Available online at www.sciencedirect.com

SCIENCE DIRECT®

Information Sciences 171 (2005) 377–402

INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL

www.elsevier.com/locate/ins

On the design of intelligent robotic agents for assembly ☆

I. Lopez-Juarez [a,*], J. Corona-Castuera [a], M. Peña-Cabrera [b], K. Ordaz-Hernandez [a]

[a] *CIATEQ A.C., Centro de Tecnología Avanzada, Manantiales 23A, Fracc. Ind. B.Q., El Marques, Querétaro, CP 76246, Mexico*

[b] *Departmento de Ingeniería en Sistemas Computacionales y Automatización, Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS), Universidad Nacional Autonoma de México (UNAM), Cd. Universitaria, D.F. CP 4510, Mexico*

Accepted 27 September 2004

Abstract

Robotic agents can greatly be benefited from the integration of perceptual learning in order to monitor and adapt to changing environments. To be effective in complex unstructured environments, robots have to perceive the environment and adapt accordingly. In this paper it is discussed a biology inspired approach based on the adaptive resonance theory (ART) and implemented on an KUKA KR15 industrial robot during real-world operations (e.g. assembly operations). The approach intends to embed naturally the skill learning capability during manufacturing operations (i.e., within a flexible manufacturing system).

The integration of machine vision and force sensing has been useful to demonstrate the usefulness of the cognitive architecture to acquire knowledge and to effectively use it

☆ Thanks are sincerely due to the following organisations who have made possible this research through different funding schemes: Rolls Royce & Associates, The Nottingham Trent University, DAAD, CONACyT, CONCyTEQ and CIATEQ.

* Corresponding author. Fax: +52 442 221 5243.

E-mail address: ilopez@ciateq.mx (I. Lopez-Juarez).

doi:10.1016/j.ins.2004.09.011

to improve its behaviour. Practical results are presented, showing that the robot is able to recognise a given component and to carry out the assembly. Adaptability is validated by using different component geometry during assemblies and also through skill learning which is shown by the robot's dexterity.

Keywords: Robotic agent; Robotic assembly; Neural networks; On-line learning; Adaptive resonance theory (ART); Skill acquisition; Knowledge discovery

1. Introduction

Industrial and personal robots will need more intelligence in the future. Current industrial robots rely heavily on the manufacturers programming language and are mostly used in well structured environments. Today's robot developers have not yet agreed in common robot languages and compilers for the development of high level programming, which is necessary to provide self-adapting behaviour to current robots. Robots within the industry must be carefully programmed and calibrated by experts before being used and later on, whenever the *task* or the *environment* changes. In recent years a great deal of effort have been focused toward making powerful artificial systems to facilitate this interface between human and computer, man to machine has become a topic of major research interest, which eventually facilitate the development and standardisation of common languages and/or robot interfaces.

The success of assembly operations using industrial robots is currently based on the accuracy of the robot itself and the precise knowledge of the environment, i.e., information about the geometry of the assembly parts and their localisation in the workspace. Techniques are sought to provide self-adaptation to 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 is preferred.

The main goal of the research presented in this paper, is to better understand biologically inspired models to recreate and embed learning and intelligence into self-adaptive industrial robots. The framework of the research is situated under the connectionist-based approach for object recognition and compliant motion learning during assembly operations toward the design of robotic agents for assembly.

The remainder of this paper is structured as follows. Section 2 reviews related work and states our contribution to the field of self-adaptive industrial robots for assembly. In Section 3 issues regarding knowledge acquisition and learning using the adaptive resonance theory (ART) model which inspired our work are described. In Section 4, the robotic agent that includes the cognitive architecture for object recognition and for part assembly is presented followed by the methodology in Section 5, which includes the hardware description as well as the methodology for object recognition and part assembly. Results from several insertions are given in Section 6 and some discussions about the insertions are provided in Section 7. Finally, conclusions and further work is presented in Section 8.

2. Related work

A few researchers have applied neural networks to assembly operations with manipulators and force feedback. Gullapalli [1] used backpropagation (BP) and reinforcement learning (RL) to control a Zebra robot. Its neural controller was based on the location error reduction beginning from a know location. Cervera [2] employed self-organizing map (SOM) and RL to control a Zebra robot, but the location of the destination piece was unknown. Howarth [3] utilised BP and RL to control a SCARA robot, without knowing the location of assembly. Howarth also propounded the employment of tasks level programming, using a BP-based neural controller. It was not implemented within a manipulator, but the simulation showed acceptable results [4]. Lopez-Juarez [5] implemented Fuzzy ARTMAP to control a PUMA robot, also with an unknown location. Jörg [6] presents the employment of vision systems and force feedback in the assembly of moving components.

Robotic assembly operations make extensive use of dedicated fixtures to hold and align parts before they are assembled. Most of the time fixtures are part specific and therefore they must be modified or replaced when product design changes. A new concept was introduced by Hoska in 1988 called "robotic fixtureless assembly" (RFA), which involves new technical challenges, but allows very potential solutions. Ro et al. [7] presented an approach for finding optimal kinematics postures for two robots performing RFA, their algorithm was demonstrated for two 2D robots using computer simulations. Ngyuen and Mills [8] and his research group have studied control issues involved in the RFA of flexible parts where they developed a dynamic model of the two robots and proposed a control algorithm, which does not require measurements of the part deflections. Plut and Bone [9,10] presented a grasp planning strategy for RFA which produces grasps which immobilise objects kinematically requiring minimal friction or clamping forces. The goal of RFA is to replace this fixtures with sensor-guided robots which can work within RFA workcells as is showed by Bone and Capson [11]. The development of such vision-guided robots equipped with programmable grippers might permit holding a wide range of part shapes without tool changing. This job can be achieved by using 2D computer vision in different manner so that 3D invariant object recognition can be achieved for aligning parts in assembly tasks. A novel method introduced by Peña [12] uses collections of 2D images to obtain a fast feature data—"current frame descriptor vector"—of an object by using image projections. This method produces "3D" POSE information for different predefined assembly parts. Iida et al. at DENSO are currently studying low-cost solutions in station-less assembly systems in which parts are assembled in moving conveyors without precise component's positioning [13].

2.1. Original contribution

The objective of the research presented in this paper is to create self-adapting robots able to perform mechanical assembly with a minimum of instructions and information.

The robot has two sensory inputs, a vision system and a wrist force sensing capability. The followed approach deals with the creation of a primitive knowledge base (PKB) that includes a 2D representation of the object to be grasped for assembly and an initial contact force–action mapping that bias initial robot's reactions to constrained forces. No information is given about part localisation of the manipulated component.

The robot is able to recognise the part to be assembled as well as its POSE (orientation and location). The rough location of the fixed part—female component—it is provided by the Vision System. After determining which part is to be grasped, then this information is sent to the robot controller.

The robot takes the part and roughly puts it onto the fixed component in readiness for assembly and the proper operation starts. The robot 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 robotic agent and the cognitive architecture (CA) are founded on the strength of ART networks to learn incrementally. New information is acquired as the operation develops without affecting the knowledge that was previously learnt. The Fuzzy ARTMAP algorithm is used and the CA training made on-line. The number of contact force patterns that the CA can accommodate in its knowledge is limited only to memory storage. The switching mechanism of the CA 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 in a KUKA KR15 industrial robot demonstrates that the robot's skill improves effectively and insertion times and errors diminish over time.

3. Knowledge acquisition, learning, and ART models

The first situation to address for the Robot agent is knowledge acquisition. Knowledge can either be built by *hand* or *empirically* as suggested by Towell and Shavlik [14]. *Empirical* knowledge can be thought of as giving examples on how to react to certain stimuli without any explanation. On the other hand, *hand-built* knowledge is acquired by only giving explanations but without examples. In robotic systems the approach would be to give the robot plenty of examples in the form of training sets. That is, building its knowledge empirically. The second approach would be to hardcode a rule-based system (hand-built knowledge). It is determined that a suitable strategy should include a combination of both methods. [1]

Learning in natural cognitive systems, including our own, follows a sequential process as it is demonstrated in our daily life. This learning is also stable because the learning of new things does not disrupt our previous knowledge. These premises are the core for the development of connectionist models of the human brain and are supported by Psychology, Biology and Computer Sciences. Psychological studies suggest the sequential learning of events at different stages or "storage levels" termed as sensory memory (SM), short term memory (STM) and long term memory (LTM) [15].

The adaptive resonance theory (ART) is a well established associative brain and competitive model introduced as a theory of the human cognitive processing, developed by Grossberg [16] at Boston University. Grossberg resumed the situations mentioned above in what he called the *Stability–Plasticity Dilemma* suggesting that connectionist models should be able to adaptively switch between its plastic and stable modes. That is, a system should exhibit plasticity to accommodate new information regarding unfamiliar events. But also, it should remain in a stable condition if familiar or irrelevant information is being presented. He identified the problem as due to basic properties of associative learning and lateral inhibition. An analysis of this instability, together with data of categorisation, conditioning, and attention led to the introduction of the ART model that stabilises the memory of self-organising feature maps in response to an arbitrary stream of input patterns [16].

The core principles of this theory and how short term memory (STM) and long term memory (LTM) interact during network processes of activation, associative learning and recall were published in the scientific literature back in the 60's [17]. The theory has evolved in a series of real-time architectures for unsupervised learning, the ART-1 algorithm for binary input patterns [18]. Supervised learning is also possible through ARTMAP [19] that uses

[1] Furthermore, this idea is supported by psychologic evidence that suggests that theory and examples interact closely during human learning [14].

two ART-1 modules that can be trained to learn the correspondence between input patterns and desired output classes. Different model variations have been developed to date based on the original ART-1 algorithm, ART-2, ART-2a, ART-3, Gaussian ART, EMAP, ViewNET, Fusion ARTMAP, LaminART just to mention but a few.

4. Robotic agent

As mentioned before, robotic assembly imposes a highly demanding task for standard industrial manipulators today. The robotic agent proposed in this paper intends to facilitate this task by integrating vision and force sensing capability. The robotic agent should emerge from the interaction between the Real-World and the robot manipulator itself. The agent should demonstrate its "intelligence" by using new knowledge, refine and apply it autonomously during skill learning showing the required skill during real world tasks.

The robot agent consists of two main blocks as it is shown in Fig. 1. These blocks are based on the cognitive architecture (CA) which will be described in Section 4.1. The assigned task to the agent is the assembly of mating pairs, this is known as the "peg in hole" operation. [2] The followed approach resembles a human agent carrying out the same task (i.e. determining the location and orientation of the mating pairs first. One part is to be grasped—typically the male component—and put above the fixed component in readiness for assembly.

The robotic operation is also divided in these two stages and each stage is carried out by the corresponding cognitive architecture, the cognitive architecture for object recognition (CA-Recognition) and the cognitive architecture for assembly (CA-Assembly). Each cognitive architecture uses an a priori primitive knowledge base (PKB) depending on the task nature. For instance, in the case of the CA-Recognition the PKB contains Image feature vectors and component type's information. In the case of CA-Assembly, the PKB contains contact force states and motion commands. The effective use of the PKB helps to start learning the tasks (recognition and assembly). Details on the PKB's formation are given in Sections 5.2 and 5.3.

In order to improve the robot's skills, it is necessary to acquire new knowledge on-line so those given tasks are achieved accurately and faster. This is achieved through feeding back relevant information that will ultimately enhance the knowledge and therefore improve the skill.

Finally, the CA-Recognition should predict the type of component (square, circular and radiused square) and send this information in the form of task

[2] The peg-in-hole operation is the most frequent task during assembly operations in industry and represents 20% of unit production costs [20].

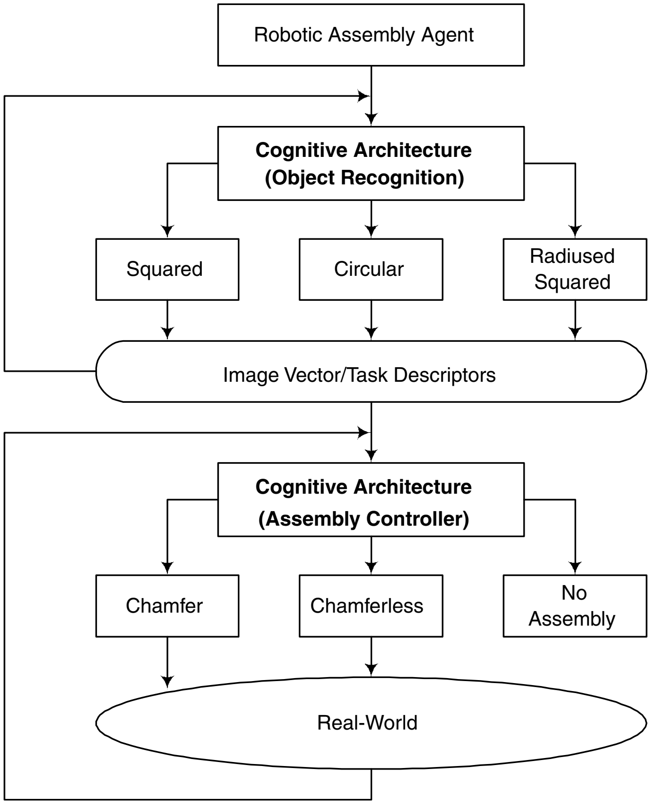


Fig. 1. Robotic agent.

descriptor to the CA-Assembly for picking up the component. The CA-Recognition produces the Image vector descriptor and task descriptor. The former is sent back for knowledge refinement and the latter is send to the CA-Assembly for motor control and to drive the manipulator to pick up the part. Also the task descriptor includes physical features of the component so that the CA-Assembly is able to select one of the available knowledge bases (chamfer, chamferless, no assembly). During assembly operations, real-world signals (force sensing) are sent back to the CA-Assembly in order to enhance the robot dexterity.

4.1. Cognitive architecture

The core of the robotic agent is the cognitive architecture (CA) which is based on ART. It is intended to design a generic architecture either for visual task (object recognition) as well as for motor control task (assembly operation). The CA is basically the development of ART as a knowledge base artificial neural network (KBANN). In KBANNs, the knowledge is inserted into the network and subsequently refined by ANN training. The idea here is to form a primitive knowledge base (PKB) by the cognitive architecture itself under real-world situations providing the robot with the capability of recognising cues or primitive descriptors during early stages of learning, so that initial conditions can be started. During knowledge refinement, and by giving more examples, this knowledge is expected to be enhanced and improved.

An overview of the designed architecture is shown in Fig. 2. The *learning and recognition* module—connectionist model—is the heart of the cognitive architecture. The architecture also includes three additional modules. The *primitive knowledge base*, the *world effector* and the *knowledge refinement* module.

The primitive knowledge base stores initial information about the environment. This information is used only during the first stage of training. In this stage the switch SW_1 will be open and the switch SW_2 closed since the initial training is made only using the PKB. After passing this initial state, the ANN will predict the next action based on the current input from the sensor (SW_1 closed and SW_2 open). Later if appropriate, the PKB will be enhanced by patterns that favoured the knowledge refinement criteria. This module keeps track of the patterns and verifies whether the change was good enough to allow the ANN to be re-trained. If this is the case, the switch SW_2 is closed and the

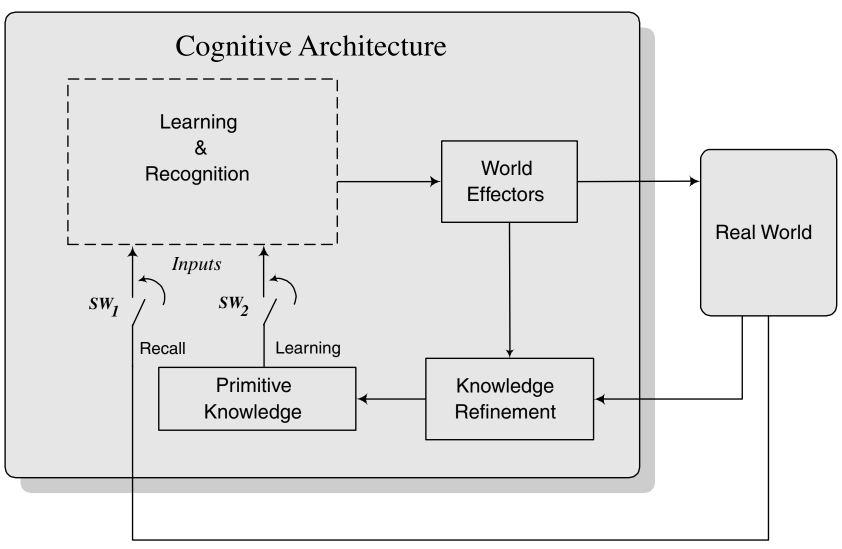


Fig. 2. Cognitive architecture (CA).

corresponding pattern-action provided to the ANN for on-line retraining. Future predictions will be based on this newly trained network. The World Effector module is basically in charge of modifying the real-world process. External components to the cognitive architecture are represented by the *Real World* block.

5. Methodology

5.1. Architecture

The robotic architecture is formed basically by a 6 DOF KUKA KR15 industrial robot, KRC2 robot controller, KUKA control panel (KCP), PC master computer, PC vision computer—not shown—JR3 F/T sensor attached to the robot's wrist, a ceiling mounted TM6710 Pulnix CCD Camera and a Conveyor Belt as it is illustrated in Fig. 3. Two working zones are defined. The Feeding Zone, which is where the manipulated components lie and the Assembly Zone which consists of a Master Block containing the fixed, female component.

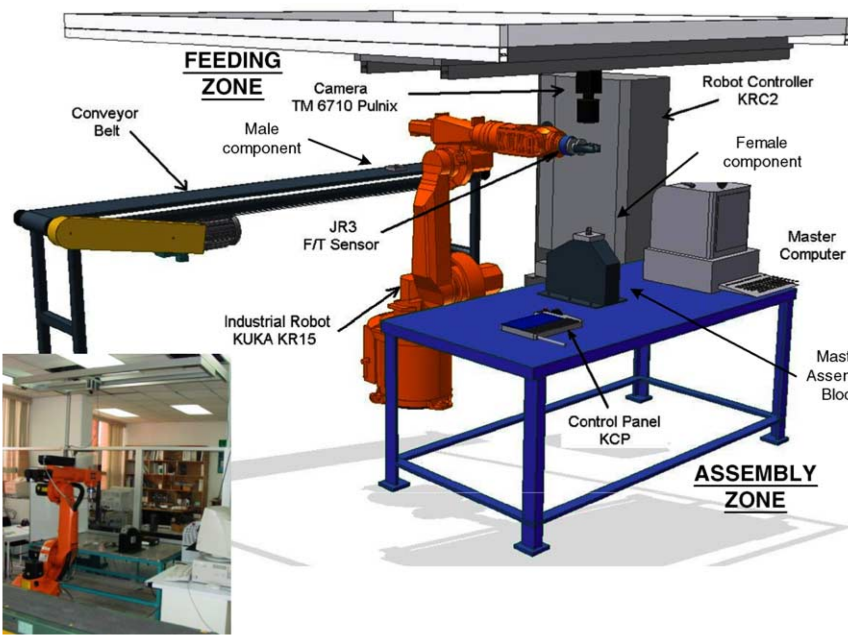


Fig. 3. Robotic system.

A schematic diagram about the hardware connections is illustrated in Fig. 4. The KUKA KR15 has a repeatability of ±0.1 mm. The KRC2 controller houses the components that control and power the robot arm. The Master Computer hosts the DSP-based JR3 F/T sensor card that communicates and power the sensor. The Master Computer also communicates with the robot controller via RS232C standard. Data sent from the Master Computer to the KRC2 controller is transmitted in the format shown below:

$\langle CODE \rangle$ NUL $\langle DIST \rangle$ NUL $\langle VEL \rangle$

where

$\langle CODE \rangle$ A byte containing the corresponding Command Code (16 motion direction commands and 9 control commands: do nothing, go to home, world coordinates, tool coordinates, joint coordinates, base coordinates, end communication, open gripper, close gripper).
NUL A byte containing the null ASCII character.
$\langle DIST \rangle$ A byte containing a distance value given in tenths of mm.
$\langle VEL \rangle$ A byte containing a velocity value given in mm/s.

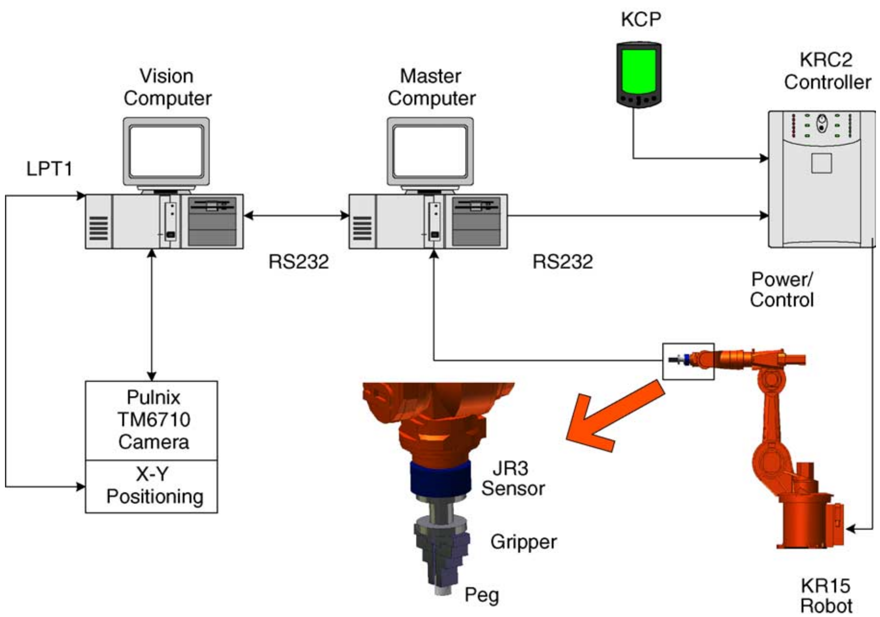


Fig. 4. Block diagram of the system.

The information packet is sent to the KRC2 controller using Xon/Xoff flow control. A monitor program in KPL—KUKA's language—is running continuously detecting any requested arm motion from the Master Computer. Depending on the value given in the CODE byte, the robot arm will move in world coordinates during gross motion (e.g. from the Feeding Zone to the Assembly Zone) or in tool coordinates at the lower level, moving the arm incrementally while in fine motion during assembly.

In practice, the F/T sampling rate by the DSP board was set to 8 kHz (0.125 ms) and the values were read from the sensor every 100 ms by the main program. However arm positions during constraint motion were updated only after a delay of 600 ms. This time was fixed and included the time for reading the sensor, testing/training the ANN on-line and repositioning the arm.

The vision system is composed by a PC Vision Computer in which a DSP-based Coreco Imaging Frame Grabber resides. This Vision Computer is in charge of positioning the ceiling-mounted camera in the $X-Y$ plane using the parallel port. The algorithms for POSE determination (orientation and location) reside in this computer. POSE information about the components on the conveyor belt in the Feeding Zone is provided serially by the Vision Computer to the Master Computer, which in turn issues position commands to the KRC2 controller for component grasping.

Contact Forces are measured at the JR3 F/T sensor which is attached to the robot's flange edge using an adapter plate and the Torque values are computed by the JR3 DSP card. The origin for the F/T coordinate frame is located in the centre of the sensor unit; however, in our experiments the origin was translated and rotated so that it was located at the peg's tip.

Programming for the CA-Recognition and CA-Assembly was developed using Visual C++ 6.0 and the monitoring program for the commanded motions to the KRC2 was developed in KPL language. Fig. 5 shows an example of the main program and two windows corresponding to the robot motion dialog. This window also includes the 3 horizontal bars that continuously monitor forces in the X, Y and Z direction. The other window dialog corresponds to neural network training parameters.

5.2. Invariant object recognition

The proposed methodology for invariant object recognition is based on the use of *canonic shapes* within the PKB. Once having embedded this knowledge the idea is to improve and refine it on-line using the CA-Recognition, which compares favourably with Gestalt principles such as grouping, proximity, similarity and simplicity [15].

To illustrate the methodology, it will be useful to consider the assembly components used during experiments. These working pieces are shown in Fig. 6(a). From this picture it can readily be recognised two canonic shapes:

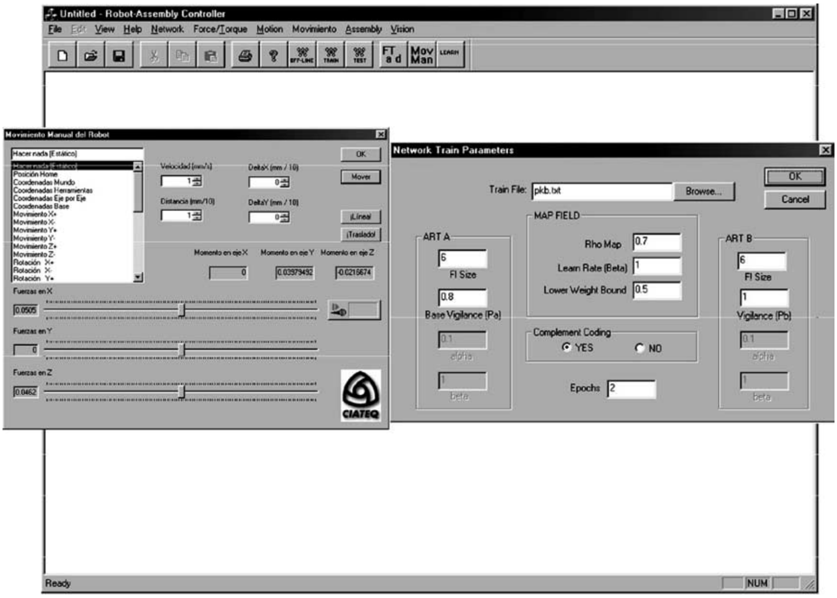
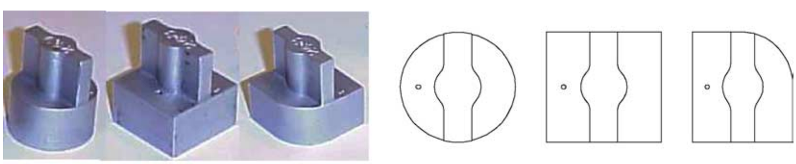


Fig. 5. Software interface.



(a) (b)

Fig. 6. (a) Working pieces and (b) canonic shapes.

circular and square. There is a third one, which is a combination of both and that we have called the ''radiused square'' shape since it is basically a square shape with one of its corners rounded, this can be well observed in Fig. 6(b).

These canonic shapes serve as ''clues'' inserted initially in the PKB which will initialise the grouping process (clustering). The knowledge acquisition is acquired by presenting multiple instances of the object such as those shown in Fig. 7. Fig. 7(a) shows an example of the circular shape and some of the possible views are illustrated in Fig. 7(b). The following step is to code the object's information so that its description be invariant to location, scaling and orientation. The algorithm is explained in the following section.

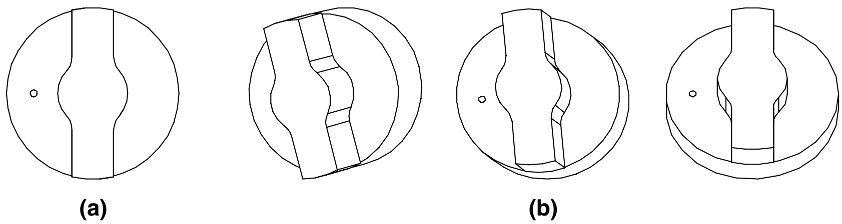


Fig. 7. Object examples.

5.2.1. Descriptor vector generation

The descriptor vector referred to as the CFD&POSE vector is formed by:

$$[\text{CFD\&POSE}] = \begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_n \\ X_c \\ Y_c \\ \phi \\ Z \\ ID \end{bmatrix}$$

where D_n is the distance from the centroid to the object's boundary, X_c, Y_c are centroid coordinates, ϕ is the component's orientation in world coordinates, [3] Z is the height in the Z axis in world coordinates. [4] ID corresponds to a codification number related to the component's type geometry.

To determine the distance from the centroid to the boundary of the object, let us first determine the boundary object function (BOF), which contains the contour and centroid of every canonical shape within the image.

The process starts by transforming in binary form the region of interest, then a Weight Transformation Matrix H_{Wf} is generated to have a relation set of

$$\text{Weight factor (Wf) (considering pixel value)} \\ \rightarrow [\text{coordinate numerical bin}] \tag{1}$$

where

$$\mathrm{Wf}_{\min} \leqslant \sum 1's \text{ within } \mathrm{Kernel}_{k\times k} \leqslant \mathrm{Wf}_{\max} \tag{2}$$

[3] The origin of the world coordinate system is located at the robot's base.

[4] Z was considered constant during experiments.

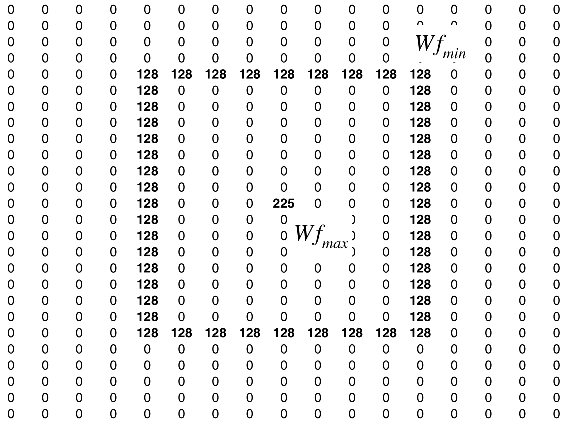


Fig. 8. Centroid and boundary points.

$\mathrm{Wf}_{\max}$ provides the centroid of an object and each $\mathrm{Wf}_{\min}$ provides a boundary point of contour (see Fig. 8). For centroid calculations a summation of all $\mathrm{Wf}_{\max}$ is made for all X–Y's as follows:

$$
\begin{aligned}
x_c &= \frac{\sum \mathrm{Wf}_{\max}(x)}{\aleph(\mathrm{Wf}_{\max}(x))} \\
y_c &= \frac{\sum \mathrm{Wf}_{\max}(y)}{\aleph(\mathrm{Wf}_{\max}(y))}
\end{aligned}
\tag{3}
$$

and the set of boundary points distances to centroid generates $\vec{X}$ and $\vec{Y}$ vectors:

$$
\begin{aligned}
\vec{X}(\mathrm{Wf}_{\min}) &= \{x_0, x_1, x_2, \ldots, x_n\} \\
\vec{Y}(\mathrm{Wf}_{\min}) &= \{y_0, y_1, y_2, \ldots, y_n\}
\end{aligned}
\tag{4}
$$

The size of the vector is up to the size of the angular grid used in CFD&POSE vector, centroid and boundary points coordinates allow calculations to get form feature extraction. Distances to get the BOF are given by

$$D_n = \sqrt{(x_n - x_c)^2 + (y_n - y_c)^2} \tag{5}$$

The orientation (ϕ), basically depends on the number of steps used in the angular grid (e.g. four steps will reference the orientation to the North, West, South or East).

5.3. Assembly

5.3.1. Robot training and the primitive knowledge base (PKB)

Once the object has been recognised, picked up and placed just above the female component then the robotic agent will initiate the assembly. This task is in charge of the CA-Assembly, the formation of the PKB in this case consists of showing the robot how to react to individual components of the F/T vector before starting any operation. While the arm is in constraint motion, the F/T pattern is acquired in the knowledge base and associated with the selected motion. The storage of the F/T vector and the Primitive Motion will form the PKB that is required to start the assembly for the very first time. The procedure is illustrated in Fig. 9.

Early work showed the usefulness of teaching the robot a PKB including 12 Primitive Motions corresponding to six F/T components and six Motion Commands associated with this F/T signal [21]. Further work resulted in the inclusion of four additional motions corresponding to four diagonal motions $(Dx^+y^+, Dx^-y^+, Dx^-y^-, Dx^+y^-)$ as illustrated in Fig. 9. A octagonal block is used as robotic tool to establish the correspondence between orthogonal and diagonal forces to motion directions.

Note that when the arm is starting an assembly and while in free-space the Primitive Motion will be in $-Z$ direction since this is the condition (minimum constraint forces) to proceed downwards during this operation. The Primitive Motion corresponding to the rotation in X and Y axis were assigned after rotating the arm in free-space at an angle so that a single mx or my component was produced. The Primitive Motion, Rz, was given to the CA-Assembly using a square peg into a square hole producing a moment around the Z axis.

The PKB used during our experiments is shown in Fig. 10, 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 -50 N and $+50$ N 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,

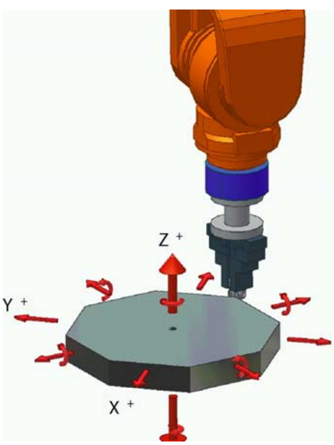


Fig. 9. Training procedure.

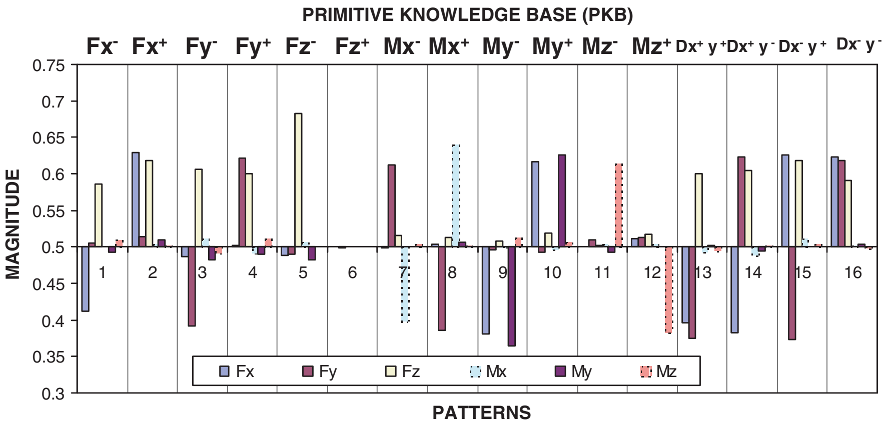


Fig. 10. Primitive knowledge base.

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 neural network. The corresponding assigned output vector is shown at the top of the graph for each pattern.

5.3.2. Knowledge enhancement

Following the CA-Assembly design, it is expected the robot's behaviour to improve after assembly operations. It is obvious that the robot have to acquire further and useful knowledge to make him skillful. This requirement clearly identifies some of the aspects to consider in order allowing new knowledge to be learned and this can be resumed in two fundamental questions:

1. What is a good motion?
2. 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 is based on the measurement increment of the F/T vector before and after the compliant motion as proposed by Ahn et al. [22]. They did not proposed a numeric value, in our experiments we used a factor of 10 between the forces measured before and after the motions, hence:

$$F_{\text{after}} < 0.1 * F_{\text{before}} \tag{6}$$

F_{after} and F_{before} are computed following the proposed heuristic equation by Ahn et al.:

$$F = \sqrt{fx^2 + fy^2 + fz^2 + mx^2 + my^2 + mz^2} \tag{7}$$

Expression (6) means that if the total force after the incremental motion is 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. $\geqslant 0.3 * F_{\text{before}}$) the network became very sensitive and showed overtraining behaviour.

Forces that are higher than the value given by $0.1 * F_{\text{before}}$ 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 Fig. 11 that shows three possible situations: learning, recall and error recovery.

The third area is a situation where $F \geqslant F_{\text{limit}}$. In this situation the user is alerted and asked to reposition the arm.

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 (6) 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 revealing that an unstable situation may appear if further learning is allowed in the insertion direction.

After the pattern-action has satisfied expression (6) 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 enhance knowledge base (EKB). Patterns that do not satisfy expression (6) 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 CA-Assembly until the end-condition is satisfied.

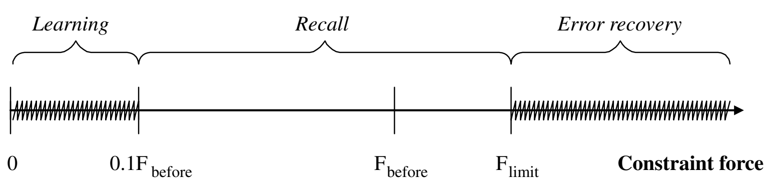


Fig. 11. Learning, recall and error recovery.

6. Results

6.1. Recognition and grasping

Several tests were carried out to assess the performance of the robotic agent using aluminium pegs. The diameter of the circular peg was 25 mm and the side of the square peg was also 25 mm. The dimensions of the radiused-square were a side 25 mm with one corner rounded to a radius of 12.5 mm. This was shown in Fig. 6(a).

A first task managed by the robotic agent is to recognise the POSE (location and orientation) of the peg in the Feeding Zone as illustrated in Fig. 3. The centroid and orientation was determined as explained in Section 5.2. An image processing picture of this task is shown in Fig. 12.

Clustering results by the CA-Recognition are given in Fig. 13.

Fig. 13 shows the image clues that were given to the Fuzzy ARTMAP algorithm in fast learning mode and the testing prototypes. Processing times are very short since the algorithm in fast PC platforms (Pentium-4 @ 2.66 GHz), performed the learning of clues and recall of prototypes in less than 2 ms, which clearly underline the real-time potential of the approach. Further details can be consulted in [12]. Results have shown that using the CFD&POSE Vector as input to the CA-Recognition is a viable methodology to recognise the three components under study. The methodology indicated that it was possible to compact 3D data in 2D data obtaining fast learning and recall. At this stage, there have been tested the clustering and learning process off-line.

In order to pick up the male component and facilitate the task, the peg's coordinate value in the Z axis was given explicitly. The robot then was moved to the specified location and grasped the part. Next, the camera was positioned in the Assembly Zone (see Fig. 3) and the centroid of the female component determined. With the current calibration of the camera, a positional resolution of ±3 mm was achieved. This positional uncertainty was chosen in order to test

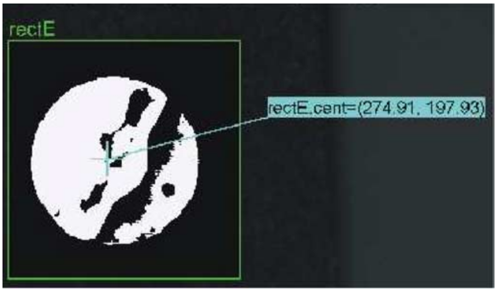


Fig. 12. Centroid determination.

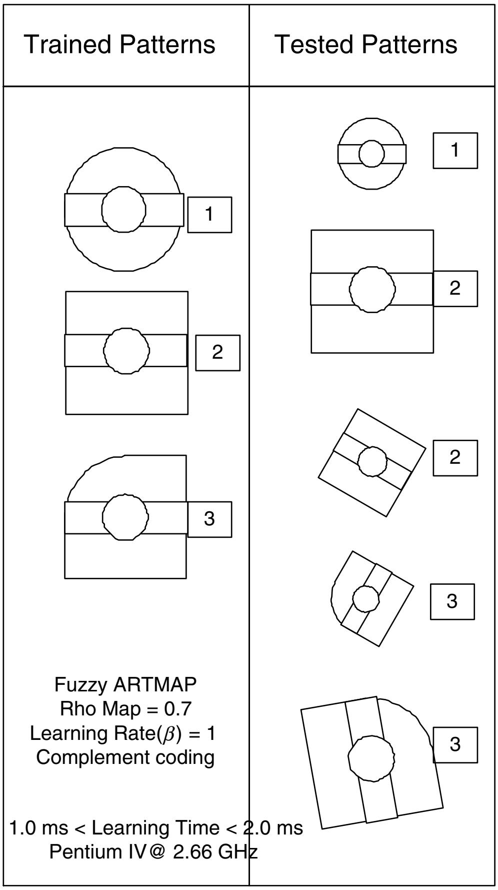


Fig. 13. Component clustering.

the robustness of the CA-Assembly, which had to compensate for this misalignment.

6.2. Assembly

During operations, clearances between pegs and mating pairs were 0.1 mm. The end-condition of the assembly was set to be 3/4 of the peg's body inside the hole. This represented 140 motion steps in the $Z-$ assembly direction without any offset. When a positional offset is given with respect to the insertion centre, this misalignment will necessarily be corrected by alignment motions in other directions different from the Z direction. A typical assembly operation it is shown in Fig. 14.

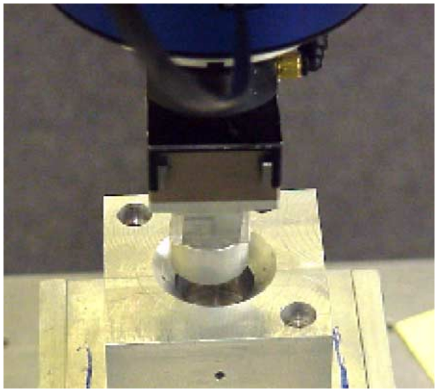


Fig. 14. Typical assembly operation.

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:

$$\overline{\rho_a} = 0.2 \text{ (base vigilance)}$$

$$\rho_{\mathrm{map}} = 0.7$$

$$\rho_b = 0.9$$

Typical results on three different geometries are summarised in Table 1:

At the start of the operation positional offsets were given as indicated in the second column. During the first insertion, learning was enabled (ON status), the network learned 3 new patterns and this operation required 155 incremental motions and only 15 alignment motions. The learned patterns were Dx^-y^- as indicated in the comments column. The processing time for the whole insertion was 1.23 min. Subsequent assemblies were carried out and the number of learned patterns decreased to none. The processing time showed only small fluctuations for insertions using the same offset. Table 1 also shows the "expertise" acquired by the robot during the operations. After six insertions there were learnt nine additional patterns. This implied that these patterns were good enough to be learned. This EKB reinforced the prediction capability of the network since the new patterns were actually generated by the particular geometry of the parts, i.e. circular. The type of learned patterns at every insertion is indicated in the comments field. With a larger offset (insertions 10–14), the

Table 1
Insertion results

Insertion	Offset ($\delta x, \delta y, \delta Rz$) (mm, mm, °)	Learning	New patterns	Alignment motions	Total motions	Processing time (min)	Comments
Circular chamfered peg insertion							
1	(1.2, 0.8, 0.0)	ON	3	15	155	1.23	$DX-Y-$
2	”	”	2	12	152	1.21	$DX-Y-$
3	”	”	3	12	152	1.21	$DX-Y-$
4	”	”	0	12	152	1.21	
5	”	”	0	12	152	1.18	
6	”	”	1	22	162	1.25	$DX-Y-$
7	”	”	0	17	157	1.21	
8	”	”	0	17	157	1.21	
9	”	”	0	17	157	1.21	
10	(2.5, 2.5, 0.0)	ON	0	26	166	1.26	
11	”	”	0	25	165	1.25	
12	”	”	0	25	165	1.25	
13	”	”	0	25	165	1.25	
14	”	”	0	25	165	1.25	
Square chamfered peg insertion							
15	(2.5, 2.5, 0.0)	ON	0	82	222	1.55	
16	”	”	0	73	213	1.51	
17	”	”	0	255	395	3.25	
Radiused-square chamfered peg insertion							
18	(−1.0, −1.0, 0.0)	”	0	259	399	3.25	
19	(−1.5, −1.0, 0.0)	”	0	148	288	2.32	$X+, Z+, DX+Y+, Y+, DX-Y-, DX-Y+$

CA-Assembly did not learned any additional pattern indicating that the network had already acquired the necessary knowledge about the chamfer and used this information effectively. As the starting point was further from the end-condition, the time to complete the insertion was proportionally longer. This is reflected in both, the number of alignment motions and the total number of motions.

6.2.1. Generalisation and expertise

Other tests were conducted to validate the generalisation feature of the CA-Assembly, which can be observed during insertion 15–17 in Table 1. At this point the circular female component was interchanged by a squared component. It was noted that the number of alignment motions increased and it took very long time to insert the peg—insertion 17—however the robot was still able to insert the squared peg using the EKB that had learned during insertion of the circular peg. Further insertions were made using the radiused-square peg with different offset—insertion 18–19—and the robot also succeeded in completing the assembly.

An example of compliant motion it is shown in Fig. 15 where acting forces and motion during insertion 14 are illustrated.

The upper and middle graph represents the force and moment traces respectively, whereas the motion directions commanded by the CA-Assembly are given in the lower graph. In the Motion Direction graph, the horizontal axis corresponds with the $Z-$ direction. Bars above the horizontal axis represents linear and diagonal alignments and below the horizontal axis angular alignments. It is observed that after 14 insertions the robot arm only moved in the $Dx^{-}y^{-}$ and Z– direction, which makes sense since the offset had been given in the $X+$ and $Y+$ direction. After passing the chamfer, the motions were only in the $Z-$ direction as it can be seen on the right side of the dotted line. Contact forces were limited to 40;N in the Z axis, if higher forces appeared then the user was required to reposition the robot arm. In practice, forces did not reach the limit if the contact was made within the chamfer or 2.5 mm from the edge of the hole in the chamferless assembly operation.

7. Discussions

7.1. Density of data and knowledge acquisition

The capability of generalisation and knowledge acquisition of the CA-Assembly has been demonstrated. Patterns that reduce significantly the contact forces during manipulations were acquired into the knowledge base and learnt. A representative learning example was shown in Section 6 with the circular chamfered insertion. In this example, the network was initially trained with

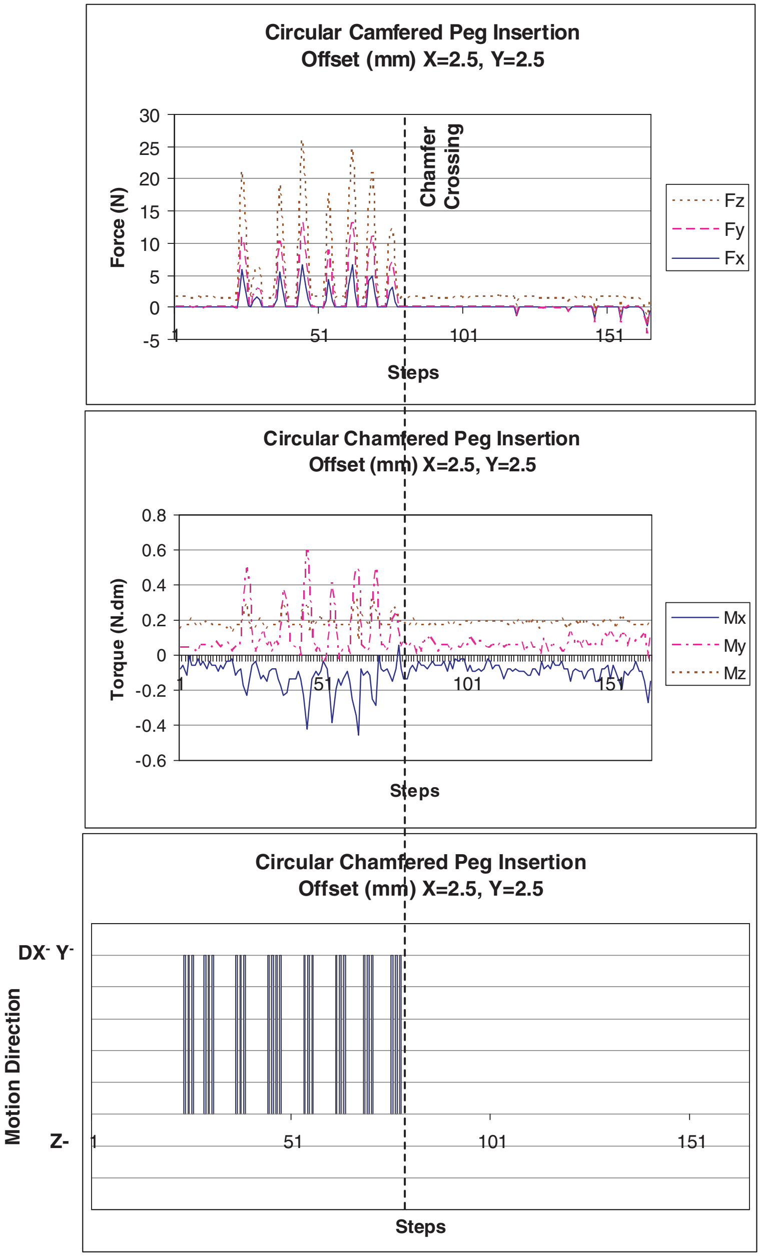


Fig. 15. Insertion example.

the PKB containing the 16 possible patterns associated with the robot's 6 DOF. This information biased the initial learning by creating 16 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.

Prior reported work used only 12 motions within the PKB [21]. It was noted that richer information in the PKB was needed; hence the robot should move in other direction apart from the main orthogonal directions. Having this in mind, diagonal motions were taught to the robot. In the given example during the chamfered peg insertion, it could be observed that this information was relevant in order to speed up the assembly (avoiding unnecessary alignment movements as it was the case in our previous work [23,21]). By having 16 primitive motions within the PKB instead of 12, the robot's behaviour was not only better but also its generalisation ability—assembling different part geometry—improved. Results showed that no additional patterns were learned despite the geometry change (see Table 1, insertions 15–17 and 18–19).

Fig. 16 shows the nature of learned patterns during the assembly of the three pegs (see the new patterns column in Table 1). According to the PKB illustrated in Fig. 10, It is easy to observe that the robot learned only diagonal motions in the $X-$ $Y-$ direction.

As established earlier, the criteria to learn new patterns was the condition given by the expression $F_{\text{after}} < 0.1 * F_{\text{before}}$. As the learning progresses, a reduction in contact forces was observed through several insertions. Forces are smaller as the robot is more skillful and from the expression above forces have also to be smaller to be accepted into the EKB. Then the expression resulted to be an effective criterion to automatically stop the learning.

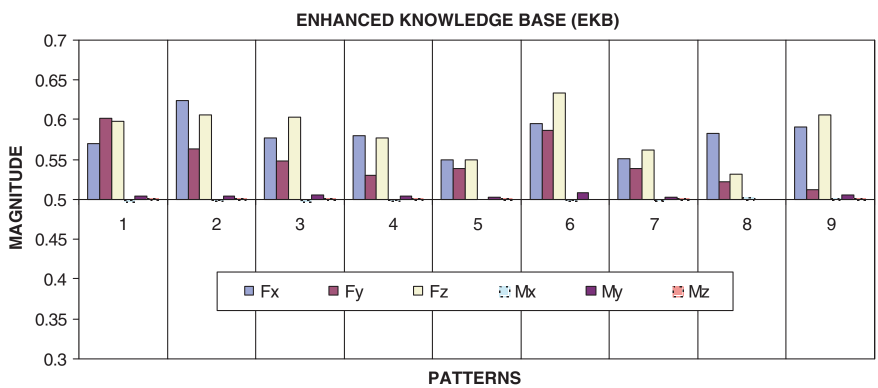


Fig. 16. Learned patterns during the circular chamfered insertion.

8. Conclusions and future work

Results from our experiments demonstrate that industrial manipulators can learn manipulative skills on-line using contact force information. The information from the environment was minimal. The knowledge was enhanced according to the part geometry and provided the required adaptation by the robot to learn a new assembly and improve its skills from experience.

Initial results employing the CA-Recognition have facilitated the assembly task. The centroid and orientation values have been very important to grasp the part. The Fuzzy ARTMAP network has been useful to classify and recognise simple parts. The novel coding of the CFD&POSE vector to invariantly represent an object and the very short convergence time makes this alternative suitable to be tested under real-time conditions. Further work is needed here, since only top views have been used. It is believed that depth information could be determined by associating the component's shadow and producing more CFD&POSE vectors that ultimately would be fed into the CA-Recognition for knowledge refinement similarly as it is made in the CA-Assembly.

Ongoing work is also looking at including complex arm motions. It is intended to include within the PKB, information regarding the stiffness of the arm, so that the arm is able to apply the required force according to the magnitude of the constrained forces. Another area of current research is the autonomous generation of the PKB, relating the part's geometry to the constraint forces and motions.

Acknowledgments

The authors wish to thank the following organisations who made possible this research through different funding schemes: Deutscher Akademischer Austausch Dienst (DAAD), Consejo Nacional de Ciencia y Tecnologia (CONACyT) and the Consejo de Ciencia y Tecnologia del Estado de Queretaro (CONCyTEQ).

References

[1] V. Gullapalli, J.A. Franklin, H. Benbrahim, Acquiring robot skills via reinforcement learning, IEEE Control Systems (February) (1994) 13–24.

[2] E. Cervera, A.P. del Pobil, Programming and learning in real-world manipulation tasks, Proceedings of 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems 1 (September) (1997) 471–476.

[3] M. Howarth, An Investigation of Task Level Programming for Robotic Assembly, PhD thesis, The Nottingham Trent University, January 1998.

[4] K. Thomas, P.D. Gentle, C.R. Howarth, M. Sivayoganathan, Task level programming using neural networks, in: Proceedings of 1995 IEE International Conference on Artificial Neural Networks, 1995, pp. 262–267.

[5] I. Lopez-Juarez, On-line learning for robotic assembly using artificial neural networks and contact force sensing. PhD thesis, Nottingham Trent University 2000.
[6] J. Stelter, C. Natale, S. Jorg, J. Langwald, G. Hirzinger, Flexible robot assembly using a multisensory approach, in: Proceedings of IEEE International Conference on Robotics and Automation, 2000, pp. 3687–3694.
[7] R. Byung, I. Ro. L. Paul, An optimum path and posture planning for fixtureless assembly, 1993.
[8] W. Ngyuen, J.K. Mills, Multi-robot control for flexible fixtureless assembly of flexible sheet metal autobody parts, in: Proceedings of IEEE International Conference on Robotics and Automation, 1996, pp. 2340–2345.
[9] W.J. Plut, G.M. Bone, Limited mobility grasps for fixtureless assembly, in: Proceedings of the 1996 IEEE International Conference on Robotics and Automation, Minneapolis, MN, 1996, pp. 1465–1470.
[10] W.J. Plut, G.M. Bone, 3-d flexible fixturing using a multi-degree of freedom gripper for robotic fixtureless assembly, in: Proceedings of the 1997 IEEE International Conference on Robotics and Automation, Albuquerque, NM, 1997, pp. 379–384.
[11] D. Capson, G.M. Bone, Vision-guided fixturless assembly of automotive components, Robotics and Computer Integrated Manufacturing (19) (2003) 79–87.
[12] M. Pena, Vision para robots en tareas de ensamble, in: Proceedings, of the Second National Congress on Mecathronics, December 2003.
[13] Y. Lida, H. Fujimoto, L.-Y. Chen, A study of automated station-less assembly system learned form human performance, DENSO Technical Review 9 (1) (2004).
[14] G.G. Towell, J.W. Shavlik, Knowledge-based artificial neural networks, Artificial Intelligence 70 (1994) 119–165.
[15] R.S. Feldman, Understanding Psychology, third ed., McGraw-Hill, Inc., UK, 1993.
[16] S. Grossberg, Adaptive pattern classification and universal receding. ii: Feedback, expectation, olfaction and illusions, Biological Cybernetics 23 (1976) 187–202.
[17] S. Grossberg, Non-linear difference-differential equations in prediction and learning theory, Proceedings of the National Academy of Science 58 (1967) 1329–1334.
[18] G.A. Carpenter, S. Grossberg, Computer Vision, Graphics, and Image Processing, Academic Press Inc., UK, 1987 (Chapter A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine pp. 54–115).
[19] G.A. Carpenter, S. Grossberg, J.H. Reynolds, Artmap: Supervised real-time learning and classification of nonstationary data by self-organizing neural network, Neural Networks (1991) 565–588.
[20] L.A. Martin-Vega, H.K. Brown, W.H. Shaw, T.J. Sanders, Industrial perspectives on research needs and opportunities in manufacturing assembly, Journal of Manufacturing Systems 14 (l) (1995) 45–58.
[21] 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 145 (1) (2002) 89–111, August.
[22] D.S. Ahn, H.S. Cho, K. Ide, F. Miyazaki, S. Arimoto, Learning task strategies in robotic assembly systems, Robotica 10 (1992) 409–418.
[23] I. Lopez-Juarez, M. Howarth, K. Sivayoganathan, Learning manipulative skills with art, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2000.