13

Distributed Architecture for Intelligent Robotic Assembly Part II: Design of the Task Planner

Jorge Corona-Castuera and Ismael Lopez-Juarez

1. Introduction

In previous chapter it has been described the overall architecture for multimodal learning in the robotic assembly domain (Lopez-Juarez & Rios-Cabrera, 2006). The acquisition of assembly skills by robots is greatly supported by the effective use of contact force sensing and objects recognition. In this chapter, we will describe the robot's ability to acquire and refine its knowledge through operations (i.e. using contact force sensing during fine motions) and how a manipulator can effectively learn the assembly skill starting from scratch.

The use of sensing to reduce uncertainty significantly extends the range of possible tasks. One source of uncertainty is that the programmer's model of the environment is incomplete. Shape, location, orientation and contact states have to be associated to movements within the robot's motion space while it is in constraint motion. Compliant motion meets external constraints by specifying how the robot's motion should be modified in response generated forces when constraints are violated. Generalizations of this principle can be used to accomplish a wide variety of tasks involving constrained motion, e.g., inserting a peg into a hole or following a weld seam under uncertainty.

The success of robotic assembly operations therefore, is based on the effective use of compliant motion, 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 within the workspace. However, in reality uncertainties due to manufacturing tolerances, positioning, sensing and control make it difficult to perform the assembly. Compliant motion can be achieved by using passive devices such as the Remote Centre Compliance (RCC) introduced by Whitney (Whitney & Nevis, 1979) or other improved versions of the device (Joo & Miyasaki, 1998). Other alternative is to use Active Compliance, which actually modifies either the position of the manipulated component as a response to constraint forces or the desired force. Some com-

Source: Manufacturing the Future, Concepts - Technologies - Visions , ISBN 3-86611-198-3, pp. 908, ARS/plV, Germany, July 2006, Edited by: Kordic, V.; Lazinica, A. & Merdan, M.

mercial devices have emerged in recent years to aid industrial applications (Erlbacher, 2004).

Active compliance can be roughly divided into fine motion planning and reactive control. Fine motion planning relies on geometrical path planning whereas reactive control on the synthesis of an accommodation matrix or mapping that transform the corresponding contact states to corrective motions. A detailed analysis of active compliance can be found in (Mason, 1983) and (De Schutter & Brussel, 1988). Perhaps, one of the most significant works in fine motion planning is the work developed by Lozano-Perez, Mason and Taylor known as the LMT approach (Lozano-Perez, et al, 1984). The LMT approach automatically synthesizes compliant motion strategies from geometric descriptions of assembly operations and explicit estimates of the errors in sensing and control. Approaches within fine motion planning can also be further divided into model-based approaches and connectionist-based approaches though, some reactive control strategies can be well accommodated within the model-based approach. In either case, a distinctive characteristic in model-based approaches is that these take as much information of the system and environment as possible. This information includes localisation of the parts, part geometry, material types, friction, errors in sensing, planning, and control, etc. On the other hand, the robustness of the connectionist-based approaches relies on the information given during the training stage that implicitly considers all the above parameters.

In this chapter we present a "Task Planner", connectionist-based approach that uses vision and force sensing for robotic assembly when assembly components geometry, location and orientation is unknown at all times. The assembly operation resembles the same operation as carried out by a blindfold human operator. The task planner is divided in four stages as suggested in (Doersam & Munoz, 1995) and (Lopez-Juarez, 2000):

Pre-configuration: From an initial configuration of the hand/arm system, the expected solutions are the required hand/arm collision-free paths in which the object can be reached. To achieve this configuration, it is necessary to recognize invariantly the components and determining their location and orientation.

Grasp: Once the hand is in the Pre-configuration stage, switching strategies between position/force controls need to be considered at the moment of contact and grasping the object. Delicate objects can be broken without a sophisticated contact strategy even the Force/Torque (F/T) sensor can be damaged.

Translation: After the object is firmly grasped, it can be translated to the assembly point. The possibility of colliding with obstacles has to be taken into account.

Assembly Operation: The assembly task requires robust and reactive positions/force control strategies. Mechanical and geometrical uncertainties make high demands on the controller.

The pre-configuration for recognition and location of components as well as the assembly operation are based on FuzzyARTMAP neural network architecture, situated under the connectionist-based approach employing reactive contact forces.
In this approach, the mapping between contact states and arm motion commands is achieved by using fuzzy rules that create autonomously an Acquired-Primitive Knowledge Base (ACQ-PKB) without human intervention. This ACQ-PKB is then further used by the Neural Network Controller (NNC) for compliance learning.

2. Related Work

The use of connectionist models in robot control to solve the problem under uncertainty has been demonstrated in a number of publications, either in simulations (Lopez-Juarez & Howarth, 1996), (Asada, 1990), (Cervera & del Pobil, 1996), or being implemented on real robots (Cervera & del Pobil, 1997), (Gullapalli, et al, 1994), (Howarth, 1998), (Cervera & del Pobil, 2002). 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 controller was a back-propagation network with 11 inputs. These are the sensed positions and forces: (X, Y, Z, θ_1, θ_2) and (F_x, F_y, F_z, m_x, m_y, m_z). The output of the network was the position commands. The performance of the operation was evaluated by a parameter r, which measured the performance of the controller. r varied between 0 to 1 and was a function of the sensed peg position and the nominal hole location. The network showed a good performance after 150 trials with insertion times lower than 100 time steps (Gullapalli, 1995). Although the learning capability demonstrated during experiments improved over time, the network was unable to generalise over different geometries. Insertions are reported with

both circular and square geometries; however, when inserting the square peg, its rotation around the vertical axis was not allowed, which facilitated the insertion. M. Howarth followed a similar approach, using also backpropagation in combination with reinforcement learning. In comparison with Gullapalli's work, where the reinforcement learning values were stochastic, Howarth's reinforcement value was based on two principles: minimization of force and moment values and continuation of movement in the assembly direction. This implied that whenever a force or moment value was above a threshold, an action (i.e., reorientation), should occur to minimize the force. Additionally, movements in the target assembly direction were favoured. 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 (Howarth, 1998). 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 (Cervera & del Pobil, 1997); 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 with any further improvement. 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.

Other interesting approaches have also been used for skill acquisition within the framework of Robot Programming by Demonstration that considers the characteristics of human generated data. Work carried out by (Kaiser & Dillman, 1996) shows that skills for assembly can be acquired through human demonstration. The training data is first pre-processed, inconsistent data pairs are removed and a smoothing algorithm is applied. Incremental learning is achieved through Radial Basis Function Networks and for the skill refinement;

the Gullapalli's Stochastic Reinforcement Value was also used. The methodology is demonstrated by the peg-in-hole operation using the circular geometry. On the other hand (Skubic & Volz, 2000 b), use a hybrid control model which provides continuous low-level force control with higher-level discrete event control. The learning of an assembly skill involves the learning the mapping of force sensor signals to Single-Ended Contact Formations (SECF), the sequences of SECFs and the transition velocity commands which move the robot from the current SECF to the next desired SECF. The first function is acquired using supervised learning. The operator demonstrates each SECF while force data is collected, and de data is used to train a state classifier. The operator then demonstrates a skill, and the classifier is used to extract the sequence of SECFs and transitions velocities which comprise the rest of the skill.

The above approaches can be divided in two groups, those providing autonomous assembly skill and those which teach the skill by demonstration. These approaches have given some inputs to our research and the work presented here is looking to improve some of their limitations. In Gullapalli's work the hole location has to be known. Howarth improved the autonomy by obviating the hole's location; however, the lengthy training process made this approach impractical. Cervera considered many contact states, which worked well also during the assembly of different type of components. In the case of teaching the skill by demonstration, the method showed by Kaiser and Dillman was lengthy for real-world problems and the work by Skubic and Volz assumes that during supervised training the operator must know which SECF classes to include in the set.
The integration of vision systems to facilitate the assembly operations in uncalibrated workspaces is well illustrated in (Jörg, et al, 2000) and (Baeten, et al, 2003) using eye-in-hand vision for different robotic tasks.

3. Workplace Description

The manufacturing cell used for experimentation is integrated by a KUKA KR15/2 industrial robot. It also comprises a visual servo system with a ceiling mounted camera as shown in figure 1. The robot grasps the male component from a conveyor belt and performs the assembly task in a working table where the female component is located. The vision system gets an image to calculate the object's pose estimation and sends the information to the robot from two predefined zones:

Zone 1 which is located on the conveyor belt. The vision system searches for the male component and determines the pose information needed by the robot.
Zone 2 is located on the working table. Once the vision system locates the female component, it sends the information to the NNC.

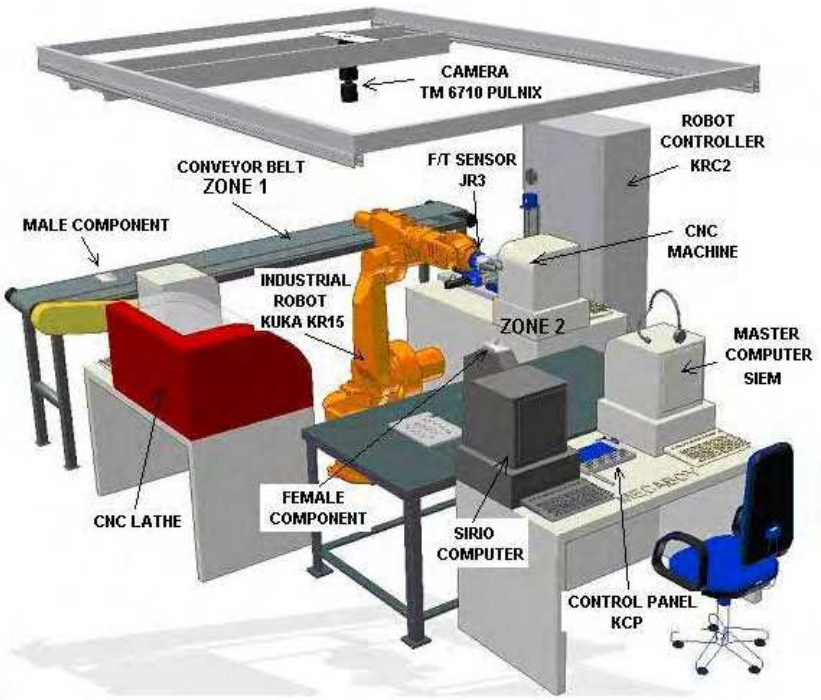


Figure 1. Manufacturing cell

The NNC for assembly is called SIEM (Sistema Inteligente de Ensamble Mecánico) and is based on a FuzzyARTMAP neural network working in fast learning mode (Carpenter, et al, 1992). The vision system, called SIRIO (Sistema Inteligente de Reconocimiento Invariante de Objetos), also uses the same neural network to learn and classify the assembly components (Pena-Cabrera & Lopez-Juarez, 2006). The SIRIO was implemented with a high speed camera CCD/B&W, PULNIX 6710, with 640x480 resolution; camera movements on the X and Y axis were implemented using a 2D positioning system.

For experimental purposes three canonical peg shapes were used: circular, square and radiused-square as it is shown in figure 2. Both, chamfered and chamferless female components were employed during experimentation.

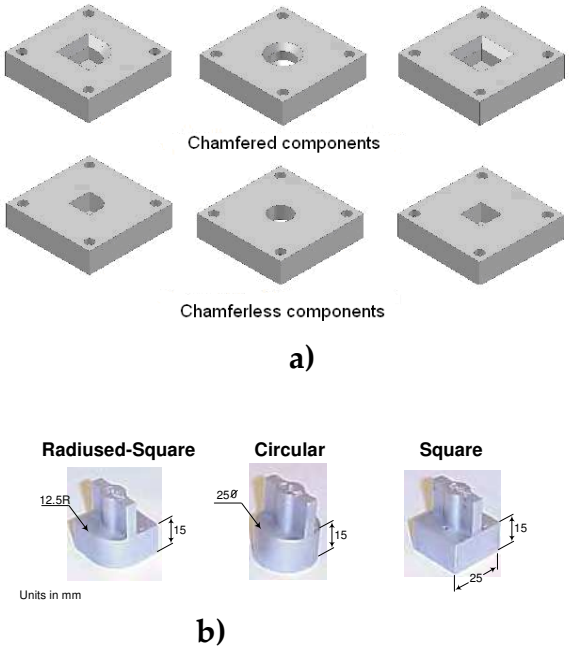


Figure 2. a) Female assembly components, b) Male assembly components

4. Assembly methodology

4.1 Pre-Configuration

4.1.1 Starting from scratch

Initially, the robot system does not have any knowledge. To accomplish the very first assembly the robot has to acquire a Primitive Knowledge Base (PKB) using an interactive method.

a) Given Primitive Knowledge Base (GVN-PKB)
The formation of the PKB basically consists of showing the robot how to react to individual components of the F/T vector. This procedure results in creating the required mapping between contact states and robot motions within the motion space– linear, angular and diagonal movements- , this is illustrated in figure 3. The Given PKB (GVN-PKB) used for the experiments reported in this chapter considered rotation around Z axis and diagonal motions as it is illustrated in figure 4.

Using the above mentioned GVN-PKB to start the learning of the assembly skill, it showed to be effective, however the robot still lacked for autonomy and it was realized that sometimes the robot did not used all the information given in the GVN-PKB and also it was noticed a difference between the taught contact forces the actual forces occurring during assembly so that an autonomously created PKB was needed in order to provide complete self-adaptive behaviour to the robot.

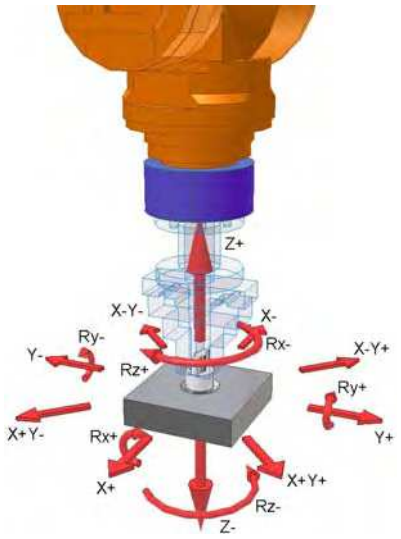


Figure 3. Motion space

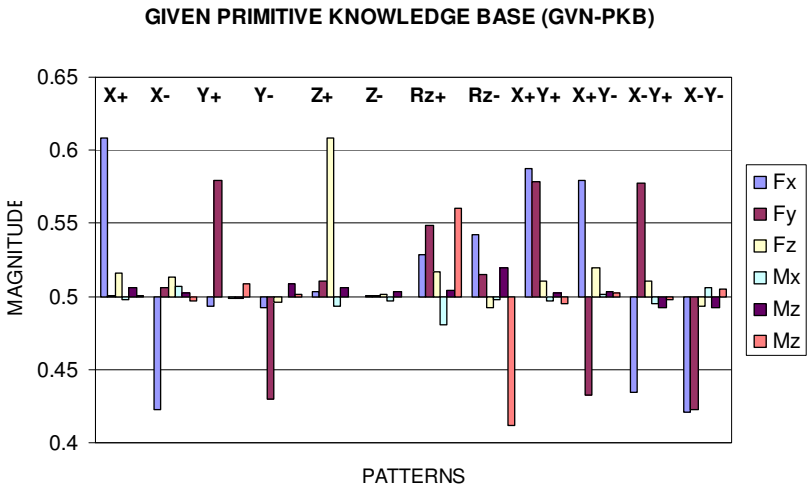


Figure 4. Given PKB (GVN-PKB)

b) Acquired Primitive Knowledge Base (ACQ-PKB)

It was decided to embed a fuzzy logic mechanism to autonomously acquire an initial knowledge from the contact states. That is, learning the mapping from scratch without knowledge about the environment. The only instruction given to the robot was the task – assembly – in order to start moving downwards. When the contact is made the robot starts acquiring information about the contact states following fuzzy rules and autonomously generating the corresponding motion commands and forming the Acquired PKB (ACQ-PKB). During the first contact, the fuzzy algorithm determines the type of operation: chamfered or chamferless assembly and chooses the rules to apply depending of moments and forces magnitude presents in X and Y directions.

Fuzzy logic have proved to be useful to model many decision taking processes in presence of uncertainty or where no precise knowledge of the process exist in an attempt to formalize experience and empiric knowledge of the experts in a specific process. The initial knowledge from our proposal comes from a static and dynamic force analysis when the components are in contact assuming that there is an error in the position with respect to the centre of insertion. With the aid of dynamic simulation software (ADAMS), the behaviour of the contact impact is obtained for different situations which are to be solved by the movements of the manipulator.

There are 12 defined motion directions (X+, X-, Y+, Y-, Z+, Z-, Rz+, Rz-, X+Y+, X+Y-, X-Y+ and X-Y-) and for each one there is a corresponding contact state. An example of these contact states for a chamfered female squared component is shown in figure 5. The contact states for linear motion X+, X-, Y+, Y-, and linear combined motions X+Y+, X+Y-, X-Y+, X-Y- are shown in figure 5(a). In figure 5(b), it is shown a squared component having four contact points. Figures 5(c) and 5(d) provide additional patterns for rotation Rz- and Rz+ respectively when the component has only one point of contact. The contact state for mapping Z+ is acquired making vertical contact between component and a horizontal surface, Z- direction is acquired with the component is in free space. This approach applies also for chamfered circular and radius-squared components as well as the chamferless components.

It is stated to use the following considerations for the generation of the fuzzy rules: a) Number of linguistic values: 2 (minimum, maximum), b) Number of input variables: 12 (*Fxp, Fxn, Fyp, Fyn, Fzp, Fzn, Mxp, Mxn, Myp, Myn, Mzp, Mzn*) and c) Maximum number of rules: $12^2 = 144$ (only 24 were used).

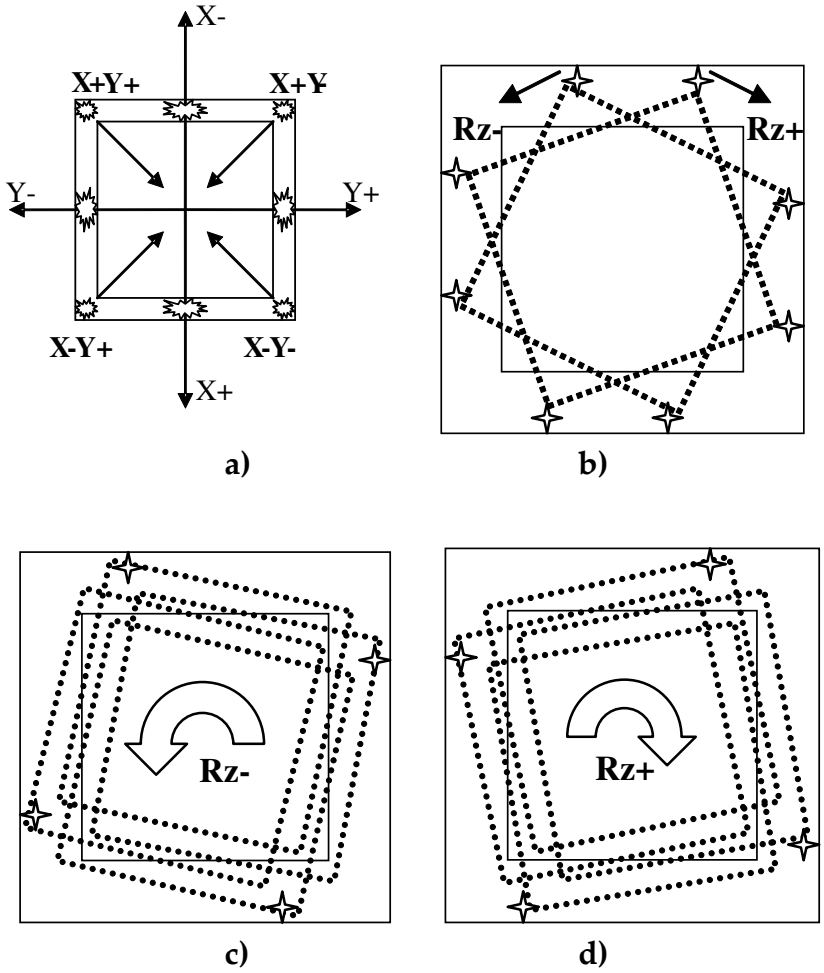


Figure 5. Contacts between chamfered components while acquiring the primitive knowledge base,

a) Linear movements,
b) Pure rotation Rz+ and Rz-,
c) Rotation Rz-,
d) Rotation Rz+.

The membership functions are stated as showed in figure 6. Forces and moments have normalised values between 0 and 1. The normalization was *ad-hoc* and considered the maximum experimental value for both, force and moment values. No belong functions were defined for the output, because our process does not includes defuzzification in the output. The function limit values are chosen heuristically and according to previous experience in the assembly operation.

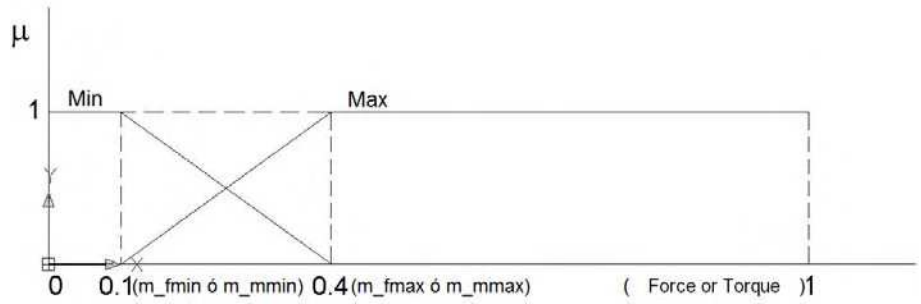


Figure 6. Membership functions

Having those membership values, antecedents and consequents defined, then the Rule Statement can be generated and the ACQ-PKB created. An example of these rules for chamfered assembly is given in table 1.

IF	Fxp	Fxn	Fyp	Fyn	Fzp	Fzn	Mxp	Mxn	Myp	Myn	Mzp	Mzn	THEN	DIR
IF	Max	Min	Min	Min	Max	Min	Min	Min	Max	Min	Min	Min	*THEN*	X+
IF	Max	Min	Min	Min	Max	Min	Max	Min	Max	Min	Min	Min	*THEN*	X+
IF	Min	Max	Min	Min	Max	Min	Min	Min	Min	Max	Min	Min	*THEN*	X-
IF	Min	Max	Min	Min	Max	Min	Min	Max	Min	Max	Min	Min	*THEN*	X-
IF	Min	Min	Max	Min	Max	Min	Min	Max	Min	Min	Min	Min	*THEN*	Y+
IF	Min	Min	Max	Min	Min	Min	Min	Min	Min	Max	Min	Min	*THEN*	Y+
IF	Min	Min	Min	Max	Max	Min	Max	Min	Min	Max	Min	Min	*THEN*	Y-
IF	Min	Min	Min	Max	Max	Min	Min	Min	Min	Min	Min	Min	*THEN*	Y-
IF	Min	Min	Min	Min	Max	Min	Min	Min	Min	Min	Min	Min	*THEN*	Z+
IF	Min	Min	Min	Min	Min	Min	Min	Min	Min	Min	Min	Min	*THEN*	Z-
IF	Min	Min	Min	Min	Max	Min	Min	Min	Min	Min	Max	Min	*THEN*	Rz+
IF	Max	Min	Min	Min	Max	Min	Min	Min	Max	Min	Max	Min	*THEN*	Rz+
IF	Min	Max	Min	Min	Max	Min	Min	Min	Min	Max	Max	Min	*THEN*	Rz+
IF	Min	Min	Max	Min	Max	Min	Min	Max	Min	Min	Max	Min	*THEN*	Rz+
IF	Min	Min	Min	Max	Max	Min	Max	Min	Min	Min	Max	Min	*THEN*	Rz+
IF	Min	Min	Min	Min	Max	Min	Min	Min	Min	Min	Min	Max	*THEN*	Rz-
IF	Max	Min	Min	Min	Max	Min	Min	Min	Max	Min	Min	Max	*THEN*	Rz-
IF	Min	Max	Min	Min	Max	Min	Min	Min	Min	Max	Min	Max	*THEN*	Rz-
IF	Min	Min	Max	Min	Max	Min	Min	Max	Min	Min	Min	Max	*THEN*	Rz-
IF	Min	Min	Min	Max	Max	Min	Max	Min	Min	Min	Min	Max	*THEN*	Rz-
IF	Max	Min	Max	Min	Max	Min	Min	Max	Max	Min	Min	Min	*THEN*	X+Y+
IF	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Min	Min	*THEN*	X+Y-
IF	Min	Max	Min	Max	Max	Min	Min	Max	Min	Max	Min	Min	*THEN*	X-Y+
IF	Min	Max	Min	Max	Max	Min	Max	Min	Min	Max	Min	Min	*THEN*	X-Y-

Table 1. Fuzzy rules for chamfered assembly

For chamferless assembly another knowledge base would have to be generated using similar rules as shown above, but without considering force in axis X and Y. The reason is that these forces in comparison with the moments generated around those axes are very small. The inference machine determines the rules to apply in a given case.

To quantify the fuzzy output response a fuzzy logic membership value is used. For the "AND" connector we used the product criteria (Driankov, et al, 1996), and to obtain a conclusion, the maximum value for the fuzzy outputs in the expression (1) response was used.

$$
\begin{aligned}
&X+ = Fxp_{\max} * Fxn_{\min} * Fyp_{\min} * Fyn_{\min} * Fzp_{\max} * Mxp_{\min} * Mxn_{\min} * Myp_{\max} * Myn_{\min} * Mzp_{\min} * Mzn_{\min}\\
&X+ = Fxp_{\max} * Fxn_{\min} * Fyp_{\min} * Fyn_{\min} * Fzp_{\max} * Mxp_{\max} * Mxn_{\min} * Myp_{\max} * Myn_{\min} * Mzp_{\min} * Mzn_{\min}\\
&X- = Fxp_{\min} * Fxn_{\max} * Fyp_{\min} * Fyn_{\min} * Fzp_{\max} * Mxp_{\min} * Mxn_{\min} * Myp_{\min} * Myn_{\max} * Mzp_{\min} * Mzn_{\min}\\
&X- = Fxp_{\min} * Fxn_{\max} * Fyp_{\min} * Fyn_{\min} * Fzp_{\max} * Mxp_{\min} * Mxn_{\max} * Myp_{\min} * Myn_{\max} * Mzp_{\min} * Mzn_{\min}\\
&Y+ = Fxp_{\min} * Fxn_{\min} * Fyp_{\max} * Fyn_{\min} * Fzp_{\max} * Mxp_{\min} * Mxn_{\max} * Myp_{\min} * Myn_{\min} * Mzp_{\min} * Mzn_{\min}\\
&Y+ = Fxp_{\min} * Fxn_{\min} * Fyp_{\max} * Fyn_{\min} * Fzp_{\min} * Mxp_{\min} * Mxn_{\min} * Myp_{\min} * Myn_{\max} * Mzp_{\min} * Mzn_{\min}\\
&Y- = Fxp_{\min} * Fxn_{\min} * Fyp_{\min} * Fyn_{\max} * Fzp_{\max} * Mxp_{\max} * Mxn_{\min} * Myp_{\min} * Myn_{\max} * Mzp_{\min} * Mzn_{\min}\\
&Y- = Fxp_{\min} * Fxn_{\min} * Fyp_{\min} * Fyn_{\max} * Fzp_{\max} * Mxp_{\min} * Mxn_{\min} * Myp_{\min} * Myn_{\min} * Mzp_{\min} * Mzn_{\min}\\
&Z+ = Fxp_{\min} * Fxn_{\min} * Fyp_{\min} * Fyn_{\min} * Fzp_{\max} * Mxp_{\min} * Mxn_{\min} * Myp_{\min} * Myn_{\min} * Mzp_{\min} * Mzn_{\min}\\
&Z- = Fxp_{\min} * Fxn_{\min} * Fyp_{\min} * Fyn_{\min} * Fzp_{\min} * Mxp_{\min} * Mxn_{\min} * Myp_{\min} * Myn_{\min} * Mzp_{\min} * Mzn_{\min}\\
&Rz+ = Fxp_{\min} * Fxn_{\min} * Fyp_{\min} * Fyn_{\min} * Fzp_{\max} * Mxp_{\min} * Mxn_{\min} * Myp_{\max} * Myn_{\min} * Mzp_{\max} * Mzn_{\min}\\
&Rz+ = Fxp_{\max} * Fxn_{\min} * Fyp_{\min} * Fyn_{\min} * Fzp_{\max} * Mxp_{\min} * Mxn_{\min} * Myp_{\max} * Myn_{\min} * Mzp_{\max} * Mzn_{\min}\\
&Rz+ = Fxp_{\min} * Fxn_{\max} * Fyp_{\min} * Fyn_{\min} * Fzp_{\max} * Mxp_{\min} * Mxn_{\min} * Myp_{\min} * Myn_{\max} * Mzp_{\max} * Mzn_{\min}\\
&Rz+ = Fxp_{\min} * Fxn_{\min} * Fyp_{\max} * Fyn_{\min} * Fzp_{\max} * Mxp_{\min} * Mxn_{\max} * Myp_{\min} * Myn_{\min} * Mzp_{\max} * Mzn_{\min}\\
&Rz+ = Fxp_{\min} * Fxn_{\min} * Fyp_{\min} * Fyn_{\max} * Fzp_{\max} * Mxp_{\max} * Mxn_{\min} * Myp_{\min} * Myn_{\min} * Mzp_{\max} * Mzn_{\min}\\
&Rz- = Fxp_{\min} * Fxn_{\min} * Fyp_{\min} * Fyn_{\min} * Fzp_{\max} * Mxp_{\min} * Mxn_{\min} * Myp_{\min} * Myn_{\min} * Mzp_{\min} * Mzn_{\max}\\
&Rz- = Fxp_{\max} * Fxn_{\min} * Fyp_{\min} * Fyn_{\min} * Fzp_{\max} * Mxp_{\min} * Mxn_{\min} * Myp_{\max} * Myn_{\min} * Mzp_{\min} * Mzn_{\max}\\
&Rz- = Fxp_{\min} * Fxn_{\max} * Fyp_{\min} * Fyn_{\min} * Fzp_{\max} * Mxp_{\min} * Mxn_{\min} * Myp_{\min} * Myn_{\max} * Mzp_{\min} * Mzn_{\max}\\
&Rz- = Fxp_{\min} * Fxn_{\min} * Fyp_{\max} * Fyn_{\min} * Fzp_{\max} * Mxp_{\min} * Mxn_{\max} * Myp_{\min} * Myn_{\min} * Mzp_{\min} * Mzn_{\max}\\
&Rz- = Fxp_{\min} * Fxn_{\min} * Fyp_{\min} * Fyn_{\max} * Fzp_{\max} * Mxp_{\max} * Mxn_{\min} * Myp_{\min} * Myn_{\min} * Mzp_{\min} * Mzn_{\max}\\
&X+Y+ = Fxp_{\max} * Fxn_{\min} * Fyp_{\max} * Fyn_{\min} * Fzp_{\max} * Mxp_{\min} * Mxn_{\max} * Myp_{\max} * Myn_{\min} * Mzp_{\min} * Mzn_{\min}\\
&X+Y- = Fxp_{\max} * Fxn_{\min} * Fyp_{\min} * Fyn_{\max} * Fzp_{\max} * Mxp_{\max} * Mxn_{\min} * Myp_{\max} * Myn_{\min} * Mzp_{\min} * Mzn_{\min}\\
&X-Y+ = Fxp_{\min} * Fxn_{\max} * Fyp_{\max} * Fyn_{\min} * Fzp_{\max} * Mxp_{\min} * Mxn_{\max} * Myp_{\min} * Myn_{\max} * Mzp_{\min} * Mzn_{\min}\\
&X-Y- = Fxp_{\min} * Fxn_{\max} * Fyp_{\min} * Fyn_{\min} * Fzp_{\max} * Mxp_{\max} * Mxn_{\min} * Myp_{\min} * Myn_{\max} * Mzp_{\min} * Mzn_{\min}
\end{aligned}
\qquad (1)
$$

Once the algorithm values have been generated, a routine which allows the manipulator for autonomous database generation is created. The mapping acquisition between generated contact states-arm motion commands starts from the insertion centre. This information is determined by calculating the centroid of the component by the vision system. Positional errors due to the image

processing are about 1 mm to 2 mm which were acceptable for the experimental work since the assembly was always successful. The manipulator starts moving in every possible direction generating a knowledge database. The results given in this research considered only 24 patterns as indicated in the fuzzy rules shown in table 1, omitting the rotations around the X and Y axis since only straight insertions were considered. Some patterns generated with this procedure for the chamfered and chamferless square peg insertion can be observed in figure 7.

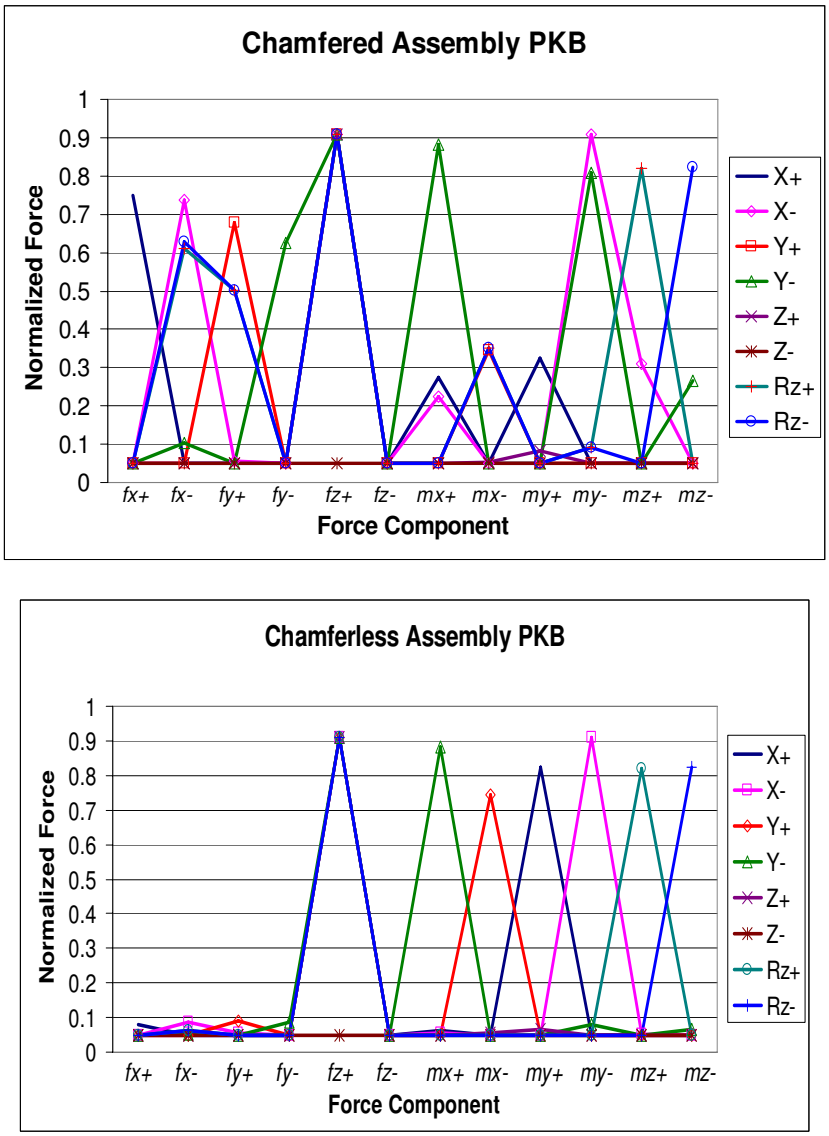


Figure 7. ACQ-PKB, left chamfered assembly, right chamferless assembly

In order to get the next motion direction the forces are read, normalized and classified using the NNC on-line. The F/T pattern obtained from the sensor provides a unique identification. The F/T vector (2) comprises 12 components given by the 6 data values (positive and negative).

$$[Current\,F/T] = \left[fx, fx-, fy, fy-, fz, fz-, mx, mx-, my, my-, mz, mz-\right]^T \quad (2)$$

4.1.2 Acquiring location and component type

The SIRIO system employs the following methodology: a) Finding the region of interest (ROI), b) Calculate the histogram of the image, d) Search for components, e) Centroid calculation, f) Component orientation, g) Calculate Boundary Object Function (BOF), distances between the centroid and the perimeter points, h) Descriptor vector generation and normalization (CFD&POSE) and i) Information processing in the neural network.
The descriptive vector is called CFD&POSE (Current Frame Descriptor and Pose) and it is conformed by (3):

$$\left[CDF\ \&\ POSE\right] = [D_1, D_2, D_3, ..., D_n, X_c, Y_c, \theta, Z, ID]^T \quad (3)$$

Where: $\mathbf{D}_i$ are the distances from the centroid to the perimeter of the object. (180 data values)

- $\mathbf{X}_C$, $\mathbf{Y}_C$, are the centroid coordinates.
- ϕ, is the orientation angle.
- **Z** is the height of the object.
- **ID** is a code number related to the geometry of the components.′

With this vector and following the above methodology, the system has been able to classify invariantly 100% of the components presented on-line even if they are not of the same size, orientation or location and for different light conditions, see (Pena-Cabrera & Lopez-Juarez, 2006) for details.

The CFD&POSE vector is invariant for each component and it is used for classification. The vector is normalized to 185 data dimension and normalized in

the range [0.0 – 1.0]. The normalization of the BOF is accomplished using the maximum divisor value of the vector distance. This method allows having very similar patterns as input vectors to the neural network, getting a significant improvement in the operation system. In our experiments, the object recognition method used the above components having 210 patterns as primitive knowledge to train the neural network. It was enough to recognize the assembly components with $\varrho_a = 0.2$ (base vigilance), $\varrho_{map} = 0.7$ (vigilance map) and $\varrho_b = 0.9$ parameters, however, the SIRIO system can recognize more complex components (Pena-Cabrera, et al, 2005).

4.2 Grasp

At this stage, the PKB has been acquired and the location information sent to the robot. The motion planning from Home position to zone 1 uses the male component given coordinates provided by SIRIO. The robot uses this information and the F/T sensor readings to grasp the piece and to control the motion in Z direction for two stages:

a) The security stage
In the event that position and orientation of the male component, given by SIRIO, have an error larger than 5 mm in *X* or *Y* axis and 10º around Z direction. Sensing is executed during 10 movements in *Z*- direction with manipulator steps of 0.2 mm. In this stage a collision is possible to occur between gripper and components. The system reacts moving to home position when a force limit in Z direction is reached (4 N). The robot continues its trajectory in Z- direction until a distance of 1 mm component is reached.

b) Grasp Component
This sensing stage begins just before the robot touches the component. The sensor is read every 0.1 mm executed by manipulator, this stage ends when the robot touches the component, in this situation the force magnitude in Z direction is at least 4 N, then the condition to grasp (close gripper) is satisfied.

4.3 Translation

The translation is similar to the grasping operation in zone 1. The path to move the robot from zone 1 to zone 2 (assembly point) is accomplished by using the

coordinates given by the SIRIO system. The possibility of collision with obstacles is avoided using bounded movements.

4.4 Assembly Operation

4.4.1 Neural Network Controller (NNC)

a) ART Models

Several works published in the literature inspired ideas about contact recognition and representation (Xiao & Liu, 1998), (Ji & Xiao, 1999), (Skubic &Volz, 1996), however the fuzzy representation appealed to be suitable to expand the NNC capability and further work was envisaged to embed the automatic mechanism to consider contact states that are actually present in a specific assembly operation. It was believed that by using only useful information, compliance learning could be effective in terms of avoiding learning unnecessary contact information, hence also avoiding unnecessary motions within the motion space.

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 Stephen Grossberg at Boston University. Grossberg suggested 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. 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 (Grossberg, 1976).

The theory has evolved in a series of real-time architectures for unsupervised learning, the ART-1 algorithm for binary input patterns (Carpenter & Grossberg, 1987). Supervised learning is also possible through ARTMAP (Carpenter, et al, 1991) that uses 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.

b) NNC Architecture

The functional structure of the assembly system is illustrated in figure 8. The Fuzzy ARTMAP (FAM) (Carpenter, et al, 1992) is the heart of the NNC. The controller includes three additional modules. The Knowledge Base that stores initial information related to the geometry of the assembling parts and which is autonomously generated. The Pattern-Motion Selection module keeps track of the appropriateness of the F/T patterns to allow the FAM network to be re-trained. If this is the case, the switch SW is closed and the corresponding pattern-action provided to the FAM for on-line retraining. The selection criterion is given by expression (3), discussed next.

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 and handling the communication with the Master Computer.

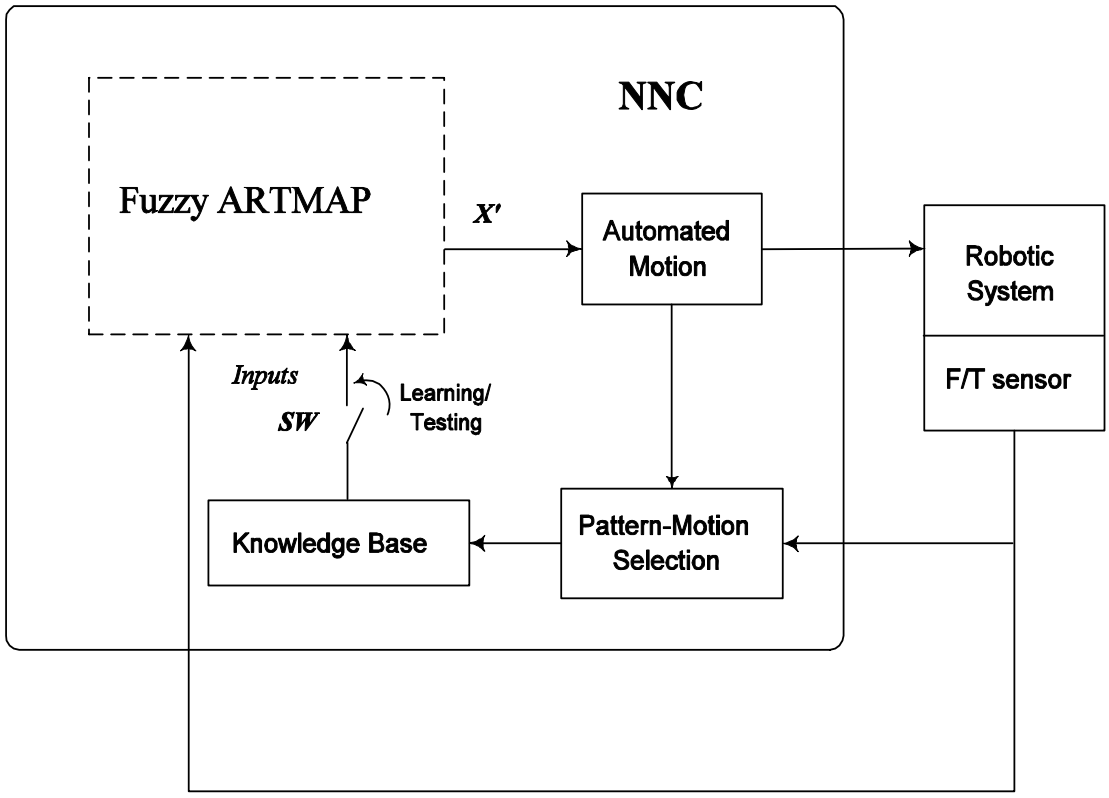


Figure 8. System Structure

c) Knowledge refinement during fine motion

There are potential overtraining problems associated with learning patterns on-line during fine motion and which are solved by the Pattern-Motion Selection module indicated in figure 8. The robot should continue moving in the insertion direction if, and only if, a minimum force value has been reached. In this situation, on-line learning is started to allow the acquisition and learning of the pattern-action pair that produced such contact state and favoured the assembly. In the event of continual learning after having reached this minimum force value, 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 to 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) How to recover from errors?
2) What is a good motion?
3) which motions should or should not be learned?

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

$$\left(F_{initial} - F_{after}\right) \geq 10 \tag{4}$$

$F_{initial}$ and F_{after} are a merit figures measured before and after the corrective motion are applied and computed using the following equation as in (Ahn, et al, 1992):

$$F = \sqrt{fx^2 + fy^2 + fz^2 + s(mx^2 + my^2 + mz^2)} \tag{5}$$

The heuristic expression (5) is used for all tasks; the scale factor s has been included in this equation in order to allow the use of different units or size com-

ponents. In our experiments, the scale factor was selected to be equal to 1 and the expression (4) is used in general for any learn task and 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.

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, see figure 3. This recovery action will certainly diminish the contact forces and will satisfy the condition given by the expression (4) in order to learn the force-action pair. However, this situation is redundant since it has already been given when the PKB was formed and further learning will corrupt the PKB changing probably the peg's assembly direction in Z+ instead Z-. Similarly, learning should not be allowed when the arm is in free-space. In this situation, $F_{initial}$ and F_{after} will be very similar and again learning another pattern in the Z- direction will be redundant. Both situations were tested experimentally and revealed that an unstable situation may appear if further learning is allowed. After the pattern-action has satisfied expression (4) and the prediction direction is not in the Z direction, the pattern is allowed to be included in the new "expertise" of the robot, PKB, now the Enhanced Knowledge Base (EKB). The above procedure can be better understood with the flowchart of the NNC processing as shown in figure 9.

4.4.2 Compliant motion during peg-in-hole operations

The robot carries out the assemblies with incremental straight and rotational motions of 0.1 mm and 0.1°, respectively. Rotation around the X and Y axes was avoided so that only straight directions were considered which means that only compliant motion in XY plane and rotation around the Z axis was considered. In order to get the next motion direction the forces are read, normalized and classified using the NNC.

Several tests were carried out to assess the compliant motion performance of the NNC using aluminum pegs with different cross-sectional geometry: circular, squared and radiused-square, see figure 2.

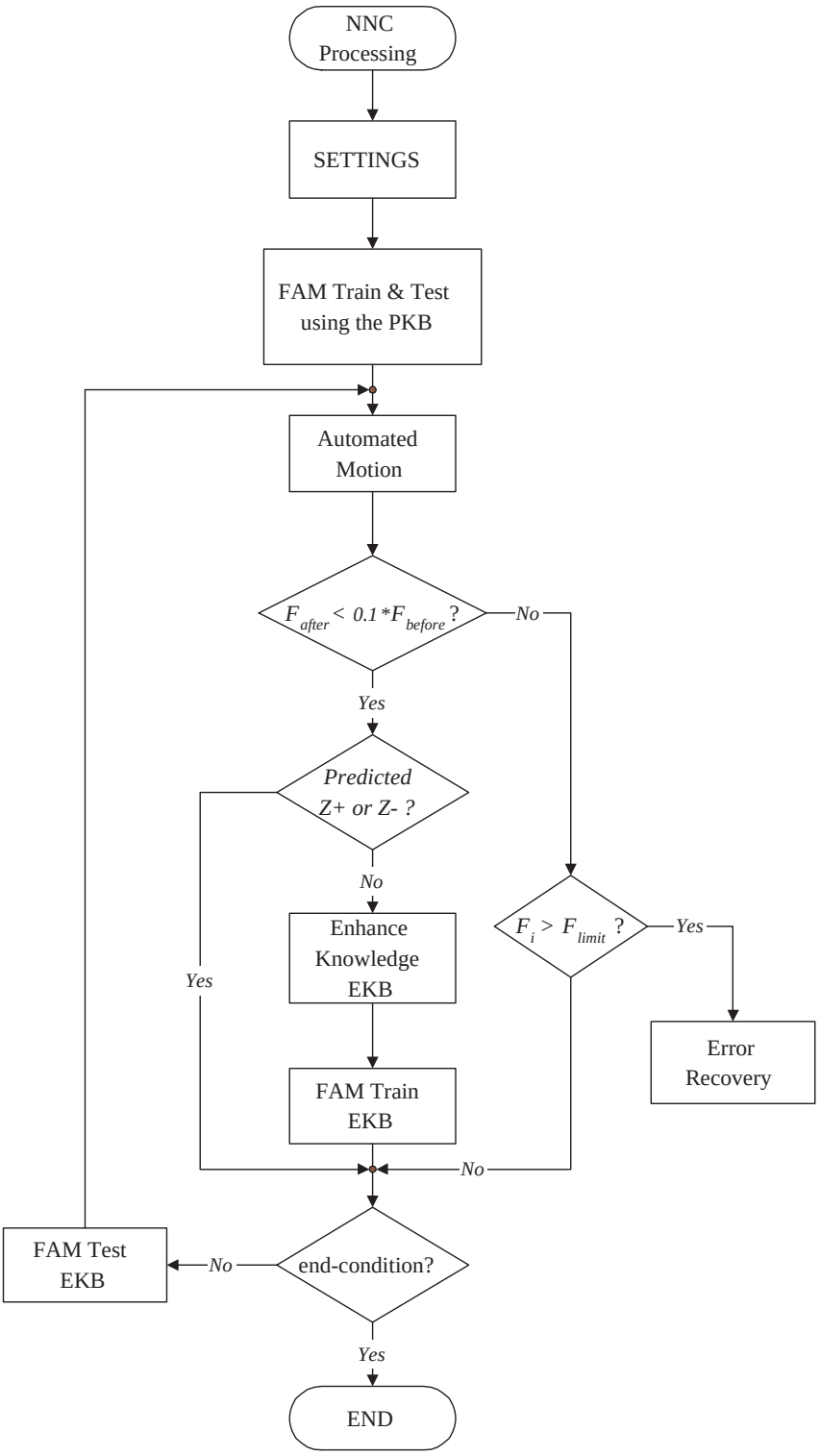


Figure 9. Flowchart of the NNC processing

The assembly was ended when 3/4 of the body of the peg were inside the hole. This represented 140 motion steps in the -Z assembly direction. A typical assembly operation is shown in figure 10.

The Fuzzy ARTMAP network parameters during experiments were set for fast learning (learning rate = 1). The values for the vigilance - in the range (0-1) - were selected based on the fact that it was required for the FuzzyARTMAP network to be as selective as possible to cluster all different patterns and which is achieved by having a high vigilance level for the of ϱ_{map} and ϱ_a; hence, this was the main criterion to select the vigilance and was not related to the task conditions (shape, offset errors). ϱ_b is small since this is increased internally according to the disparity between the input patterns and the previous recognition categories in the match tracking mechanism, for a detailed description of the Fuzzy ARTMAP architecture see (Carpenter, et al, 1992). In our experiments the values for the vigilance were as follows: $\varrho_a = 0.2$ (base vigilance), $\varrho_{map} = 0.9$ and $\varrho_b = 0.9$.

Figure 10. Peg-in-hole operation

5. Assembly Results

5.1 Assembly results using ACQ-PKB

Typical results in a chamfered squared peg insertion using the GVN-PKB are summarised in table 2.

Insertion	Offset (δx, δy, δR_z) (mm, mm, °)	Using GVN-PKB				Using ACQ-PKB			
		New Patterns	Alignment Motions	Total Motions	Time (s)	New Patterns	Alignment Motions	Total Motions	Time (s)
1	(0.7, 0.8, 0.8)	0	23	173	47.08	0	26	166	44.53
2	(-0.8, 1.1, -0.8)	1	24	178	48.19	1	36	176	47.83
3	(-0.7, -0.5, 0.8)	2	65	213	57.78	0	22	162	43.55
4	(0.8. -0.9, -0.8)	0	20	160	43.41	0	25	165	44.56
5	(0.7, 0.8, -0.8)	1	28	174	47.11	0	20	160	43.14
6	(-0.8, 1.1, 0.8)	3	30	170	46.27	1	32	173	46.48
7	(-0.7, -0.5, -0.8)	2	21	171	46.3	0	26	168	45.56
8	(0.8. -0.9, 0.8)	0	17	157	42.58	0	22	162	43.50
9	(0.7, 0.8, 0.8)	0	18	158	42.92	0	27	167	44.80
10	(-0.8, 1.1, -0.8)	3	18	158	42.77	0	28	172	46.22
11	(-0.7, -0.5, 0.8)	4	31	171	46.55	1	19	159	42.78
12	(0.8. -0.9, -0.8)	0	19	159	43.08	0	25	173	46.59
13	(0.7, 0.8, -0.8)	0	68	210	56.98	0	20	162	43.62
14	(-0.8, 1.1, 0.8)	3	38	184	49.91	1	28	168	45.30
15	(-0.7, -0.5, -0.8)	0	21	161	43.66	1	22	162	43.94
16	(0.8. -0.9, 0.8)	0	32	172	46.72	0	20	160	42.94

Table 2. Results using a GVN-PKB and ACQ-PKB

At the start of the operation different positional offsets were given as indicated in the second column. During all insertions the robot's learning ability was enabled. During the first insertion, for instance, the network learned 0 new patterns requiring 140 motions in the Z- direction and 23 motions for alignment to complete the assembly, making a total of 173 motions. The processing time for the whole insertion was 47 seconds.

Using the knowledge acquired from the squared peg insertion, the robot was also able to perform the assembly. For comparison purposes, insertions using the same offset as before were carried out and the results are given in table 2.

From the results given above in table 2, using both, the GVN-PKB and the ACQ-PKB, it can be observed that the number of new patterns using the GVN-PKB was much higher (19) compared with the number of new patterns acquired by using the ACQ-PKB (5). Learning a lower number of new patterns indicates that when using the acquired knowledge the robot needs only few

examples more which are acquired on-line. However, when using the GVN-PKB, the required number of contact force patterns needed for that specific assembly is much higher, which demonstrates a lower compliant motion capability. The robot's behaviour improved over time in terms of the assembly speed and in the number of alignment motions when the ACQ-PKB was used.
A quality measure that helps to assess the robot's dexterity is the force and moment traces during assembly and while in constraint motion. This quality measure can be obtained from the continuous monitoring of the force and torque. The quality measure during experiments using the GVN-PKB and the ACQ-PKB is illustrated in figure 11 for forces and in figure 12 for torques.

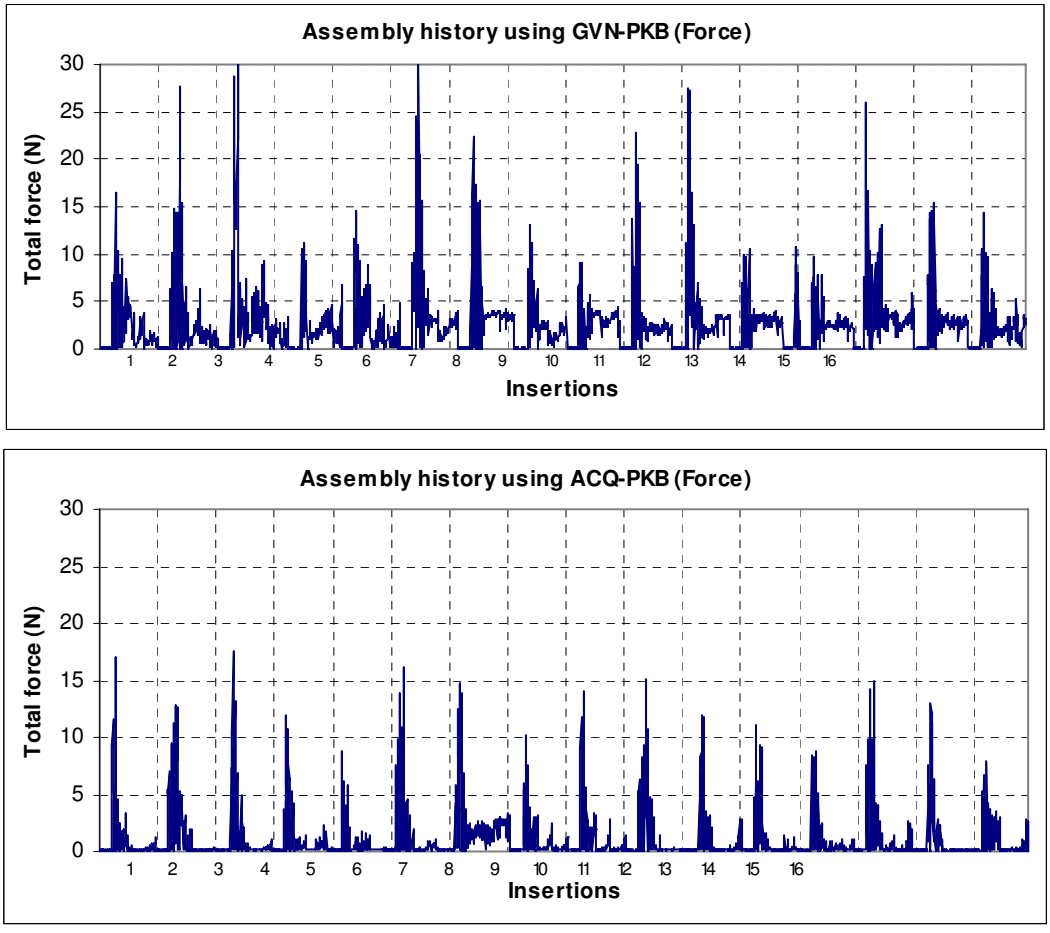


Figure 11. Forces during square chamfered peg insertion

From Figure 11 and Figure 12; it can be observed that when using the ACQ-PKB the magnitude of the forces and torques were significantly lower and in

certain cases they were almost half the value in the same experiments when using the GVN-PKB.

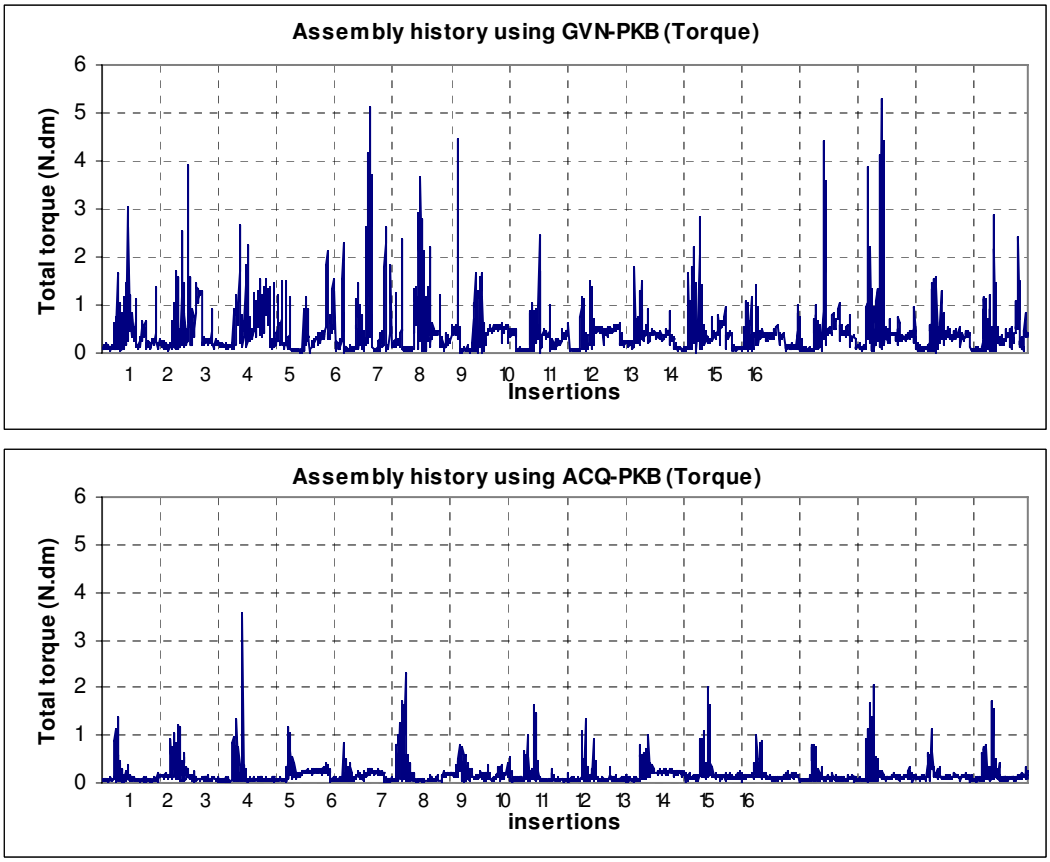


Figure 12. Torque during square chamfered peg insertion

Some forces and torques remain at the end of the insertion when the GVN-PKB is used. These residues are due to the orientation error (Rz) which is not completely recovered. The recovery of the orientation error is illustrated in figure 13, when the ACQ-PKB is used the orientation error is recovered in almost all insertions.

The total distance on XY plane by the robot is showed in figure 14 for both, GVN-PKB and ACQ-PKB. The ideal distance is de minimum distance required to reach the center point of the insertion.

Figure 15 evaluates if the robot reached de center point of the insertion in XY coordinates after the assembly end condition was satisfied, when is used an ACQ-PKB the center point was reached more efficiently than with GVN-PKB.

It was also tested the generalisation capability of the NNC by assembling different components using the same ACQ-PKB. Results are provided in table 3. For the insertion of the radiused-square component, the offsets were the same as before and for the insertion of the circular component a higher offset was used and no rotation was given. The time for each insertion was computed with the learning ability on (L_{on}) and also with learning inhibited (L_{off}); that is, using only the initial ACQ-PKB. The assembly operation was always successful and in general faster in most cases when the learning was enabled compared with inhibited learning.

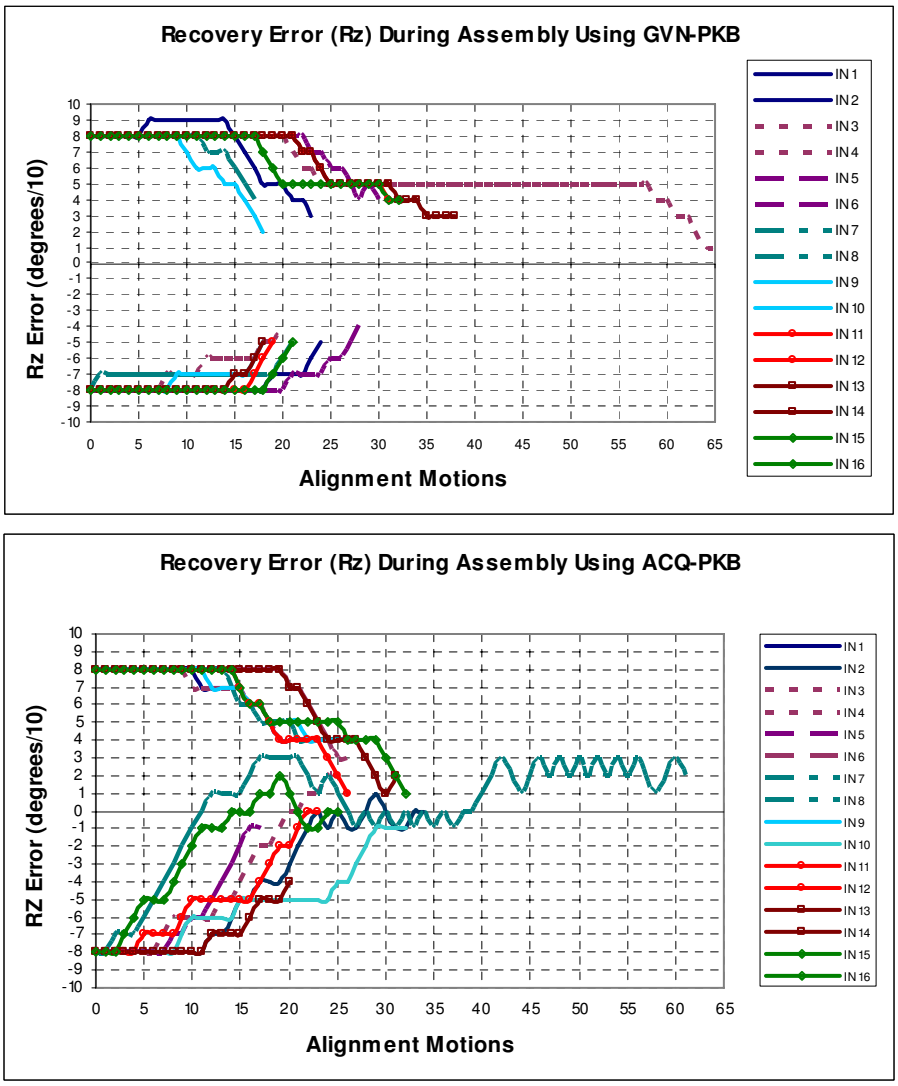


Figure 13. Recovery error (Rz) during assembly

Radiused-square chamfered peg insertion				Circular chamfered peg insertion		
Inser-tion	Offset (dx, dy, dRz) (mm, mm, °)	L_{on} time (s)	L_{off} time (s)	Offset (dx, dy, dRz) (mm, mm, °)	L_{on} time (s)	L_{off} time (s)
1	(0.7, 0.8, 0.8)	45	48	(0.7, 0.8, 0)	42	43
2	(-0.8, 1.1, -0.8)	45	51	(-0.8, 1.1, 0)	41	41
3	(-0.7, -0.5, 0.8)	43	47	(0.8. -0.9, 0)	40	42
4	(0.8. -0.9, -0.8)	50	54	(0.8. -0.9, 0)	41	41
5	(0.7, 0.8, -0.8)	44	44	(-0.8, 1.1, 0)	41	41
6	(-0.8, 1.1, 0.8)	53	51	(0.8. -0.9, 0)	41	42
7	(-0.7, -0.5, -0.8)	54	55	(1.4, 1.6, 0)	45	45
8	(0.8. -0.9, 0.8)	50	49	(1.6. -1.8, 0)	43	45
9	(0.7, 0.8, 0.8)	46	46	(1.4, 1.6, 0)	43	44
10	(-0.8, 1.1, -0.8)	45	55	(-1.4, -1, 0)	42	43
11	(-0.7, -0.5, 0.8)	44	45			
12	(0.8. -0.9, -0.8)	53	51			
13	(0.7, 0.8, -0.8)	43	43			
14	(-0.8, 1.1, 0.8)	53	51			
15	(-0.7, -0.5, -0.8)	44	59			
16	(0.8. -0.9, 0.8)	45	50			

Table 3. Results using an ACQ-PKB

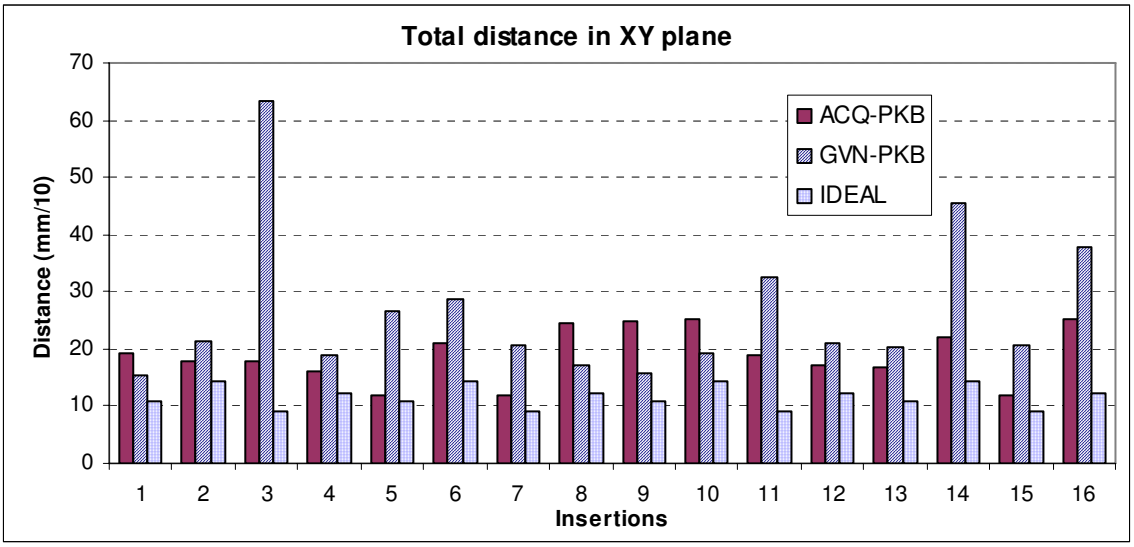


Figure 14. Total distance on XY plane

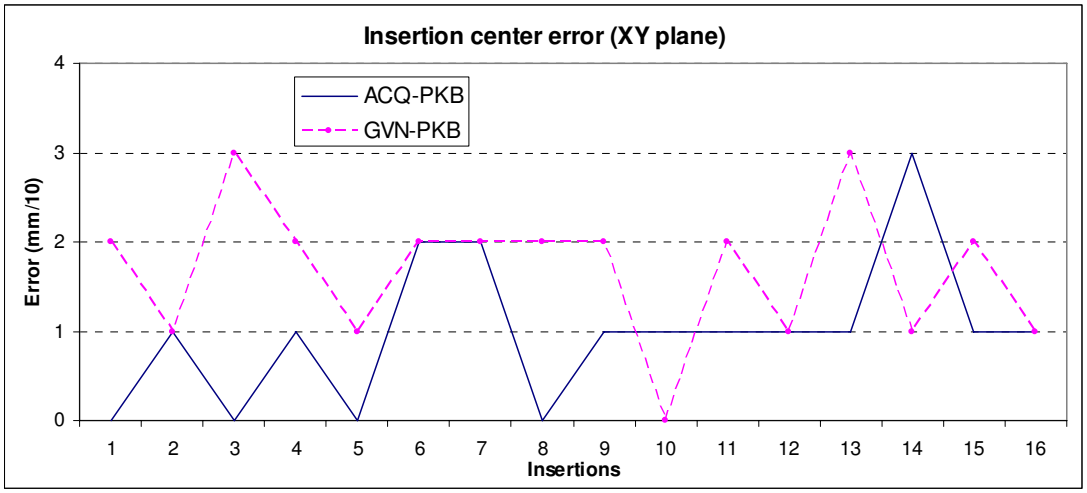


Figure 15. Insertion center error on XY plane

5.2 Whole assembly process results

Several tests were carried out to asses the performance. The diameter of the male components was 24.8 mm whereas the diameter of female components was 25 mm; the chamfer was set to 45° and 5 mm width. Results are contained in table 4. In zone 2 the SIRIO only provides location (X,Y) because the female component orientation was fixed, however an error occurs and it is related to the component's tolerance. The error for the chamfered square component is 0.8°, 0.5° for the chamfered radiused-square and 0.4° for the chamferless square and 0.6° for the chamferless radiused-square. Error recovery is illustrated in figure 18. The assembly operation ends when ¾ of the body of male component is in the hole, this represents 14 mm. The NNC was operated during the first 10 mm (100 manipulator steps), the FuzzyARTMAP parameters were: $\varrho_a = 0.2$, $\varrho_{map} = 0.9$ and $\varrho_b = 0.9$.

Table 4 shows the position errors in zone 2 which is represented in figures 16 and 17 as the trajectory followed by the robot. The minimum time of assembly cycle was **1:11** min, the maximum was **1:24** min and the average time was 1.17 min.

The system has an average angular error of **3.11**° and a maximum linear position error from -1.3 mm to 3.1 mm due to the camera positioning system in Zone 1.

# IN	P	Ch	TC (min)	TA (s)	ZONE 1			Zone 1 Error			ZONE 2		Zone 2 Error		NC
					Xmm	Ymm	RZ°	Xmm	Ymm	RZ°	Xmm	Ymm	Xmm	Ymm	
1	S	Y	1:15	32.5	62.4	144.1	10	0.2	-1.3	0	84.6	102.1	0.3	-1	Y
2	S	Y	1:15	30.4	62.4	45.7	12	1.8	0.2	2	85.6	101.1	-0.7	0	Y
3	S	Y	1:15	31.8	178.7	47.7	23	0.9	-0.8	3	84.7	100.9	0.2	0.2	Y
4	R	Y	1:11	30.1	181.6	147	29	-0.3	-0.7	-1	84.7	100.6	0.2	0.5	Y
5	R	Y	1:14	29.4	62.4	145.1	36	0.2	-0.3	-4	84.9	100.7	0	0.4	Y
6	R	Y	1:19	29.6	67.3	44.8	48	3.1	-0.7	-2	85.3	101.6	-0.4	-0.5	Y
7	C	Y	1:15	29.6	180.6	49.6	57	1	1.1	-3	84.6	102.4	0.3	-1.3	Y
8	C	Y	1:13	30.2	180.6	148	77	-0.7	0.3	7	84.3	101	0.6	0.1	Y
9	C	Y	1:14	30.2	61.5	146	79	-0.7	0.6	-1	83.9	101.6	1	-0.5	Y
10	S	N	1:18	29.9	63.4	45.7	83	-0.8	0.2	-7	85.4	100.5	-0.5	0.6	Y
11	S	N	1:19	30.4	179.6	48.6	104	0	0.1	4	83.2	100.8	1.7	0.3	Y
12	S	N	1:22	34.6	180.6	147	104	-0.7	-0.7	-6	83.2	101.8	1.7	-0.7	Y
13	R	N	1:22	38.3	61.5	146	119	-0.7	0.6	-1	84.8	102.8	0.1	-1.7	Y
14	R	N	1:22	36.8	63.4	43.8	126	-0.8	1.7	-4	83.6	101.8	1.6	-0.7	Y
15	R	N	1:24	36.6	179.6	47.7	138	0	-0.8	-2	83.2	101.7	1.7	-0.6	Y
16	C	N	1:17	30.5	182.6	149	150	1.3	1.3	0	83.7	101.2	1.2	-0.1	Y
17	C	N	1:15	28.3	63.4	146	155	1.2	0.6	-5	84.6	100.7	0.3	0.4	Y
18	C	N	1:15	29.7	64.4	47.7	174	0.2	2.2	4	83.9	101.1	1	0	Y

Table 4. Eighteen different assembly cycles, where IN= Insertion, P=piece, Ch=chamfer present, TC=Assembly cycle time, TA= Insertion time, NC=correct neural classification, S=square, R=radiused-square, C=circle, N=no and Y=yes.

The force levels in chamferless assemblies are higher than the chamfered ones. In the first one, the maximum value was in Z+, 39.1 N for the insertion number 16, and in the chamfered the maximum value was 16.9 N for the insertion number 9.

In chamfered assembly, in figure 16, it can be seen that some trajectories were optimal like in insertions 2, 5, 7, 8 and 9, which was not the case for chamferless assembly; however, the insertions were correctly completed.

In figure 17, each segment corresponds to alignment motions in other directions different from Z-. The lines mean the number of Rz+ motions that the robot performed in order to recover the positional error for female components. The insertion paths show how many rotational steps are performed. The maximum alignment motions were 22 for the chamfered case in comparison with 46 with the chamferless component.

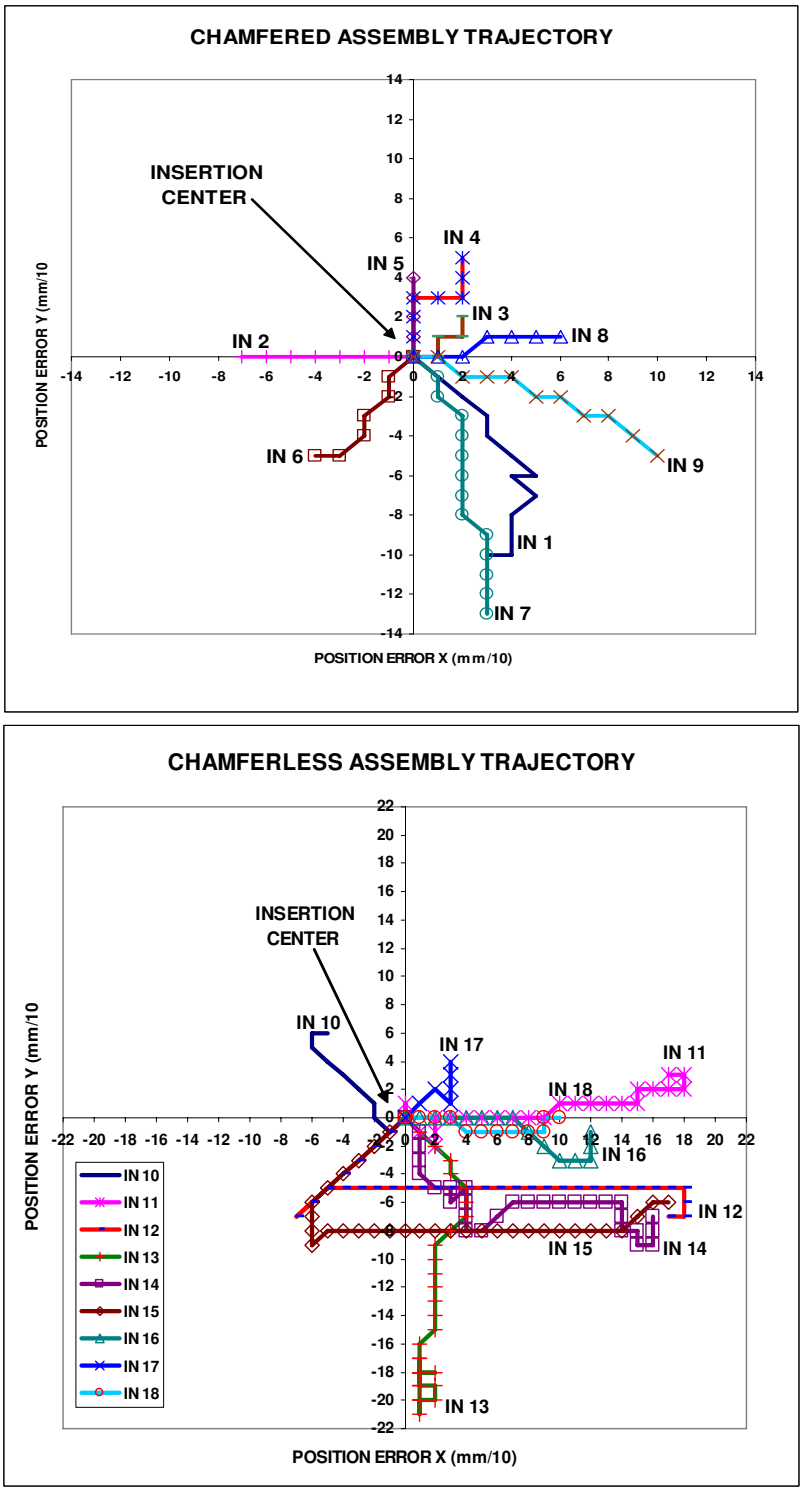
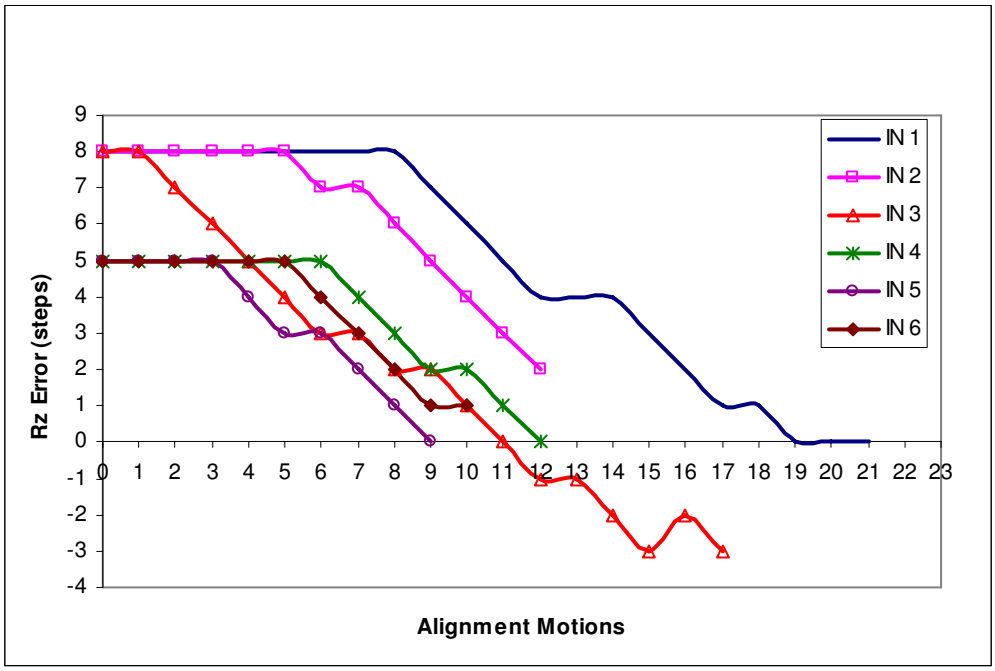


Figure 16. Assembly trajectory in top view for each insertion in zone 2. The trajectory starts with the labels (INx) and ends at 0,0 origins coordinate



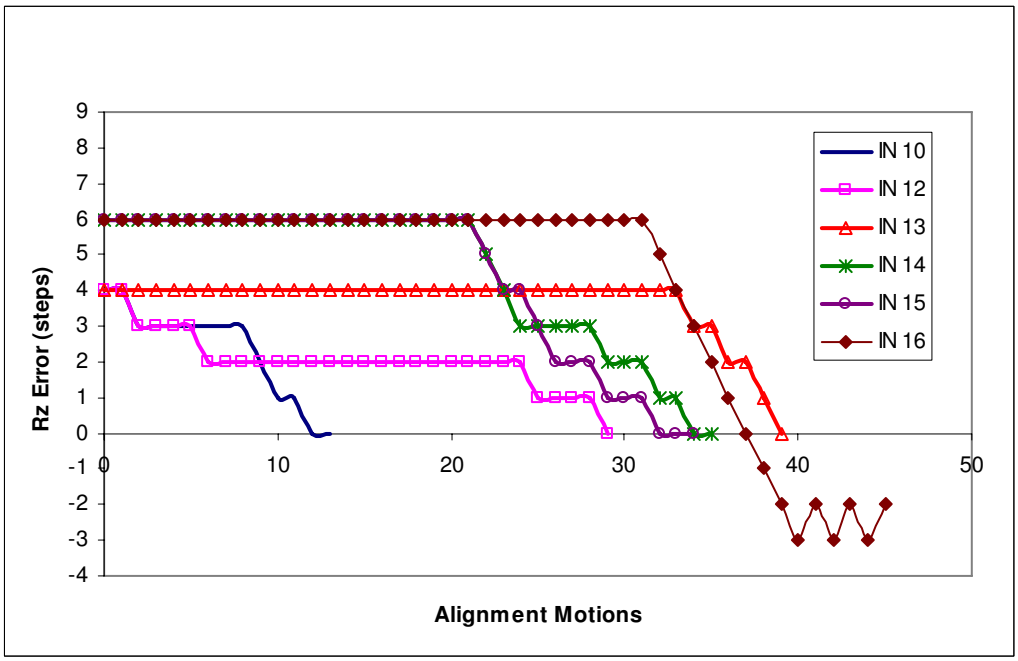


Figure 17. Compliant rotational motions (only Rz+) for each insertion in zone 2, left chamfered assembly, right chamferless assembly

6. Conclusions

A task planner approach for peg-in-hole automated assembly was presented. The proposed methodologies were used to achieve the tasks and tested successfully in the real world operations using an industrial manipulator. The robot is able to perform not only the assembly but also it can start working without initial knowledge about the environment, and it can increase its PKB at every assembly if it is necessary.

The presented approach using the vision and force sensing system has envisaged further work in the field of multimodal learning in order to fuse information and to increase the prediction capability of the network which contributes towards the creation of truly self-adaptive industrial robots for assembly.

All assemblies were successful showing the system robustness against different uncertainties and its generalization capability. The generalization of the NNC has been demonstrated by assembling successfully different component geometry using different mechanical tolerances and offsets employing the same acquired knowledge base.

Initial knowledge is acquired from actual contact states using explorative motions guided by fuzzy rules. The knowledge acquisition stops once the ACQ-PKB is fulfilled. Later this knowledge is refined as the robot develops new assembly tasks.

The dexterity of the robot improves using the ACQ-PKB by observing the magnitude of forces and moments as shown in Figures 11 and 12. Values are significantly lower, hence motions were more compliant in this case indicating that information acquired directly from the part geometry allowed also lower constraint forces during manipulation. Having implemented the knowledge acquisition mechanism, the NNC acquires only real contact force information from the operation. In comparison with our previous results, insertion trajectories improved enormously; we believe that given *a priori* knowledge (GVN-PKB) is fine, but contact information extracted directly form the operation itself provides the manipulator with better compliant motion behaviour.

Results from this work have envisaged further work in the area of multimodal data fusion (Lopez-Juarez, et al, 2005). We expect that data fusion from the F/T sensor and the vision system result in an improved confidence for getting the contact information at the starting of the operation providing also important information such as chamfer presence, part geometry and pose information, which will be the input data to a hierarchical task level planner as pointed out by (Lopez-Juarez & Rios-Cabrera, 2006).

7. References

Ahn, D.S.; Cho, H.S.; Ide, K.I.; Miyazaki, F.; Arimoto, S. (1992). Learning task strategies, in robotic assembly systems. *Robotica* Vol. 10, 10 (409–418)

Asada, H. (1990). Teaching and Learning of Compliance Using Neural Nets. IEEE *Int Conf on Robotics and Automation,* 8 (1237-1244)

Baeten, J.; Bruyninckx, H.; De Schutter, J. (2003). Integrated Vision/Force Robotic Servoing in the Task Frame Formalism. *The International Journal of Robotics Research*. Vol. 22, No. 10-11, 14 (941-954)

Carpenter, G. A.; Grossberg, S. (1987). A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine. *Computer Vision, Graphics, and Image Processing,* Academic Press, Inc. 62 (54-115)

Carpenter, G. A.; Grossberg, S.; Reynolds, J. H. (1991). ARTMAP: Supervised Real-Time Learning and Classification of Nonstationary Data by Self-Organizing Neural Network. *Neural Networks,* 24 (565-588)

Carpenter, G.A.; Grossberg, J.; Markunzon, N.; Reynolds, J.H.; Rosen, D.B. (1992). Fuzzy ARTMAP: A Neural Network Architecture for Incremental Learning of Analog Multidimensional Maps. IEEE *Trans. Neural Networks,* Vol. 3, No. 5, 36 (678-713)

Cervera, E.; del Pobil, A. P. (1996). Learning and Classification of Contact States in Robotic Assembly Tasks. Proc *of the 9th Int. Conf. IEA/AIE*, 6 (725-730)

Cervera, E.; Del Pobil, A.P. (1997). Programming and Learning in Real World Manipulation Tasks. In: *Int. Conf. on Intelligent Robot and Systems (IEEE/RSJ,* Proc. 1, 6 (471-476)

Cervera, E.; del Pobil, A. P. (2002). Sensor-based learning for practical planning of fine motions in robotics. *The International Journal of Information Sciences,* Special Issue on Intelligent Learning and Control of robotics and Intelligent Machines in Unstructured Environments, Vol. 145, No. 1, 22 (147-168)

De Schutter, J.; Van Brussel, H. (1988). Compliant Robot Motion I, a formalism for specifying compliant motion tasks. *The Int. Journal of Robotics Research*, Vol. 7, No. 4, 15 (3-17)

Doersam, T.; Munoz Ubando, L.A. (1995). Robotic Hands: Modelisation, Control and Grasping Strategies. In: *Meeting annuel de L'Institute Fanco-Allemand pour les Application de la recherche IAR*

Driankov, D.; Hellendoorn, H.; Reinfrank, M. (1996). *An Introduction to Fuzzy Control*. 2nd ed. Springer Verlag

Erlbacher, E. A. *Force Control Basics*. PushCorp, Inc. (Visited December 14th,

2004). http://www.pushcorp.com/Tech%20Papers/Force-Control-Basics.pdf

Grossberg, S. (1976). Adaptive Pattern Classification and universal recoding II: Feedback, expectation, olfaction and illusions. *Biological Cybernetics,* Vol. 23, 16 (187-202)

Gullapalli, V.; Franklin, J. A.; Benbrahim, H. (1994). Acquiring Robot Skills via Reinforcement Learning. IEEE *Control Systems,* 12 (13-24)

Gullapalli, V.; Franklin, J.A.; Benbrahim, H. (1995). Control Under Uncertainty Via Direct Reinforcement Learning. *Robotics and Autonomous Systems*. 10 (237-246)

Howarth, M. (1998). *An Investigation of Task Level Programming for Robotic Assembly*. PhD thesis. The Nottingham Trent University, UK

Ji, X.; Xiao, J. (1999). Automatic Generation of High-Level Contact State Space. *Proc. of the Int. Conf. on Robotics and Automation*, 6 (238-244)

Joo, S.; Miyasaki, F. (1998). Development of a variable RCC and its applications. *Proceedings of the 1998 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems,* Vol. 2, 7 (1326-1332)

Jörg, S.; Langwald, J.; Stelter, J.; Natale, C.; Hirzinger, G. (2000). Flexible Robot Assembly Using a Multi-Sensory Approach. In: *Proc. IEEE Int. Conference on Robotics and Automation,* 8 (3687-3694)

Kaiser, M.; Dillman, M.R. (1996). Building Elementary Robot Skills from Human demonstration. IEEE *International Conference on Robotics and Automation*, Minneapolis, Minnesota, 6 (2700 – 2705)

Lopez-Juarez, I.; Howarth, M.; Sivayoganathan, K. (1996). Robotics and Skill Acquisition. A. Bramley; T. Mileham; G. Owen. (eds.) In: *Advances in Manufacturing Technology X,* ISBN 1 85790 031 6, 5 (166-170)

Lopez-Juarez, I. (2000). *On-line learning for robotic assembly using artificial neural networks and contact force sensing*. PhD thesis, Nottingham Trent University, UK

Lopez-Juarez, I.; Ordaz-Hernandez, K.; Pena-Cabrera, M.; Corona-Castuera, J.; Rios-Cabrera, R. (2005). On the Design of A multimodal cognitive architecture for perceptual learning in industrial robots. In MICAI 2005: Advances in Artificial Intelligence. LNAI 3789. Lecture Notes in Artificial Intelligence. A Gelbukh, A de Albornoz, and H Terashima (Eds.). 10 (1062-1072). Springer Verlag, Berlin

Lopez-Juarez, I.; Rios-Cabrera, R. (2006). Distributed Architecture for Intelligent Robotic Assembly, Part I: Design and Multimodal Learning. Ad-

vances Technologies: Research-Development-Application. Submitted for publication

Lozano-Perez T.; Mason, M.T.; Taylor R. H. (1984). Automatic Synthesis of Fine Motion Strategies. *The Int. Journal of Robotics Research,* Vol. 3 No. 1, 22 (3-24)

Mason, M. T. (1983). *Compliant motion,* Robot motion, Brady M et al eds. Cambridge: MIT Press

Peña-Cabrera, M.; López-Juárez, I.; Ríos-Cabrera R.; Corona-Castuera, J. (2005). Machine vision learning process applied to robotic assembly in manufacturing cells. *Journal of Assembly Automation*, Vol. 25, No. 3, 13 (204-216)

Peña-Cabrera, M.; Lopez-Juarez, I. (2006). Distributed Architecture for Intelligent Robotic Assembly, Part III: Design of the Invariant Object Recognition System. Advances Technologies: Research-Development-Application. Submitted for publication

Skubic M.; Volz, R. (1996). Identifying contact formations from sensory patterns and its applicability to robot programming by demonstration. *Proceedings of the 1996 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, Osaka, Japan

Skubic, M.; Volz, R.A. (2000). Acquiring Robust, Force-Based Assembly Skills from Human Demonstration. IEEE *Trans. on Robotics and Automation,* Vol. 16, No. 6, 10 (772-781)

Whitney. D.; Nevis, J. (1979). What is the Remote Center Compliance (RCC) and what can it do?. *Proceedings of the 9th Int. Symposium on Industrial Robots*, 18 (135-152)

Xiao J.; Liu, L. (1998). Contact States: Representation and Recognizability in the Presence of Uncertainties. IEEE/RSJ *Int. Conf. Intell. Robots and Sys*

Manufacturing the Future

Edited by Vedran Kordic, Aleksandar Lazinica and Munir Merdan

ISBN 3-86611-198-3

Hard cover, 908 pages

Publisher Pro Literatur Verlag, Germany / ARS, Austria

Published online 01, July, 2006

Published in print edition July, 2006

The primary goal of this book is to cover the state-of-the-art development and future directions in modern manufacturing systems. This interdisciplinary and comprehensive volume, consisting of 30 chapters, covers a survey of trends in distributed manufacturing, modern manufacturing equipment, product design process, rapid prototyping, quality assurance, from technological and organisational point of view and aspects of supply chain management.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Jorge Corona Castuera and Ismael Lopez Juarez (2006). Distributed Architecture for Intelligent Robotic Assembly Part II: Design of the Task Planner, Manufacturing the Future, Vedran Kordic, Aleksandar Lazinica and Munir Merdan (Ed.), ISBN: 3-86611-198-3, InTech, Available from: http://www.intechopen.com/books/manufacturing_the_future/distributed_architecture_for_intelligent_robotic_assembly_part_ii__design_of_the_task_planner

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821