturing operations. An alternative to robot programming is the creation of self-adaptive robots based on Artificial Neural Networks (ANN's) in order to use industrial manipulators in unstructured working environments.

The research presented in this article shows how force sensing data can be used by an ANN to map contact force states with robot's motion. The methodology has been succesfully tested during assembly tasks showing that an industrial robot can learn different operations. The method is generic and it has been tested using two industrial manipulators.

1 Introduction

Industrial robots are reliable machines. The success of the operation is based on the accuracy of the robot itself and the precise knowledge of the environment, i.e. information about the geometry of the parts to be assembled, located, machined, welded, etc. Combining these elements and using the manufacturers programming language efficient programs can be written. When parameters change, the robot program has to be amended to take into account new conditions. The adaptation to these new conditions is explicitly given by the programmer. Industrial robots are currently being programmed using this technique, hence, robots are still unable to be self-adaptive to varying conditions and this is possibly one of the major drawbacks that has limited their extensive use in manufacturing.

Techniques are sought to provide self-adaptation for robots. Robot manipulators operate in real world situations with a high degree of uncertainty and require sensing systems to compensate from potential errors during operations. Uncertainties come from a wide variety of sources such as robot positioning errors, gear backlash, arm deflection, ageing of mechanisms and disturbances. Controlling all the above aspects would certainly be a very difficult task, therefore a simpler approach based on force control is preferred. By using force control the overall effect of the contact force between the environment and the manipulator are considered as a whole. Robots can also recognise force/torque patterns and operate in unstructured environments specifically performing assembly tasks.

Related work using ANN is reviewed first. The system architecture is described and the nature of the force patterns explained. Based on this background the ANN Adaptive Resonance Theory (ART) is presented continuing with the description of the Neural Network Controller (NNC). Experimental results showing the NNC's performance are discussed and finally some conclusions given.

2 Connectionist Models and Related Work

Force control can be roughly divided in Model-based and Connectionist-based approaches. The model-based approach takes as much information of the system and environment as possible. This information includes localisation of the parts, geometry of the parts, materials, friction, etc. The connectionist-based approach is based on connectionist models and its robustness relies on the information given during the training stage that implicitly considers all of the above parameters. Model-based methods do not offer a complete solution due to the uncertainties associated during assembly as mentioned earlier. On the other hand, connectionist-based techniques have proved to work reliably when uncertainty is involved due to their generalisation property.

The use of connectionist models in robot control to solve the problem of assembling parts under uncertainty has been demonstrated in a number of publications, either in simulations [1], [4], [5], or being implemented on real robots [6], [7], [8]. In these methods, Reinforcement Learning (RL), unsupervised and supervised type networks have been used. The reinforcement algorithm implemented by V. Gullapalli demonstrated to be able to learn circular and square peg insertions. The network showed a good performance after 150 trials with insertion times lower than 100 time steps [9]. Although the learning capability demonstrated during experiments improved over time the network is unable to generalise over different geometries. Insertion are reported with both circular and square geometries, however, when inserting the square peg, its rotation around the vertical axis was restricted otherwise the insertion would not have been possible. M. Howarth followed a similar approach, using Backpropagation in combination with reinforcement learning. During simulation it was demonstrated that 300 learning cycles were needed to achieve a minimum error level with his best network topology during circular insertions [8]. A cycle meant to be an actual motion that diminished the forces acting on the peg. For the square peg, the number of cycles increased dramatically to 3750 cycles. These figures are important, especially when fast learning is desired during assembly. On the other hand, E. Cervera using SOM networks and a Zebra robot (same used by Gullapalli) developed similar insertions as the experiments developed by Gullapalli. Cervera in comparison with Gullapalli improved the autonomy of the system by obviating the knowledge of the part location and used only relative motions. However, the trade-off with this approach was the increment of the number of trials to achieve the insertion [6], the best insertions were achieved after 1000 trials. During Cervera's experiments the network considered 75 contact states and only 8 out of 12 possible motion directions were allowed. For square peg insertions, there were needed 4000 trials to reach 66% success of insertion and that did not improved any further. According to Cervera's statement, " We suspect that the architecture is suitable, but the system lacks the necessary information for solving the task". The situation clearly recognises the necessity to embed new information in the control system as it is needed, which is likely to be achieved with an architecture such as ART.

3 The Research

In our research, the robot is provided only with contact force information and a Primitive Knowledge Base (PKB), which is an initial contact force-action mapping that bias its initial reactions to constrained forces. No information is given about the localisation of the parts. The arm increases its knowledge on-line based on the success of the predicted motion. The robot actually increases and enhances its knowledge as the operation progresses. The time that the robot takes to complete a similar operation is reduced and also mistakes made earlier do not recur, which demonstrates the new expertise of the robot.

The design of the novel Neural Network Controller (NNC) is founded on the strength of ART networks to learn incrementally. The new information is acquired as the operation develops without affecting the knowledge that was previously learnt. The Fuzzy ARTMAP algorithm is used and the NNC training made on-line. The number of contact force patterns that the NNC can accommodate in its knowledge is limited only to memory storage. The switching mechanism of the NNC is regulated by the development of the operation. New knowledge information is only accepted in the Knowledge Base when it has strongly contributed towards the success of the assembly. The resulting Enhanced Knowledge Base (EKB) at the end of the assembly can be used for similar operations. Results on industrial robots demonstrate that the robot's skill improves effectively and the insertion times and the errors diminish over time. Furthermore this is, to the best knowledge of the authors, the first time the Fuzzy ARTMAP network has been applied to an industrial robot manipulator.

4 System Architecture

The hardware architecture is formed by an industrial master computer in whi the DSP based F/T sensor card resides, the industrial manipulator KUK KR15, KRC2 controller, Kuka Control Panel (KCP) and JR3 F/T sensor illustrated in Figure 1. The main units of the robot system are the KRC controller and the robot arm itself. Power and data are transmitted betwee the two units through two interconnecting cables. The KRC2 controller hous the components that control and power the robot arm. The master comput communicates with the controller via serial port using the Xon/Xoff protocol. Additionally, the robot is also provided with vision and speech recognition systems, which enable the robot to recognise simple 2D geometries and to commanded via voice. For further details on these other systems the reader referred to [3]

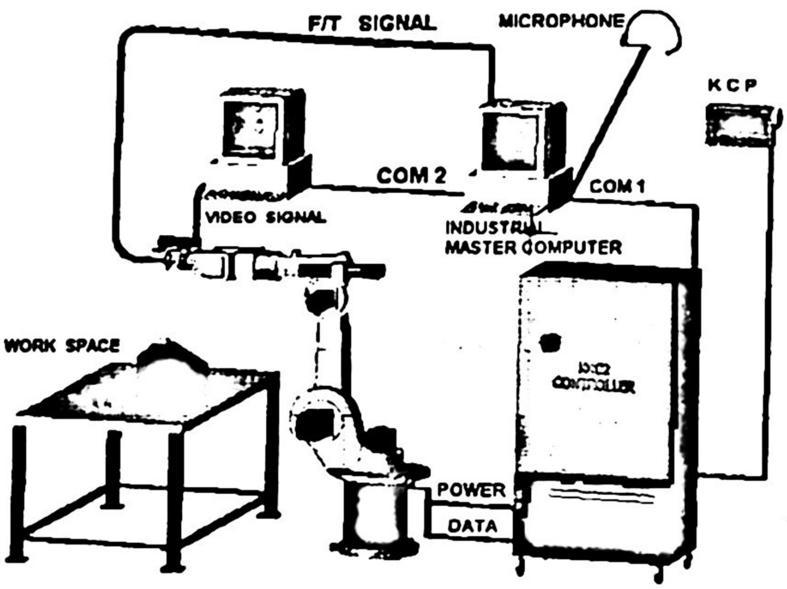


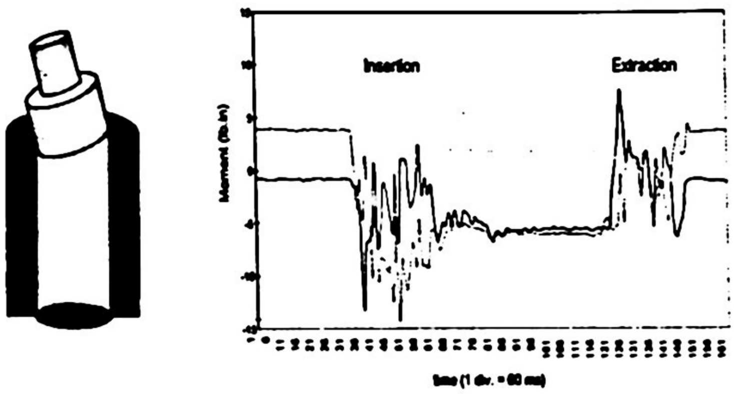
Figure 1: System Architecture

5 The problem and nature of forces

Figure 2(a) shows a typical peg in hole insertion, which is a canonical operation for performance assessment. The force traces occurring during this type operation are given in Figure 2(b).

This type of signal is normally acquired by using a F/T sensor mounted in robot's wrist. The sensor provides the required input information to the NNC. The signal patterns contain information regarding the force and torque "felt" at the robot wrist. With this information is possible to determine how much force is being applied to the end-effector or gripper[1] and how these forces affect

[1]The gripper is a mechanical device to grasp and hold the assembly part. which normally consist of two or more fingers.



(a) (b)

Figure 2: (a) Peg-in-hole insertion (b)Contact force

the orientation of the peg by means of the moment value and sign. Although the graph only shows the moment information, the information available to the NNC is a F/T vector containing six elements i.e. fx, fy, fz, mx, my and mz.

5.1 Nature of contact forces

Peg-in-hole operations were carried out using pegs with different cross-section: circular, square and square with a rounded corner, (this peg will henceforth be referred to as a radiused-square peg). The female parts were made with chamfer at 45°. A typical force trace of the insertion and extraction of the square peg is shown in Figure 3a. It should be noted that magnitudes were scaled to have better interpretation of the proportionality and similarity properties between force and moment signals. Signals corresponding to Z axis (insertion direction in tool coordinates) are not given since they were completely different.

It was observed that signals corresponding to peg insertions with symmetric cross-section (square and circular) followed a similar pattern, while for the radiused-square peg insertion the patterns were totally different. This occurred because the distribution of the contact forces on the pegs were also dissimilar due to the non-symmetric shape of the peg. Therefore, it can be said that the type of pattern depends on the *magnitude* of the force applied during assembly and the *shape* of the peg but, it is important to mention that this assumption is valid only when both mating pairs are aligned, that is, the peg has been aligned perpendicular to the female component. Additionaly, the type of proportionality and cross-correlation changes according to the *offset* location of the peg whitin the X-Y plane quadrants shown in Figure 3b.

The peg and the female block are shown in top view. The female component has been enlarged to make obvious the placement of the peg within quadrants. The level of cross-correlation between force and moment patterns depends on the placement of the peg within these quadrants. An important observation is that the correlation was related to the symmetry of the mating pairs. Circular and

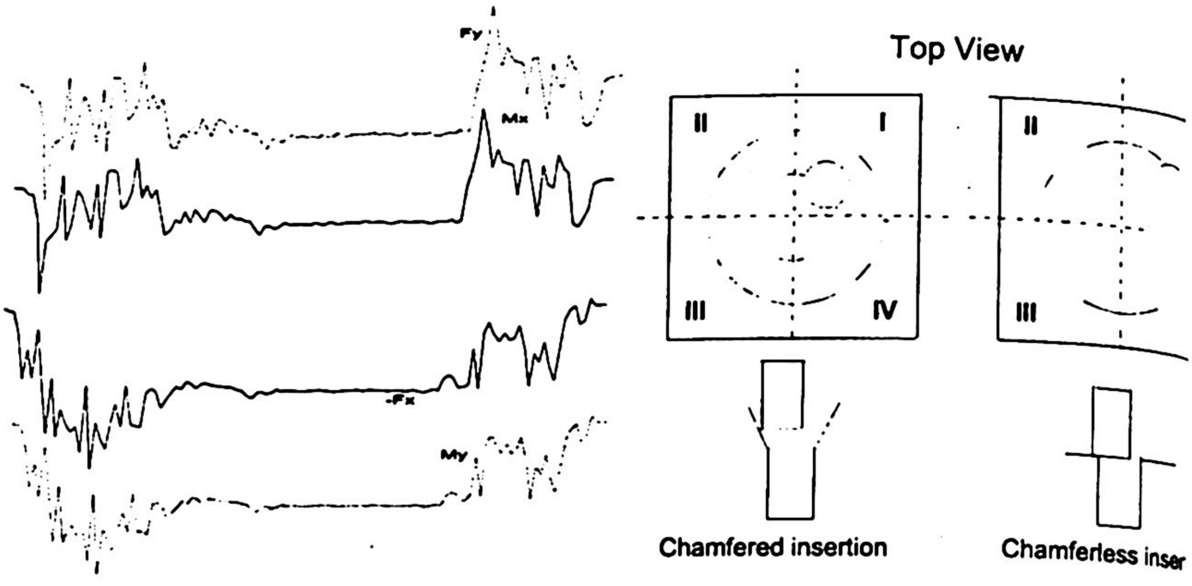


Figure 3: (a) *Insertion-extraction square peg* *(b) Symmetry and contact forces*

square pegs showed higher correlation (infinite rotational symmetry) compared to the radiused-square peg (one rotational symmetry).

The complexity of the system becomes evident, especially if the infinite number of patterns that can be generated through the assembly process considered. To deal with this complexity the proposed NNC has to classify recognise these patterns first.

6 ART and FuzzyARTMAP

The Adaptive Resonance Theory (ART) was developed by Stephen Grossberg and Gail Carpenter at Boston University. Different model variations have been developed to date based on the original ART-1 algorithm [10]. The mechanics a basic ART module are as follows: It consists of two subsystems as illustrated in Figure 4. The attentional subsystem is made up of two layers of nodes F_1 and F_2. In an ART network, information in the form of processing-element output reverberates back and forth between layers. If a stable resonance takes place learning or adaptation can occur. On the other hand, the orienting subsystem is in charge of resetting the attentional subsystem when an unfamiliar event occurs.

A *resonant state* can be attained in one of two ways. If the network has learned previously to recognise an input vector, then a resonant state will be achieved quickly when that input vector is presented. During resonance, the adaptation process will reinforce the memory of the stored pattern. If the input vector not immediately recognised, the network will rapidly search through its stored patterns looking for a match. If no match is found, the network will enter a resonant state whereupon the new pattern will be stored for the first time. Thus, the network responds quickly to previously learned data, yet remains able

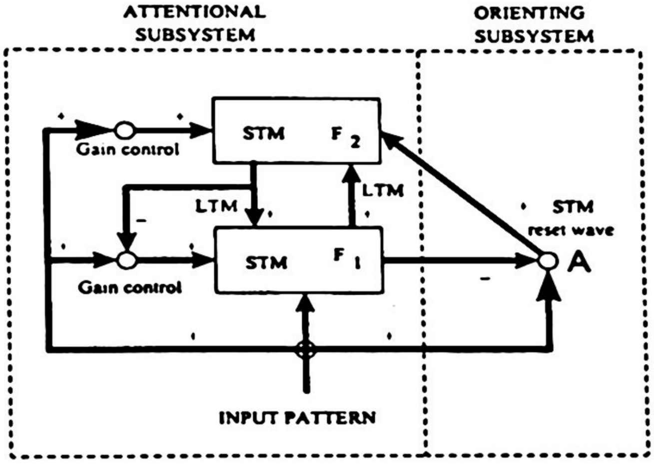


Figure 4: ART architecture

learn when novel data is presented, hence solving the so-called *stability-plasticity* dilemma. The activity of a node in the F_1 or F_2 layer is called *short-term* memory (STM) whereas the adaptive weights are called *long-term* memory (LTM). Gain controls handle the discrete presentation of the input signals. A vigilance parameter measures how much mismatch is tolerated between the input data and the stored patterns, which can be used to control the category coarseness control of the classifier.

Supervised learning is also possible through ARTMAP that uses two ART modules or its variants, such as Fuzzy ARTMAP (FAM) that incorporates fuzzy set theory operations in order to handle analogue data between 0 and 1 [11]. The NNC was designed based on this FAM network due to its capabilities of fast incremental learning (typically one epoch). The mechanics of the NNC which incorporates the FAM network is explained in the next section.

7 Neural Network Controller (NNC)

The functional structure of the assembly system is illustrated in Figure 5. The FAM is the heart of the NNC. The controller includes three additional modules. The Knowledge Base that stores the initial information related to the geometry of the assembling parts. This information is used only during the first assembly operation, later this is enhanced by patterns that favour the assembly and whose inclusion is regulated by the Pattern Selector Module. The Pattern Selector section keeps track of the F/T patterns and verifies whether the action is good enough to allow the FAM network to be retrained. If this is the case, the switch SW is closed and the corresponding pattern-action provided to the FAM for on-line retraining.

Future predictions will be based on this newly trained FAM network. The Automated Motion module basically is in charge of sending the incremental motion request to the robot controller. External components to the NNC are

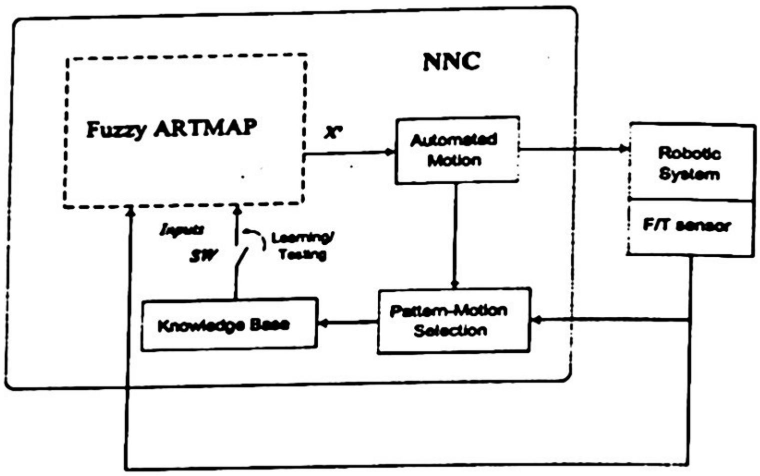


Figure 5: Neural Network Controller (NNC)

the robot controller, image system, the manipulator itself and the F/T sensor that provides the pattern information. The programs for the NNC were created using Visual C++ 6.0 and implemented in a 800MHz Pentium III Industrial Computer.

7.1 Initial training and PKB formation

The formation of the PKB basically consists of showing the robot how to act to individual components of the F/T vector. The influence of each vector component requires a motion opposite to the direction of the applied force diminish its effect. The procedure is illustrated in Figure 6. For simplicity, only the lower arm of the manipulator has been shown.

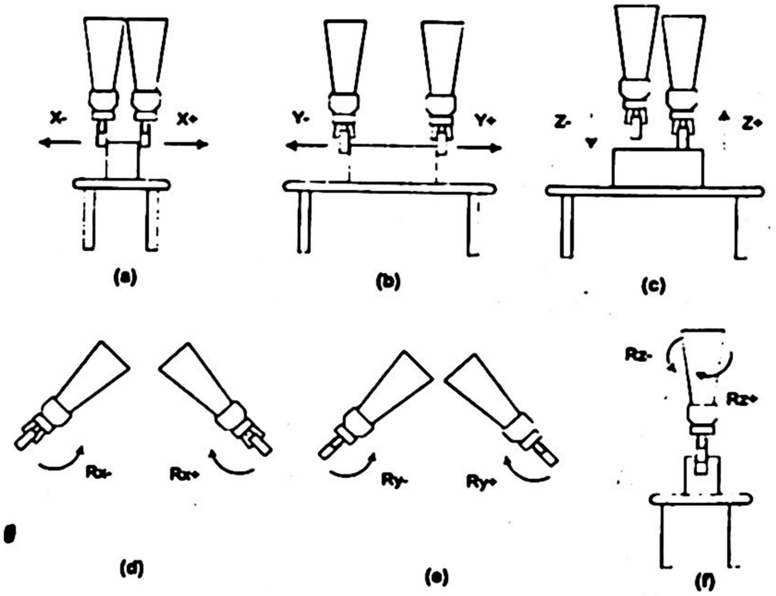


Figure 6: Training Procedure

Every motion of this type is referred to as a Primitive Motion (PM) and the idea is to teach the robot *where* to move when single F/T components, i.e. fx, fy, fz, mx, my, or mz are applied to the workpiece. Figure 6 illustrates the PM needed to diminish the corresponding constraint force in all possible motions. The storage of the F/T vector and the PM will form the PKB that is required to start the assembly for the very first time. Once the first insertion has been completed, the robot may possibly have increased its knowledge. If so, the PKB is enhanced and an Enhanced Knowledge Base (EKB) version will be used during the following insertion.

The PKB used during our experiments is shown in Figure 7, The F/T data from the sensor was scaled to the range [0,1], where the extreme values 0 and 1 corresponded to a force of -15 lb and +15 lb respectively. Negative values were assigned to the interval [0,0.5) and positive values were assigned to the interval (0.5,1]. It should be noted that the origin in the graph is set to 0.5, where positive and negative values are represented in the upper and lower halves of the graph respectively. Every column corresponded to an input vector to the network. The corresponding assigned output vector is shown at the top of the graph for each pattern.

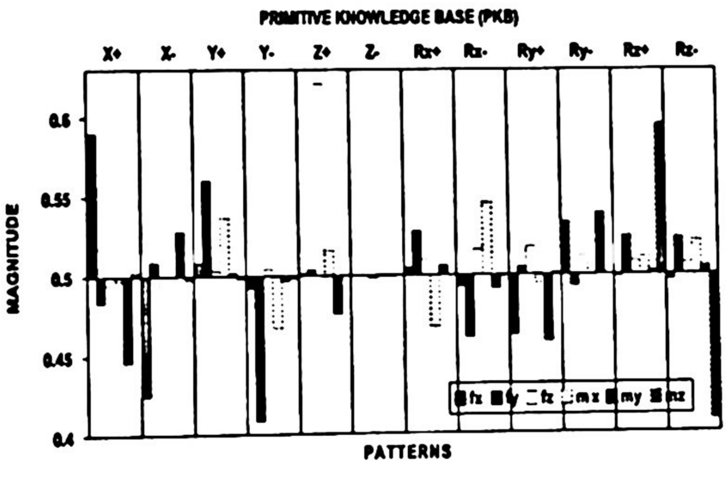


Figure 7: PKB

8 Pattern-Motion selection and knowledge enhancement

There are potential problems associated with the learning mechanism which are solved by the Pattern-Motion Selection module. The robot should continue moving in the insertion direction if, and only if, a minimum force value has been reached. This situation should trigger the learning mechanism in order to allow the acquisition and learning of the pattern-action pair that produced such a situation. In the event of continual learning after having reached this point, the performance of the NNC might decay. This situation is similar to what is known as overtraining, overfitting or overlearning in ANNs. At this point the learning should be stopped because if the robot learns other patterns under the above circumstances, eventually the minimum force value will be different leading wrong motions. The same applies to the condition when the end-effector meets a force higher than the force limit. There should not be any further learning during this situation since learning a higher force would probably damage the sensor.

The above situations can be resumed in three fundamental questions:

1. What is a good motion?
2. How to recover from errors?
3. Which motions should or should not be learned?

Having an assembly system which is solely guided by contact force states, the criterion to decide whether the motion was good enough to be learnt based on the following expression:

$$F_{after} < 0.1 * F_{before} \tag{1}$$

F_{after} and F_{before} are computed using the following equation:

$$F = \sqrt{fx^2 + fy^2 + fz^2 + mx^2 + my^2 + mz^2} \tag{2}$$

Expression 1 means that if the total force after the incremental motion significantly reduced then that pattern-action will be considered good to be included in the knowledge base. Experiments showed that if this threshold value was set higher (*i.e.* $\geq 0.3 * F_{before}$) the network became very sensitive and showed overtraining behaviour.

Forces that are higher than the value given by $0.1 * F_{before}$ and lower than the F_{limit} are still good values. However, the corresponding pattern-action pair will only be used during network recall. This situation is illustrated in Figure 8 that shows three possible situations: learning, recall and error recovery. The third area is a situation where $F \geq F_{limit}$. In this situation the user is alerted and asked to reposition the arm.

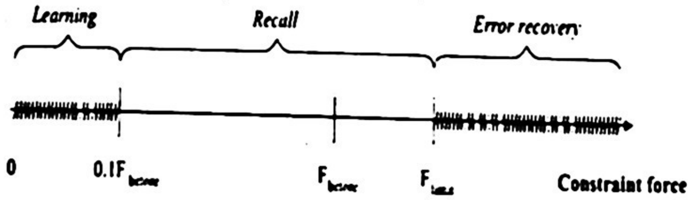


Figure 8: Learning, Recall and Error Recovery

There will be ambiguous situations in which learning should not be permitted. This applies to patterns in the insertion direction (usually Z direction). Consider downward movements in the Z- direction. At the time the peg makes contact with the female block, there may well be a motion prediction in the Z+ direction. This recovery action will certainly diminish the contact forces and will satisfy the condition given by the expression 1 in order to learn the force-action pair. However, this situation is redundant since it was given when the PKB was formed and it is likely that it will corrupt the PKB. Similarly, learning should not be allowed when the arm is in free-space. In this situation, F_{after} and F_{before} will be very similar and again learning another pattern in the Z- direction will be redundant. Both situations were tested experimentally by the author and revealed that an unstable situation may appear if further learning is allowed in the insertion direction.

After the pattern-action has satisfied expression 1 and the prediction direction is not in the Z direction, the pattern is allowed to be included in the new "expertise" of the robot, the EKB. Patterns that do not satisfy expression 1 and whose values are lower than the F_{limit} will only be used to recall previous knowledge. The knowledge refinement process will continue in the NNC until the end-condition is satisfied.

9 Results

The testbed for the assembly experiments is shown in Figure 9a and a typical peg in chamfered hole insertion is shown in Figure 9b. The Intelligent assembly was carried out using aluminium pegs with different cross-sectional geometry: circular, square and radiused-square. Clearances between pegs and mating pairs were 0.1 mm. The assembly was ended when 3/4 of the body of the peg was inside the hole.

Figure 9: (a) Testbed (b) Peg in hole assembly operation

The Fuzzy ARTMAP network parameters during experiments were set for fast learning (learning rate = 1). The base vigilance $\overline{\rho_a}$ had a low value since it has to be incremented during internal operations. ρ_{map} and ρ_b were set much higher to make the network more selective creating as many clusters as possible.

The vigilance parameters used for the experiments reported in this article are as follows: $\overline{\rho_a} = 0.2$ (base vigilance), $\rho_{map} = 0.7$ and $\rho_b = 0.9$.

9.1 Expertise test

Results from several peg-in-hole operations have shown that the robot can acquire manipulative assembly skills. At the starting of the operation the robot has only the PKB which is being enhanced though the assemblies. The robot demonstrates its expertise by reducing the number of alignment motions and consequently the insertion speed. The acquisition of the skill takes approximately 1 minute [2].

Figure 10a and Figure 10b illustrate two different learning situations in this operation. Figure 10a shows an insertion directed by the NNC and with learning enabled, this meant that the NNC was allowed to learn new patterns if the expression $F_{after} < 0.1F_{initial}$ was satisfied. After 14 insertions the robot had learned 9 new patterns, which complemented the knowledge base. From this insertion onwards there were no further patterns learned.

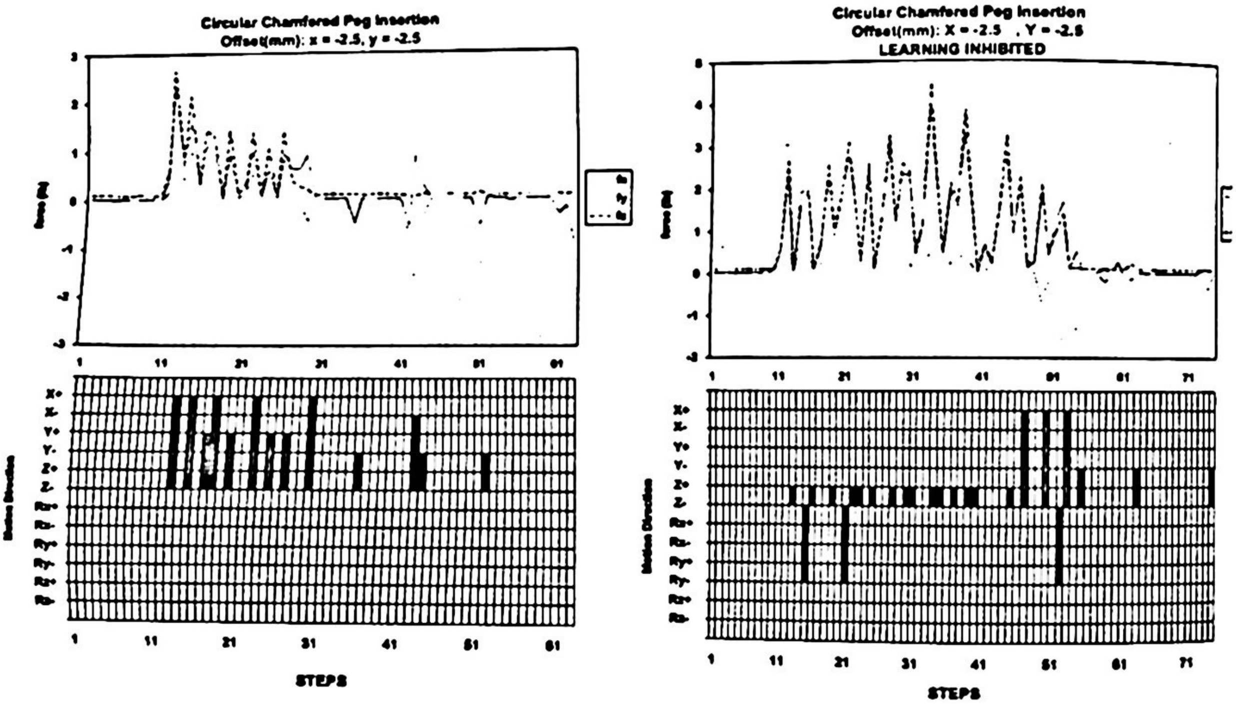


Figure 10: (a) learning enabled (b) learning inhibited

For comparison purposes, another insertion with the same offset was carried out, but in this case the robot's learning capability was inhibited (See Figure 10b). This means that the robot uses solely the PKB and no patterns are allowed to be learned during the operation. In both Figures, the upper graph represents the force traces whereas the motion directions commanded by the NNC are given in the lower graph. In the Motion Direction graph, the horizontal axis corresponds with the Z- direction. Bars above the horizontal axis represent linear alignments and below the horizontal axis represent angular alignments.

Despite that the offset was the same, the number of alignment motions and insertion time were higher. With the learning inhibited, the robot was not allowed to learn contact states within the chamfer hence the NNC generated motions based only on its initial PKB. This resulted in motions that produced an excessive fz. As a result, the NNC predicted a series of compensatory movements in Z+ and Ry- to recover from these situations. The robot was ultimately able to insert the workpiece, however the performance was poorer in terms of alignment and consequently speed.

It can clearly be observed that the same operation with the same offset can be achieved more efficiently and faster if the robot uses the EKB. In other words, the robot shows its *dexterity* when it is allowed to use its expertise.

9.2 Density of data and knowledge acquisition

The capability of generalisation and knowledge acquisition of the NNC has been demonstrated. Patterns that reduce significantly the contact forces during manipulations were acquired into the knowledge base and learnt. In the circular chamfered insertion example the network was initially trained with the PKB containing the 12 possible patterns associated with the robot's 6 DOF. This information biased the initial learning by creating 12 categories to allocate every possible motion direction. From these results, it was verified that subsequent patterns corresponding to contact states within the chamfer were effectively allocated into these categories. However, the pattern population within certain categories produced high density of data within regions in the feature space. For instance in X+ direction, which is explained below.

During the chamfered circular peg insertion only four patterns were learnt. These patterns corresponded to the X+, X-, Y+ and Y-. This can be appreciated in Figure 11a that shows the nature of learned patterns. The new patterns were valuable to speed up the insertion and to improve the insertion trajectory as it was shown during the test. However, these patterns were present within the data more than once and a total of 13 patterns were acquired after 14 insertions which implied that certain categories were more populated.

As it can be seen, patterns belonging to the same category were very similar. The patterns corresponding to the X+ direction were allowed to be learnt 8 times. This implied that the contact forces were significantly reduced in eight occasions. This high number of patterns populated more the feature space in that area, which is represented in Figure 11b.

For simplicity, only four major areas of action (X+, X-, Y+ and Y-) are represented. Initially, the main groups are formed, this is represented by the big black dots as illustrated at the beginning of the operation using what has been termed PKB. The smaller dots represent additional patterns that have been clustered within the same major region. As it is observed, the region belonging to the X+ direction was more populated than in the others. The high density of data only implies that there are more data in the region and the cost is memory space. However, since the criteria to learn new patterns was the condition given by the expression $F_{after} < 0.1 * F_{before}$, then as the learning progresses, a reduction in contact forces is expected, as it was demonstrated during the experiments, since the robot became more skillful. Being this statement true, it

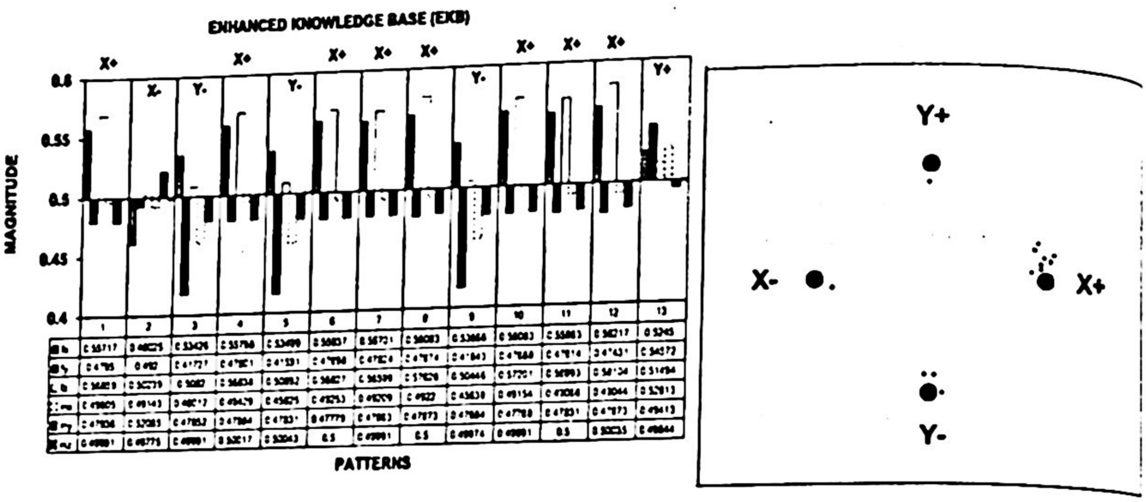


Figure 11: (a) Learned patterns during insertion (b) Data density

is also true that the knowledge acquisition becomes more strict. This obeys to the fact that forces are smaller as the robot is more skillful and from the above expression forces have also to be smaller to be accepted into the EKB. Also, as the robot's dexterity improved, the trend in the number of patterns that were accepted into the EKB decreased. The above expression for allowing the patterns to be learnt resulted to be a criterion to stop automatically the learning. With this reasoning in mind, it can be demonstrated that the density does not corrupt the selectivity of the NNC, but only affects the memory resources to allocate the learned patterns.

10 Conclusions

Results from our experiments demonstrate that industrial manipulators can learn manipulative skills on-line using only contact force pattern information. The use of the Fuzzy ARTMAP based Controller has provided the on-line learning capability needed by the task. The robot is able to learn incrementally new patterns and consequently new assembles and to improve effectively its skills from experience. The methodology is generic and has been tested in two industrial manipulators [12], [2]. On going work in looking at the implementation of an adaptive preprocessing stage which will allow the system to have the input pattern values scaled according to the current input range. The design of a Multimodal ART architecture to reinforce the confidence in the motion prediction has been envisaged. With this architecture we expect to fuse image and tactile information to aid complex operations.

Acknowledgements

The authors wish to thank The Mexican and The Queretaro's Science and Technology Councils (CONACyT, CONCyTEQ) for partially funding this project.

References

[1] I. Lopez-Juarez, M. Howarth, K. Sivayoganathan. *Robotics and Skill Acquisition*. A. Bramley; T. Mileham; G. Owen. (eds.): In Advances in Manufacturing Technology X. ISBN 1 85790 031 6. (1996) 166-170

[2] I Lopez-Juarez, M Howarth *Knowledge Acquisition and Learning in Unstructured Robotic Assembly Environments*. The International Journal of Information Sciences, Special Issue on Intelligent Learning and Control of robotics and Intelligent Machines in Unstructured Environments. To appear.

[3] I. Lopez-Juarez, M Pena, J Corona, K Ordaz, F Aliew. *Towards the Integration of Sensorial Perception in Industrial Robots* Fifth International Conference on Application of Fuzzy Systems and Soft Computing Milan, Italy. September 17-19, 2002. 218-223. ISBN 3-933609-13-5.

[4] H. Asada. *Teaching and Learning of Compliance Using Neural Nets*. IEEE Int Conf on Robotics and Automation. (1990) 1237-1244

[5] Enrique Cervera, Angel P. del Pobil. *Learning and Classification of Contact States in Robotic Assembly Tasks*. Proc of the 9th Int. Conf. IEA/AIE. (1996)725-730

[6] Enrique Cervera, Angel P. del Pobil. *Programming and Learning in Real-World Manipulation Tasks*. IEEE/RSJ Int. Conf. on Intelligent Robot and Systems. (1997) 471-476

[7] Vijaykumar Gullapalli, Judy A Franklin, Hamid Benbrahim. *Acquiring Robot Skills via Reinforcement Learning* IEEE Control Systems. (1994) 13-24

[8] Martin Howarth. *An Investigation of Task Level Programming for Robotic Assembly*. Ph.D. Thesis. The Nottingham Trent University. (1998)

[9] Vijaykumar Gullapalli. *Skillful control under uncertainty via direct reinforcement learning*. Robotics and Autonomous Systems. (1995) 237-246

[10] Gail A. Carpenter, Stephen Grossberg. Computer Vision, Graphics, and Image Processing. *A Massively Parallel Architecture for a Sel-Organizing Neural Pattern Recognition Machine*. Academic Press, Inc. (1987) 54-115

[11] Gail A. Carpenter, Stephen Grossberg, Natlya Markunzon, John H. Reynolds; David B. Rosen. *Fuzzy ARTMAP: A Neural Network Architecture for Incremental Learning of Analog Multidimensional Maps*. IEEE Transactions on Neural Networks. Vol. 3 No. 5 (1992) 698-713

[12] I. Lopez-Juarez, M. Howarth, K. Sivayoganathan. *An Adaptive Learning Approach to Control Contact Force in Assembly*.IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'98). Vol 3.(1998)1443-1448