Robotica (2006) volume 24, pp. 657–671. © 2006 Cambridge University Press
doi:10.1017/S0263574706002839 Printed in the United Kingdom

Behaviour-based approach for skill acquisition during assembly operations, starting from scratch

J. Corona-Castuera* and I. Lopez-Juarez

Grupo de Investigación en Mecatrónica y Sistemas Inteligentes de Manufactura (GIMSIM) CIATEQ A.C., Centro de Tecnología Avanzada, Manantiales 23A, Fracc. Ind. B.Q., El Marques, Querétaro CP 76246 (Mexico)

(Received in Final Form: March 2, 2006. First published online: May 11, 2006)

SUMMARY

Industrial robots in poorly structured environments have to interact compliantly with this environment for successful operations. In this paper, we present a behaviour-based approach to learn peg-in-hole operations from scratch. The robot learns autonomously the initial mapping between contact states to motion commands employing fuzzy rules and creating an Acquired-Primitive Knowledge Base (ACQ-PKB), which is later used and refined on-line by a Fuzzy ARTMAP neural network-based controller. The effectiveness of the approach is tested comparing the compliant motion behaviour using the ACQ-PKB and *a priori* Given-Primitive Knowledge Base (GVN-PKB). Results using a KUKA KR15 industrial robot validate the approach.

KEYWORDS: Compliant motion; Knowledge discovery; Peg-in-hole; Fuzzy ARTMAP; Skill acquisition.

1. INTRODUCTION

The success of assembly operations 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[1] or other improved versions of the device.[2] Another alternative is to use Active compliance, which modifies either the position of the manipulated component as a response to constraint forces or the desired force. Some commercial devices have emerged in recent years to aid industrial applications.[3]

Active compliance can be roughly divided into fine motion planning and reactive control. Fine motion planning relies on geometrical path planning whereas reactive control relies on the synthesis of an accommodation matrix or mapping that transforms the corresponding contact states to corrective motions. A detailed analysis of active compliance can be found in works by Mason[4] and De Schutter.[5] Perhaps, one of the most significant works in fine motion planning is the one developed by Lozano-Perez, Mason and Taylor known as the LMT approach.[6] 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 divided further 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.

The framework of the research presented here is situated under the connectionist-based approach employing reactive contact forces; hence, a behaviour-based approach for compliant motion during part mating. In this behaviour-based approach, the task is learned from scratch since no information is given about the environment. In previous works, the robot was provided with *a priori* Contact States-Arm Motion Commands mapping knowledge given by a human operator (see Section 4.2.1), which provided a primitive reflex system to build-up its knowledge. In this work, the mapping is achieved by using fuzzy rules that create autonomously an Acquired-Primitive Knowledge Base (ACQ-PKB) without human intervention. This ACQ-PKB is then used by the Neural Network Controller (NNC) for compliance learning. The NNC has a mechanism to assess its performance while in compliant motion, during assembly. If appropriate, new encountered contact states that favoured the assembly are incorporated into the knowledge base improving the robot's dexterity as measured in reduced assembly times and improved assembly trajectories.

The remainder of this paper is structured as follows. Section 2 reviews related work and states our contribution to the field of behaviour-based approaches to compliant motion for robotic assembly. In Section 3, issues regarding knowledge acquisition and learning using the Adaptive Resonance Theory (ART) model, which inspired our work,

* Corresponding author. E-mail: jcorona@ciateq.mx

are described and the ART-based NNC explained. Section 4 describes the robotic work cell used for the experimental work, and the results using both the ACQ-PKB and the Given-PKB (GVN-PKB) are presented. Finally, conclusions are presented and future work is described.

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,[7–9] or in implementation on real robots.[10–13] In these methods, Reinforcement Learning (RL), unsupervised- and supervised-type networks have been used.

The reinforcement algorithm implemented by V. Gullapalli demonstrated the ability to learn circular and square peg insertions. The controller was a backpropagation 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 command. The performance of the operation was evaluated by a parameter r, which measured the performance of the controller. It varied between 0 and 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.[14] 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 backpropagation also 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.[12] A cycle means an actual motion that diminished the forces acting on the peg. For the square peg, the number of cycles increased dramatically to 3750. 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 as 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[10]; 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, 4000 trials were needed 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.

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 and Dillman[15] 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 skill refinement, 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 and Volz,[16] use a hybrid control model, which provides continuous low-level force control with higher level discrete event control. Learning of an assembly skill involves learning the mapping of force sensor signals to Single-Ended Contact Formations (SECF), the sequences of SECFs and the transition velocity commands that 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 the 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 transition velocities that comprise the rest of the skill.

The above approaches can be divided in two groups: those providing autonomous assembly skill and those that teach the skill by demonstration. These approaches have given some inputs to our research and the work presented here aims 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 also worked well during the assembly of different types of components. In the case of teaching the skill by demonstration, the method shown by Kaiser and Dillman was lengthy for real-world problems and the work by Skubic and Volz assumed that during supervised training the operator must know which SECF classes are to be included in the set.

2.1. *Original contribution*

As stated earlier, three stages are clearly recognised within the domain of robotic assembly. The first stage to accomplish for compliant motion was addressed by Lopez-Juarez using an unsupervised ART-1 network that showed the possibility of recognising and classifying all contact states occurring during peg-in-hole operations.[17] Further research looked at motor control, compliant behaviour evaluation and knowledge refinement.[18] However, the robot was still recognised as lacking autonomous operation for task-level programming since the mapping between contact states to corrective motions was provided with an *a priori* knowledge base.[19] The grounding idea for the work reported in this

https://doi.org/10.1017/S0263574706002839 Published online by Cambridge University Press

paper was to learn the assembly skill from scratch, without any knowledge and just instructing the robot with the task: assembly.

Related work was reviewed in the previous section. Some work has been done in simulations and a little with industrial robots considering real-world uncertainties. Our NNC outperforms previous approaches, which is demonstrated by its generalization and robustness properties. The generalization it is demonstrated by assembling different types of components and its robustness by completing always the assembly tasks. These properties are very important when working in real operations under extreme uncertainty. More important is the fact that the required assembly knowledge is embedded into the NNC from the beginning through the contact of the mating pairs and no supervision is needed.

To the best of our knowledge, this is the first time results are being reported for autonomous knowledge acquisition and refinement using ART theory and industrial robots, which also provide a foundation for the creation of truly self-adaptive industrial robots for assembly.

3. NEURAL NETWORK CONTROLLER

3.1. Inspiring ideas and ART models

Several works published earlier inspired ideas about contact recognition and representation,[20–22] however, the fuzzy representation seemed 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.

Knowledge can be built either empirically or by hand as suggested by Towell and Shavlik.[23] Empirical knowledge can be thought of as giving examples on how to react to certain stimuli without any explanation and hand-built knowledge where the knowledge is acquired by only giving explanations without examples. It was determined that in robotic systems, a suitable strategy should include a combination of both methods. Furthermore, this idea is supported by psychological evidence that suggests that theory and examples interact closely during human learning.[24] Following this approach, knowledge is first inserted into the Artificial Neural Network (ANN) using symbolic information in the form of fuzzy rules. Once the knowledge is in neural representation (numeric or subsymbolic representation), then the initial knowledge can be refined by using on-line training examples and the resulting ANN can then be used, if required, for extracting symbolic information (if–then rules).

Learning in natural cognitive systems, including our own, follows a sequential process as demonstrated in our daily life. Events are learnt incrementally, for instance, during childhood when we start making new friends, we also learn more faces and this process continues throughout life. This learning is also stable because the learning of new faces 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).[24]

The Adaptive Resonance Theory is a well-established associative brain and competitive model introduced as a theory of human cognitive processing developed by Stephen Grossberg at Boston University. Grossberg suggested that connectionist models should be able to adaptively switch between their *plastic* and *stable* modes. That is, a system should not only exhibit plasticity to accommodate new information regarding unfamiliar events, but also 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.[25]

The theory has evolved in a series of real-time architectures for unsupervised learning, the ART-1 algorithm for binary input patterns.[26] Supervised learning is also possible through ARTMAP[27] 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 name a few.

3.2. NNC architecture

The functional structure of the assembly system is illustrated in Fig. 1. The Fuzzy ARTMAP (FAM)[28] 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 using fuzzy rules (if–then). This information is used only during the first assembly operation; later this is enhanced by patterns that favour the assembly and whose inclusion is regulated by the Pattern-Motion Selection module. This module keeps track of the appropriateness of the F/T patterns to allow the FAM network to be retrained. If this is the case, the switch SW is closed and the corresponding pattern-action is provided to the FAM for on-line retraining. The selection criterion is given by expression (1), discussed in Section 3.2.1.

Future predictions will be based on this newly trained FAM network. The Automated Motion module is basically in charge of sending the incremental motion request to the robot controller and handling the communication with the Master Computer. External components to the NNC are the robot controller, the manipulator itself and the F/T sensor that provides the pattern information. The programs for the NNC were created using Visual C++ 6.0 and implemented in an 800 MHz Pentium III Computer.

3.2.1. Knowledge refinement during fine motion. There are potential overtraining problems associated with learning patterns on-line during fine motion, which are solved by the Pattern-Motion Selection module indicated in Fig. 1. The

https://doi.org/10.1017/S0263574706002839 Published online by Cambridge University Press

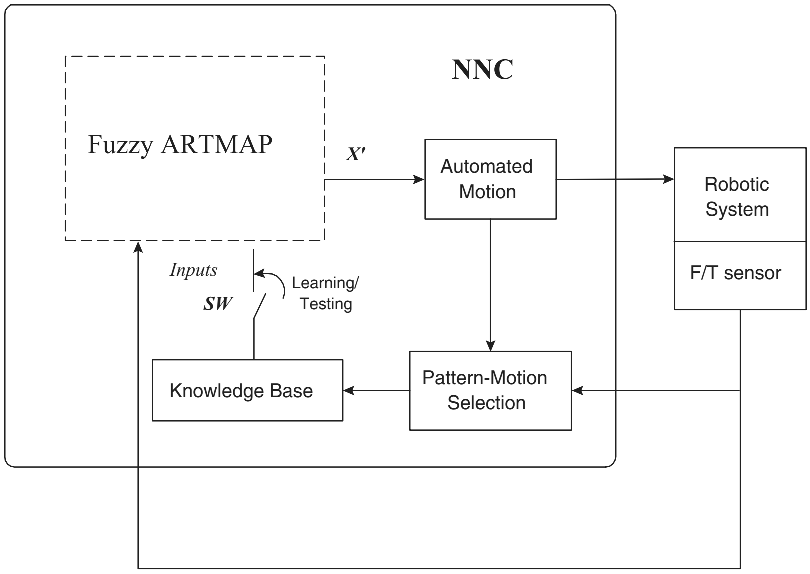

Fig. 1. System structure.

robot should continue to move 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-mentioned 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 in 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 that 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:

$$F_{\text{after}} < 0.1 \times F_{\text{before}}, \tag{1}$$

where F_{after} and F_{before} are merit figures calculated before and after the corrective motion is applied and are computed using the following equation as in Ahn's work[29]:

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

Expression (1) is generic for the experimental work reported in this paper (i.e. peg-in-hole operation of the components considering dimension and geometry as given in Section 4.3) meaning that if F_{after} is significantly less than F_{before}, then that pattern-action will be considered good to be included in the knowledge base. Experiments have shown that if this threshold value was set higher (i.e., $\geq 0.3 \times F_{\text{before}}$) the network became very sensitive and showed overtraining behaviour. It is also important to mention that in order to compare the robot's behaviour using different knowledge bases, the units to be used in Eq. (2) have to be the same (e.g. N and N · dm) to avoid the comparison of inconsistent data. A scale factor S has been included in Eq. (2) to allow the use of different units or size components. In our experiments, the scale factor was selected to be equal to 1. Should different units or size components be used, then the scale factor S has to be modified, hence the F values in Eq. (2) will be different and expression (1) would probably have to be reviewed in order to define the learning, recall and error recovery zones described later.

There are three possible situations: learning, recall and error recovery as illustrated in Fig. 2. During learning, values that are lower than the threshold value given by $0.1 \times F_{\text{before}}$

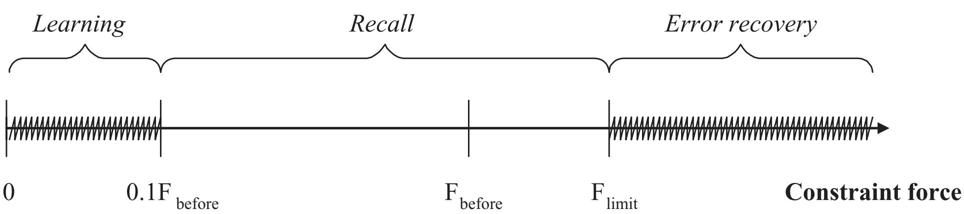

Fig. 2. Learning, recall and error recovery.

https://doi.org/10.1017/S0263574706002839 Published online by Cambridge University Press

are always stored in the knowledge base and learned on-line. Values higher than $0.1 \times F_{\text{before}}$ but lower than F_{limit} are used only for network recall – testing mode – and learning is not allowed. The third area, error recovery, is a situation where $F \geq F_{\text{limit}}$, as shown in Fig. 2. 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 (see Fig. 5). This recovery action will certainly diminish the contact forces and will satisfy the condition given by expression (1) 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_{after} and F_{before} will be very similar and again learning another pattern in the $Z-$ direction will be redundant. Both the situations were tested experimentally and it was revealed that an unstable situation might appear if further learning is allowed. After the pattern-action has satisfied expression (1) and the prediction direction is not in the Z direction, the pattern is allowed to be included in the new "expertise" of the robot, PKB, now the Enhanced Knowledge Base (EKB). The above procedure can be better understood with the flowchart of the NNC processing as shown in Fig. 3.

At the beginning of the operations, the following settings for the NNC are provided:

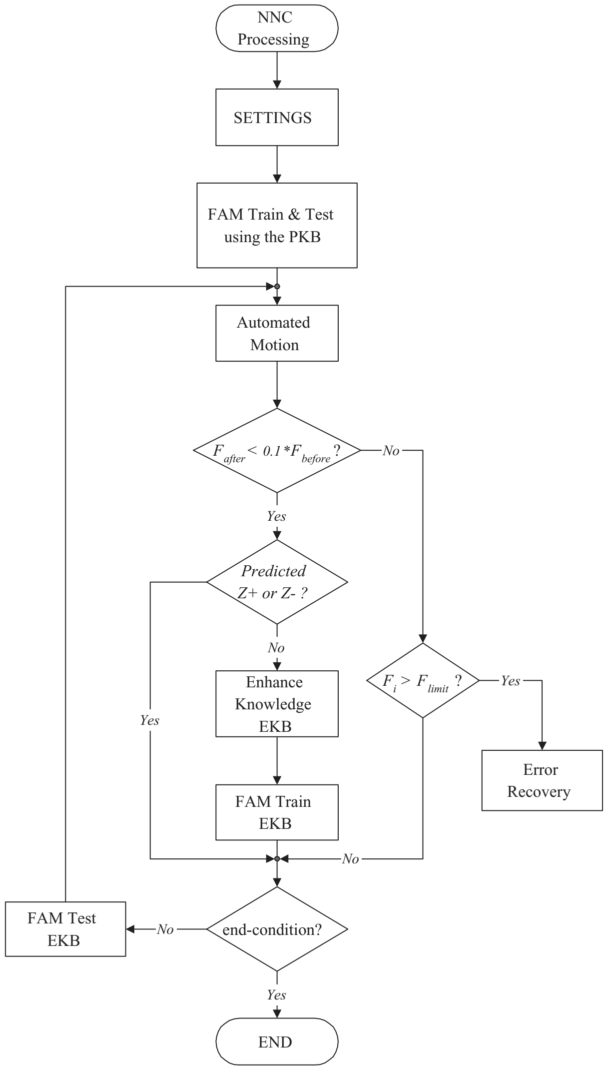

Fig. 3. Flowchart of the NNC processing.

- *System initialisation*, such as information about the type of coordinate system: this information is required by the Automated Motion module in order to move the manipulator either in Tool or World coordinates. Settings for the F/T sensor (translation and rotation in the sensor frame) and also threshold values to secure the operations (40 N for force and 20 N · dm for torque).
- *End-condition*: in this field, the number of incremental motions during a successful insertion is given (140 motions in the $Z-$ direction).
- *FuzzyARTMAP parameters*: information regarding learning rate, size of the input vectors (I_a and I_b), vigilance parameters, number of epochs, etc. are given at this time ($\rho_a = 0.2$ (base vigilance), $\rho_{\text{map}} = 0.9$, $\rho_b = 0.9$, I_a and $I_b = 6$, epochs $= 2$).

Threshold values for the F/T sensor and the end-condition are specific to the peg-in-hole task. In the case of the Fuzzy ARTMAP parameters, these values were selected to work in fast-learning mode.[28] By using a high-vigilance (ρ) value, the network was highly selective to discriminate between similar patterns. These parameters are generic and could be used for any other learning task that requires this learning feature.

4. RESULTS: ISSUES AND EXPERIMENTS

4.1. Workcell architecture

The workcell is formed basically by a 6 DOF KUKA KR15 industrial robot, KRC2 robot controller, KUKA Control Panel (KCP), PC Master Computer, JR3 F/T sensor attached to the robot's wrist, a ceiling mounted CCD camera and a conveyor belt as illustrated in Fig. 4. The main units of the robot system are the KRC2 controller and the robot arm itself. The KRC2 controller houses the components that control and power the robot arm. The Master Computer hosts the DSP-based F/T sensor card and also communicates with the robot controller at lower level via serial port using the Xon/Xoff protocol. The vision system uses an auxiliary computer – not shown – in which algorithms for POSE determination (orientation and location) reside. POSE information about the components on the conveyor belt is provided by the Auxiliary Computer to the Master Computer, which in turn issues proper motion commands to the KRC2 controller for component grasping. Once the part (male component) is held by the robot, then the vision system also determines the female location at the Master Assembly Block and sends the female centroid information to the Master Computer in order

https://doi.org/10.1017/S0263574706002839 Published online by Cambridge University Press

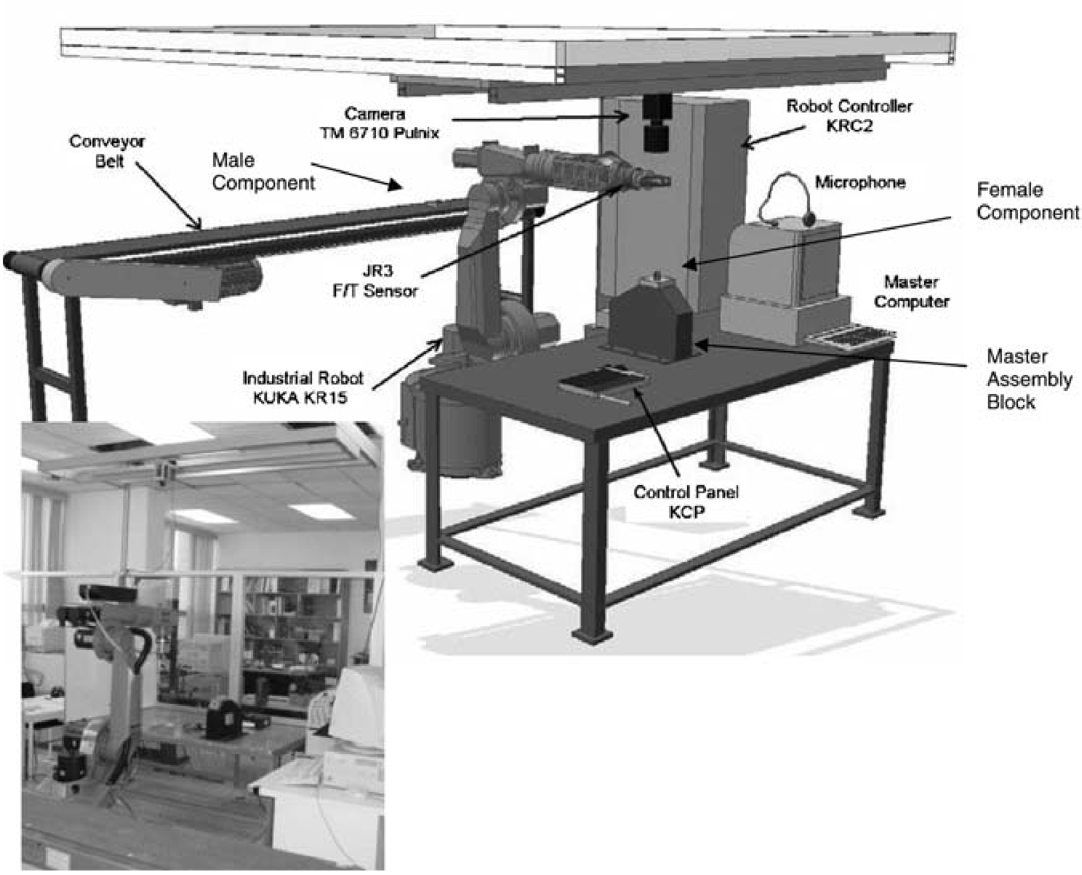

Fig. 4. System architecture.

to move the male component above the female component in readiness for assembly. The location of the female component is provided with an average error of 0.86 mm in both *X* and *Y* directions.

4.2. Knowledge acquisition

4.2.1. 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 – which is illustrated in Fig. 5. The GVN-PKB used for the experiments reported in this paper considered rotation around the *Z* axis and diagonal motions as illustrated in Fig. 6. The PKB values were obtained normalizing to the maximum experienced contact force during trials.

Using the above-mentioned GVN-PKB to start the learning of the assembly skill, it proved to be effective (see Section 4.3.1 for results); however, the robot still lacked the autonomy and it was realized that sometimes the robot did not use all the information given in the PKB and it was also noticed that a difference between the taught contact forces and the actual forces occurred during assembly so that an autonomously created PKB was needed in order to provide complete self-adaptive behaviour to the robot.

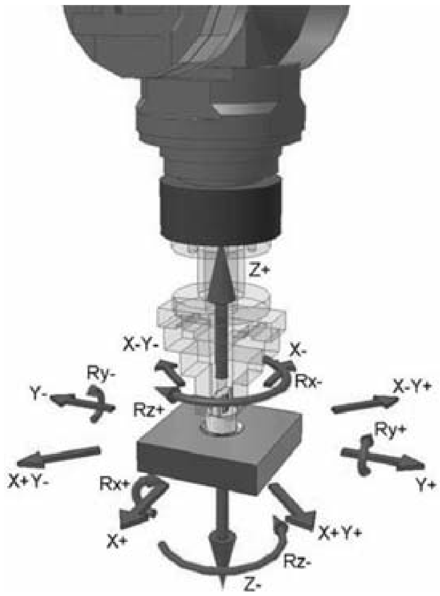

Fig. 5. Motion space.

4.2.2. Acquired-primitive knowledge base (ACQ-PKB). It was decided to embed a fuzzy logic mechanism to autonomously acquire any 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 contact is made the robot starts

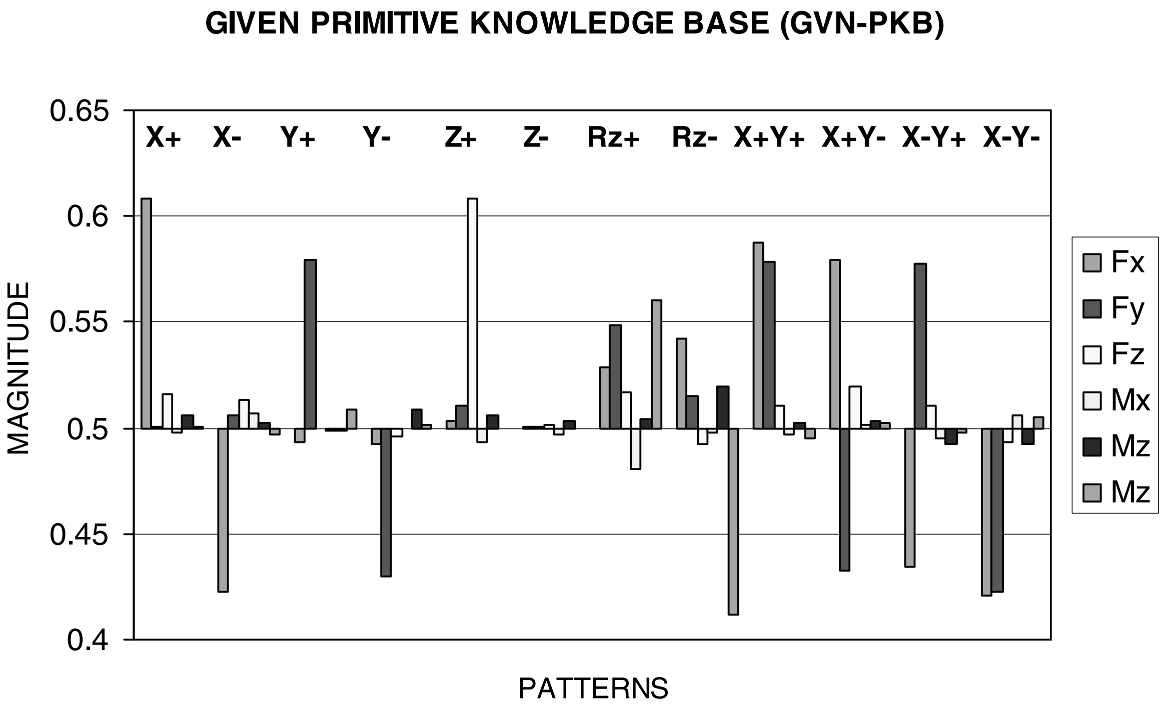

Fig. 6. Given-PKB (GVN-PKB).

acquiring information about the contact states following fuzzy rules and autonomously generates the corresponding motion commands to form the 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 on the magnitude of the moment and force values in the X and Y directions.

Fuzzy logic has proved to be useful to model many decision-taking processes in the presence of uncertainty or where no precise knowledge of the process exists in an attempt to formalize experience and empiric knowledge of the experts in a specific process. The initial knowledge for our fuzzy rules 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.

The following considerations apply for the generation of the fuzzy rules:

No. of linguistic values: 3 (negative, medium and positive)
No. of input variables: 6 (Fx, Fy, Fz, Mx, My, Mz)
Maximum no. of rules: $6^3 = 216$ (only 12 were used)

The membership functions are (stated as) shown in Fig. 7. Forces and moments have normalized values between 0 and 1. The normalization was *ad hoc* and considered the maximum experimental value for both, force and moment values. A value of 0.5 corresponds to zero, negative values are considered to be below 0.5 and positive values above 0.5. No belong functions were defined for the output, because our process does not include defuzzification in the output. The function limit values are chosen heuristically and according to previous experience in the assembly operation.

The fuzzification stage is performed with an algorithm that quantifies the membership value for similar triangles,

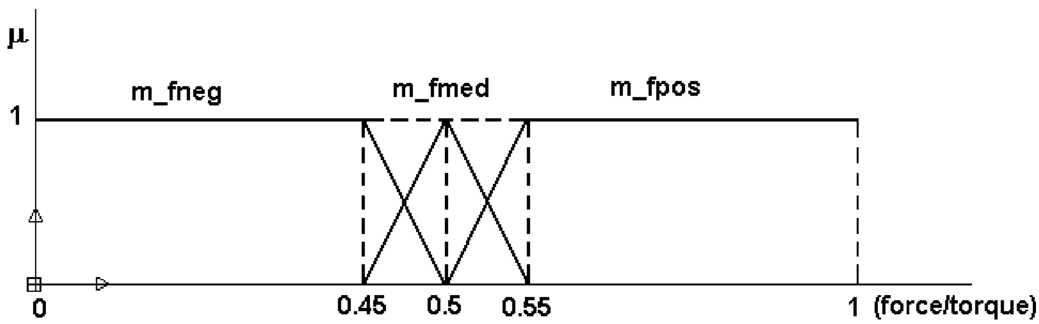

Fig. 7. Membership functions.

https://doi.org/10.1017/S0263574706002839 Published online by Cambridge University Press

for example the force in the X direction:

```
if (Fx < m_fneg) //Membership values for fuzzy collection of forces in X.
  {Fxneg = 1; Fxmed = 0; Fxpos = 0; }
else
  {if(Fx >= m_fneg && Fx < 0.5)
    {Fxneg = (0.5 − Fx)/(0.5 − m_fneg); Fxmed
    = (Fx − m_fneg)/(0.5 − m_fneg); Fxpos = 0; }
    else
      {if(Fx >= 0.5 && Fx <= m_fmed)
        {Fxmed = (m_fmed − Fx)/(m_fmed − 0.5);
        Fxpos = (Fx − 0.5)/(m_fmed − 0.5);
        Fxneg = 0; }
      else
        {if (Fx > m_fmed) {Fxpos = 1; Fxmed = 0; Fxpos = 0; }}
      }
  }
```

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

If (Fx is pos) and (Fy is med) and (Fz is pos) and (Mx is med) and (My is pos) and (Mz is med) then (Dir is X+)
If (Fx is neg) and (Fy is med) and (Fz is pos) and (Mx is med) and (My is neg) and (Mz is med) then (Dir is X−)
If (Fx is med) and (Fy is pos) and (Fz is pos) and (Mx is neg) and (My is med) and (Mz is med) then (Dir is Y+)
If (Fx is med) and (Fy is neg) and (Fz is pos) and (Mx is pos) and (My is med) and (Mz is med) then (Dir is Y−)
If (Fx is med) and (Fy is med) and (Fz is pos) and (Mx is med) and (My is med) and (Mz is med) then (Dir is Z+)
If (Fx is med) and (Fy is med) and (Fz is med) and (Mx is med) and (My is med) and (Mz is med) then (Dir is Z−)
If (Fx is med) and (Fy is med) and (Fz is pos) and (Mx is med) and (My is med) and (Mz is pos) then (Dir is Rz+)
If (Fx is med) and (Fy is med) and (Fz is pos) and (Mx is med) and (My is med) and (Mz is neg) then (Dir is Rz−)
If (Fx is pos) and (Fy is pos) and (Fz is pos) and (Mx is neg) and (My is pos) and (Mz is med) then (Dir is X+Y+)
If (Fx is pos) and (Fy is neg) and (Fz is pos) and (Mx is pos) and (My is pos) and (Mz is med) then (Dir is X+Y−)
If (Fx is neg) and (Fy is pos) and (Fz is pos) and (Mx is neg) and (My is neg) and (Mz is med) then (Dir is X−Y+)
If (Fx is neg) and (Fy is neg) and (Fz is pos) and (Mx is pos) and (My is neg) and (Mz is med) then (Dir is X−Y−)

For chamferless assembly, another knowledge base would have to be generated also using the above rules, but without considering force in the X and Y axis. 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,[30] and to obtain a conclusion, the maximum value for the fuzzy output response used was as follows:

```
Xpos = (Fxpos*Fymed*Fzpos*Mxmed*Mypos*Mzmed);
Xneg = (Fxneg*Fymed*Fzpos*Mxmed*Myneg*Mzmed);
Ypos = (Fxmed*Fypos*Fzpos*Mxneg*Mymed*Mzmed);
Yneg = (Fxmed*Fyneg*Fzpos*Mxpos*Mymed*Mzmed);
Zpos = (Fxmed*Fymed*Fzpos*Mxmed*Mymed*Mzmed);
Zneg = (Fxmed*Fymed*Fzmed*Mxmed*Mymed*Mzmed);
Rzpos = (Fxmed*Fymed*Fzpos*Mxmed*Mymed*Mzpos);
Rzneg = (Fxmed*Fymed*Fzpos*Mxmed*Mymed*Mzneg);
XposYpos = (Fxpos*Fypos*Fzpos*Mxneg*Mypos*Mzmed);
XposYneg = (Fxpos*Fyneg*Fzpos*Mxpos*Mypos*Mzmed);
XnegYpos = (Fxneg*Fypos*Fzpos*Mxneg*Myneg*Mzmed);
XnegYneg = (Fxneg*Fyneg*Fzpos*Mxpos*Myneg*Mzmed);
```

Once the algorithm values are 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 1–2 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. Results given in this paper considered only 12 patterns ($X+$, $X-$, $Y+$, $Y-$, $Z+$, $Z-$, $Rz+$, $Rz-$, $X+Y+$, $X+Y-$, $X-Y+$, $X-Y-$), omitting the rotations around the X and Y planes since only straight insertions were considered. The database generated with this procedure for the chamfered square peg insertion is shown in Fig. 8.

4.3. Compliant motion during peg-in-hole operations

Several tests were carried out to assess the compliant motion performance of the NNC using aluminium pegs with different cross-sectional geometry: circular, squared and radiused-square (termed radiused-square because it was a square peg with one corner rounded). These components are shown in Fig. 9(a). The diameter of the circular peg was 25 mm and the side of the square peg was also 25 mm. The dimensions of the nonsymmetric part, the radiused-square, were the same as the squared peg with one corner rounded to a radius of 12.5 mm. Clearances between pegs and mating pairs were 0.1 mm, chamfers were set at 45° with 5 mm width. The assembly was ended when three-fourths of the body of the peg was inside the hole. This represented 140 motion steps in the $-Z$ assembly direction. A typical assembly operation is shown in Fig. 9(b). 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 on the basis of the fact that the Fuzzy ARTMAP network was required to be as selective as possible to cluster all different patterns, which is achieved by having a high vigilance level for ρ_{map} and ρ_b; hence, this was the main criterion to select the vigilance and was not related to the task conditions (shape, offset errors). The value of ρ_a 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 (a detailed description of the Fuzzy ARTMAP architecture is given in ref. 28). In our experiments the values for the vigilance were as follows:

$\rho_a = 0.2$ (base vigilance)
$\rho_{map} = 0.9$
$\rho_b = 0.9$

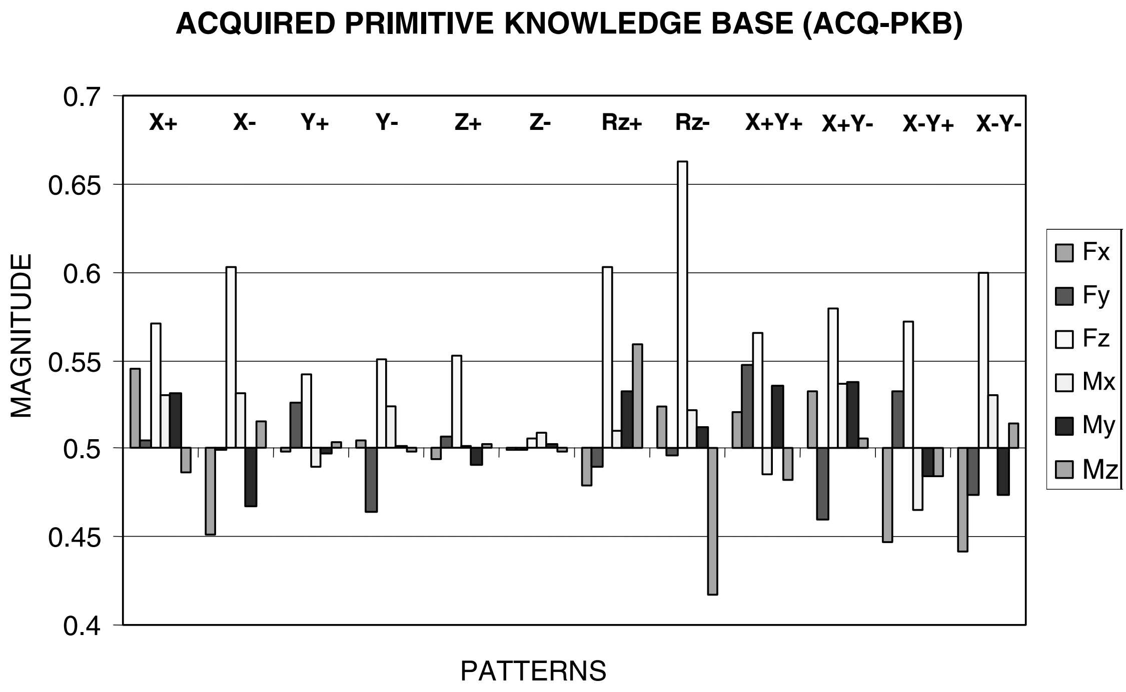

Fig. 8. ACQ-PKB.

The NNC has been tested in a number of assembly operations using circular, squared and radiused-square pegs employing an *a priori* knowledge base formed as described in Section 4.2.1. The NNC was indeed able to generalise its knowledge by assembling different component types. This includes the use of different part geometry, clearance and chamfered parts. Results also showed the robot's capability to acquire assembly skill and to improve its compliant motion through time, since assembly times are reduced and erratic motions do not recur. It was also recognised that in order to be completely self-adaptive, the robot had to acquire the skill autonomously.[19]

The work presented in this paper improves the early approach giving full robot autonomy by acquiring and using contact force information as needed. In the following sections results obtained from both, a GVN-PKB and an ACQ-PKB are described and the main differences explained.

4.3.1. Using the GVN-PKB. Typical results in a chamfered squared peg insertion using the GVN-PKB are summarised in Table I. 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 s.

4.3.2. Using the ACQ-PKB. By using the knowledge acquired from the squared peg insertion as explained in Section 4.2.2, the robot was also able to perform the assembly. For comparison purposes, several insertions using the same offset as before were carried out and the results are given in Table II.

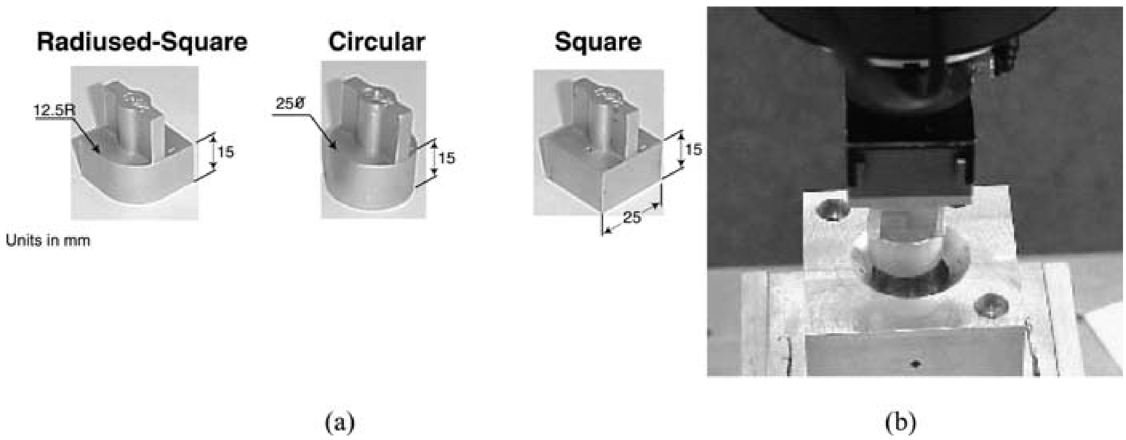

Fig. 9. (a) Assembly components; (b) peg-in-hole operation.

https://doi.org/10.1017/S0263574706002839 Published online by Cambridge University Press

Table I. Results using a GVN-PKB.

Insertion	Offset (δx, δy, δR_z) (mm, mm, °)	New patterns	Alignment motions	Total motions	Time (s)
1	(0.7, 0.8, 0.8)	0	23	173	47.08
2	(−0.8, 1.1, −0.8)	1	24	178	48.19
3	(−0.7, −0.5, 0.8)	2	65	213	57.78
4	(0.8, −0.9, −0.8)	0	20	160	43.41
5	(0.7, 0.8, −0.8)	1	28	174	47.11
6	(−0.8, 1.1, 0.8)	3	30	170	46.27
7	(−0.7, −0.5, −0.8)	2	21	171	46.30
8	(0.8, −0.9, 0.8)	0	17	157	42.58
9	(0.7, 0.8, 0.8)	0	18	158	42.92
10	(−0.8, 1.1, −0.8)	3	18	158	42.77
11	(−0.7, −0.5, 0.8)	4	31	171	46.55
12	(0.8, −0.9, −0.8)	0	19	159	43.08
13	(0.7, 0.8, −0.8)	0	68	210	56.98
14	(−0.8, 1.1, 0.8)	3	38	184	49.91
15	(−0.7, −0.5, −0.8)	0	21	161	43.66
16	(0.8, −0.9, 0.8)	0	32	172	46.72

4.3.3. Robot's dexterity – learning new patterns. From the results given in Tables I and II, it can be observed that the number of new patterns using the GVN-PKB was much higher (19) compared to 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 a few more examples that 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.

4.3.4. Robot's dexterity – constraint motion. 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 measures during experiments using the GVN-PKB and the

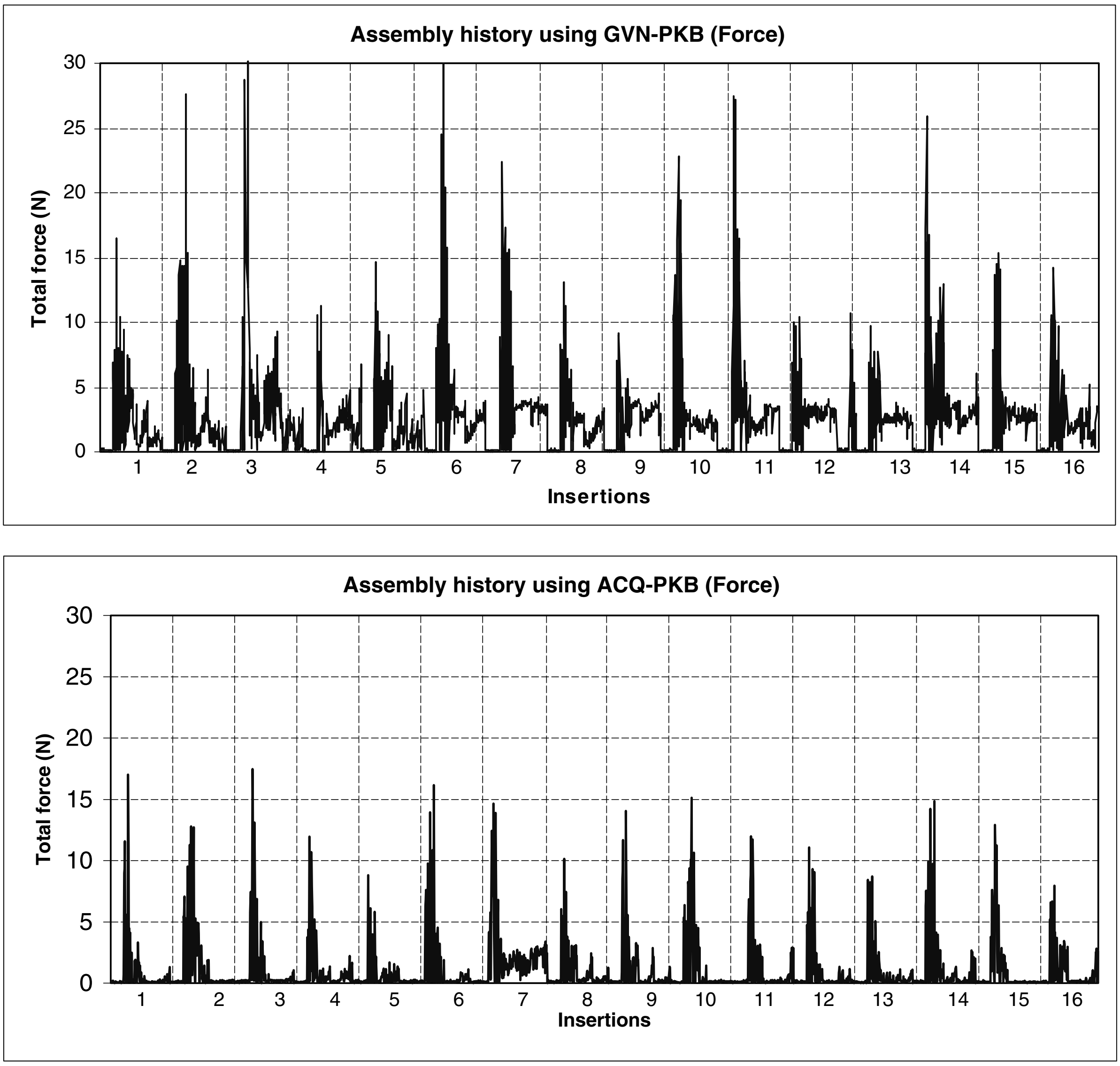

Fig. 10. Forces during square chamfered peg insertion.

https://doi.org/10.1017/S0263574706002839 Published online by Cambridge University Press

Table II. Results using an ACQ-PKB.

Insertion	Offset (δx, δy, δR_z) (mm, mm, °)	New patterns	Alignment motions	Total motions	Time (s)
1	(0.7, 0.8, 0.8)	0	26	166	44.53
2	(−0.8, 1.1, −0.8)	1	36	176	47.83
3	(−0.7, −0.5, 0.8)	0	22	162	43.55
4	(0.8, −0.9, −0.8)	0	25	165	44.56
5	(0.7, 0.8, −0.8)	0	20	160	43.14
6	(−0.8, 1.1, 0.8)	1	32	173	46.48
7	(−0.7, −0.5, −0.8)	0	26	168	45.56
8	(0.8, −0.9, 0.8)	0	22	162	43.50
9	(0.7, 0.8, 0.8)	0	27	167	44.80
10	(−0.8, 1.1, −0.8)	0	28	172	46.22
11	(−0.7, −0.5, 0.8)	1	19	159	42.78
12	(0.8, −0.9, −0.8)	0	25	173	46.59
13	(0.7, 0.8, −0.8)	0	20	162	43.62
14	(−0.8, 1.1, 0.8)	1	28	168	45.30
15	(−0.7, −0.5, −0.8)	1	22	162	43.94
16	(0.8, −0.9, 0.8)	0	20	160	42.94

ACQ-PKB were obtained using the following equations:

$$\text{Total force} = \sqrt{fx^2 + fy^2 + fz^2} \tag{3}$$

$$\text{Total torque} = \sqrt{mx^2 + my^2 + mz^2} \tag{4}$$

These values are shown in Figs. 10 and 11, respectively. From Figs. 10 and 11, it can be observed that when using the ACQ-PKB, the magnitudes of the forces and torques were significantly lower and in certain cases they were almost half the value compared to the value obtained in the same experiments when using the GVN-PKB.

4.3.5. Robot's dexterity – followed trajectory. The type of the followed path towards the insertion centre can be viewed as a robot's performance indicator. An optimum trajectory is a peg straight motion from the offset point to the insertion centre so that there is neither angular nor translational misalignment.

In order to assess this performance, the residual angular misalignment (Rz) was recorded through different insertions using the GVN-PKB and ACQ-PKB as shown in Fig. 12. It can be observed that when using the GVN-PKB the residual angular error was higher compared to the results when using the ACQ-PKB. In other words, using the acquired knowledge led to a better fit at the end of the task. These results compare favourably with the constraint forces appearing in Figs. 10 and 11. The pegs were assembled successfully in both cases; however, motion was more compliant when using information obtained directly from the contact states (ACQ-PKB).

The other performance indicator is the translational error measured from the starting offset point to the insertion centre.

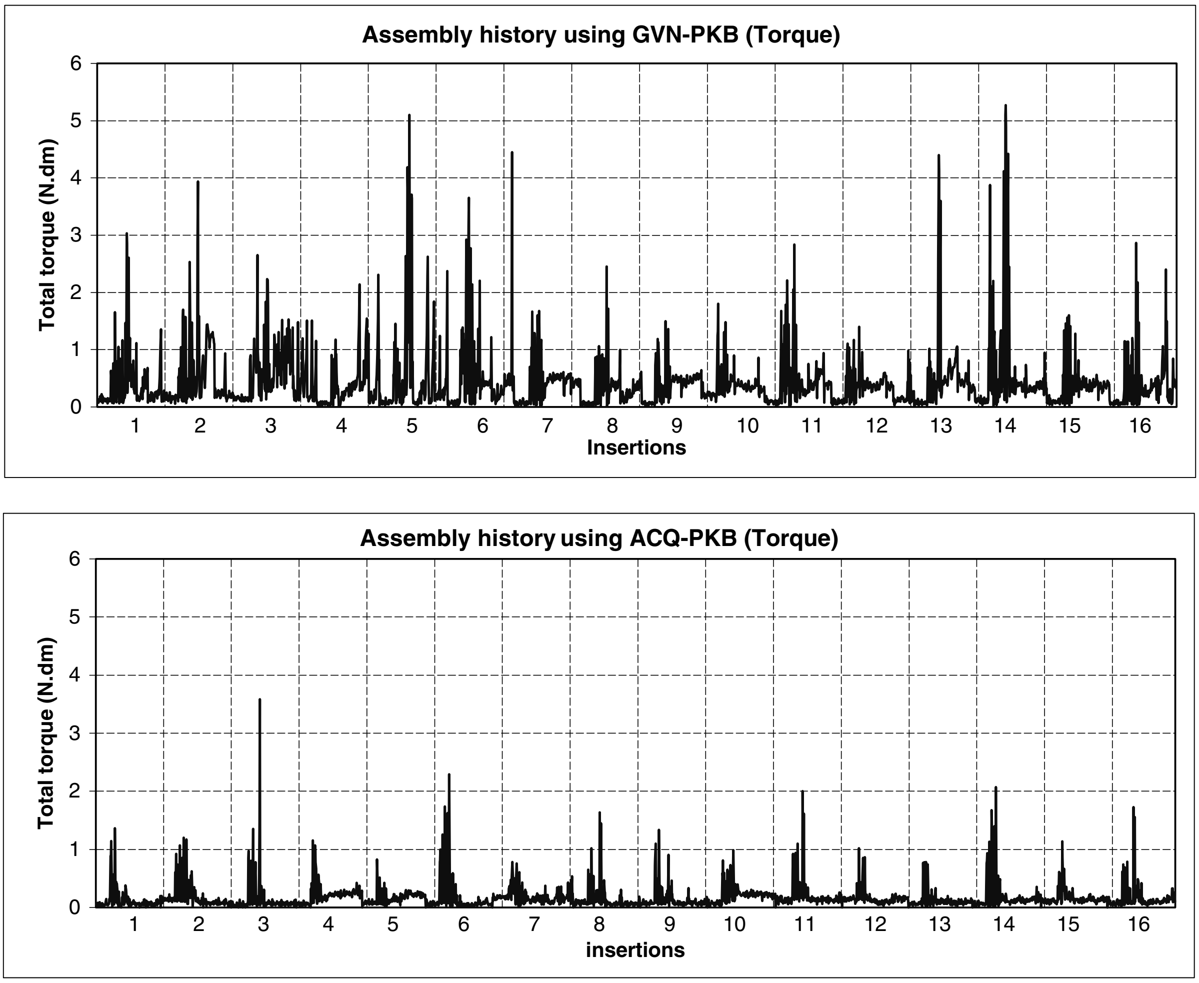

Fig. 11. Torque during square chamfered peg insertion.

https://doi.org/10.1017/S0263574706002839 Published online by Cambridge University Press

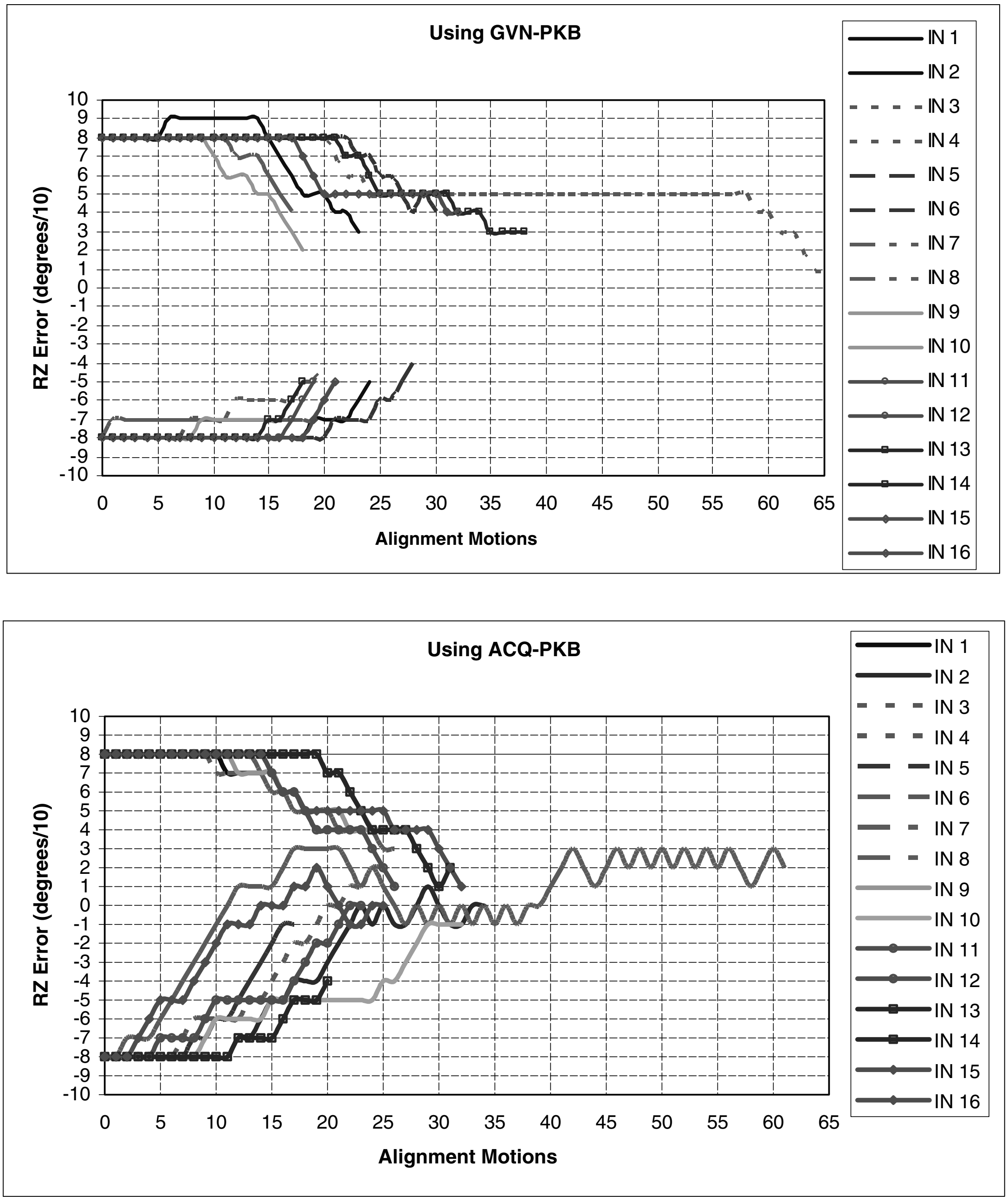

Fig. 12. Angular misalignment.

The translational error value is the absolute value of the sum of both the error values in each direction. These values were recorded and are illustrated in Fig. 13. From this graph it can be observed that in most cases the distance error was lower when using the ACQ-PKB.

Finally, another performance indicator is the absolute measurement of the followed trajectory. These values were also recorded and compared to the ideal trajectory (straight line) in all cases. The results are given in Fig. 14. It was observed that using the acquired knowledge led to better – shorter – trajectories during assembly except in four cases as indicated by the black arrows in the figure.

4.3.6. Generalisation using the ACQ-PKB. The generalisation capability of the NNC was also tested by assembling different components using the same ACQ-PKB. Results are provided in Table III. 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 capability enabled (L_{on}) and also with the learning capability disabled (L_{off}); that is, only using the initial ACQ-PKB. The assembly operation was always successful and in general it was faster when the learning state was enabled.

5. CONCLUSIONS AND FUTURE WORK

The robot has demonstrated its abilities not only to acquire the assembly skill but also to learn the operation from scratch. Initial knowledge is acquired from actual contact states using

https://doi.org/10.1017/S0263574706002839 Published online by Cambridge University Press

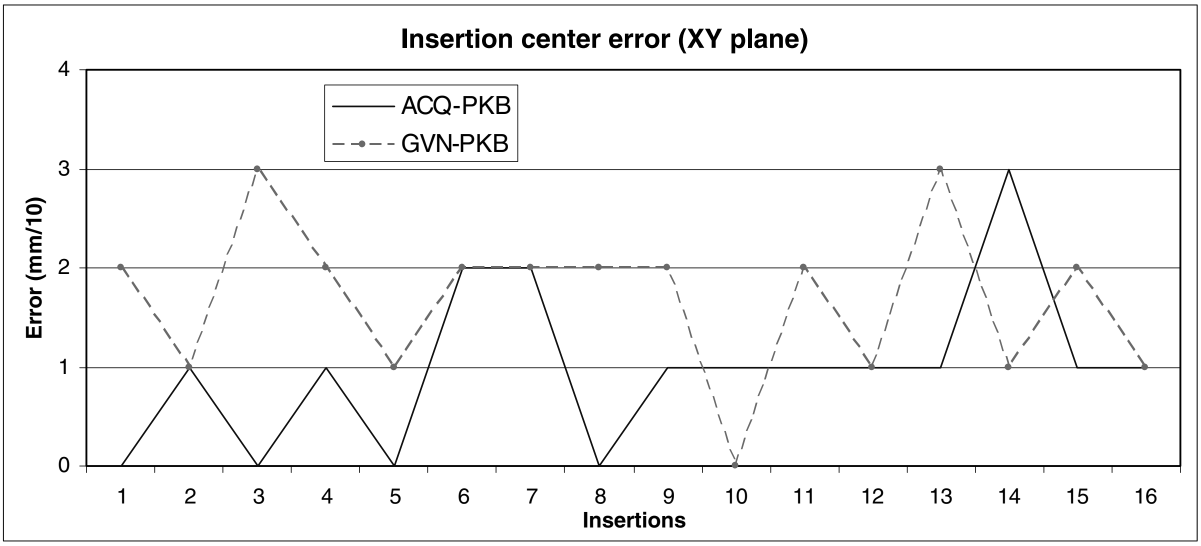

Fig. 13. Translation error in the *XY* plane.

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.

In both cases, using the ACQ-PKB and GVN-PKB, the robot achieved 100% success in performing the assembly task.

The compliant behaviour of the robot improved using the proposed approach; this can be appreciated by observing the magnitude of the merit figure F obtained from Eq. (2) as shown in Figs. 10 and 11. It can be seen that the magnitudes of forces and moments are lower when using the ACQ-PKB. Values are significantly lower, hence motions were more compliant in this case indicating that information acquired directly from the part geometry also allowed 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 found that given *a priori* knowledge (GVN-PKB) is fine, but contact information extracted directly from the operation itself provides the manipulator with better compliant motion behaviour.

During experiments, it was also noticed that angular and translational errors, measured between the starting position and the final assembly point, diminished when using the ACQ-PKB as was observed in Figs. 12 and 13. Furthermore, the absolute distance from the starting to the assembly final point was also reduced in most cases when using the ACQ-PKB.

Chamferless insertion has not been tested using an ACQ-PKB; ongoing work is looking at this possibility by using data from the camera system in order to train the robot autonomously. Here, the strategy would be to place the peg in contact with the female component, establish an offset and then generate the mapping moment values around the X and Y axis and the corresponding linear motions.

The approach was demonstrated under real-world operations using an industrial manipulator. Results from this work have envisaged further work in the area of multimodal

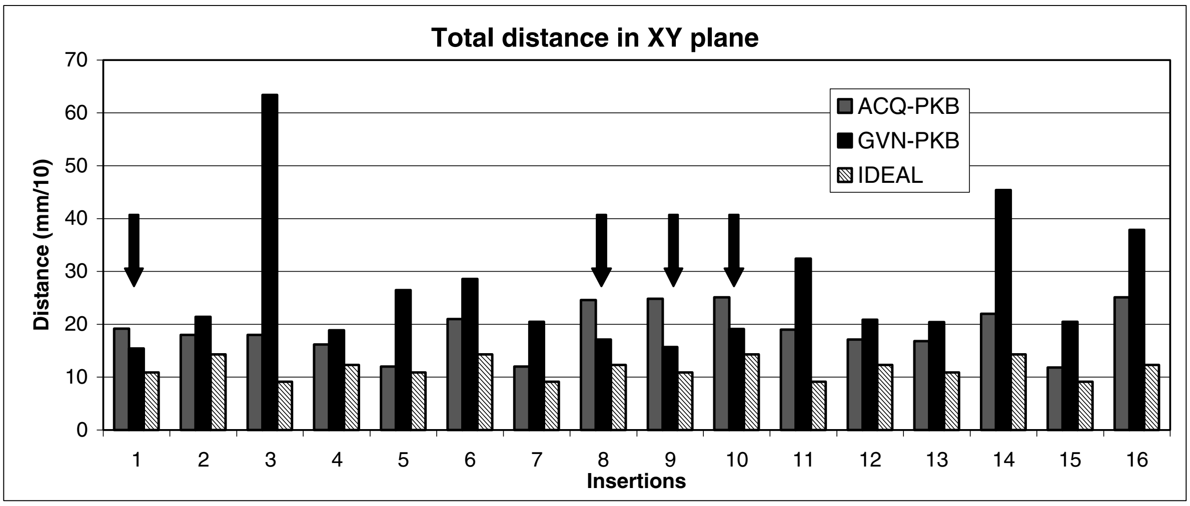

Fig. 14. Absolute distance in the *XY* plane.

https://doi.org/10.1017/S0263574706002839 Published online by Cambridge University Press

Table III. Results using an ACQ-PKB.

Insertion	Offset (d_x, d_y, d_{Rz}) (mm, mm, °)	Time (s) L_{on}	Time (s) L_{off}
Radius-square chamfered peg insertion			
1	(0.7, 0.8, 0.8)	45	48
2	(−0.8, 1.1, −0.8)	45	51
3	(−0.7, −0.5, 0.8)	43	47
4	(0.8. −0.9, −0.8)	50	54
5	(0.7, 0.8, −0.8)	44	44
6	(−0.8, 1.1, 0.8)	53	51
7	(−0.7, −0.5, −0.8)	54	55
8	(0.8, −0.9, 0.8)	50	49
9	(0.7, 0.8, 0.8)	46	46
10	(−0.8, 1.1, −0.8)	45	55
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
Circular chamfered peg insertion			
1	(0.7, 0.8, 0)	42	43
2	(−0.8, 1.1, 0)	41	41
3	(0.8. −0.9, 0)	40	42
4	(0.8, −0.9, 0)	41	41
5	(−0.8, 1.1, 0)	41	41
6	(0.8, −0.9, 0)	41	42
7	(1.4, 1.6, 0)	45	45
8	(1.6, −1.8, 0)	43	45
9	(1.4, 1.6, 0)	43	44
10	(−1.4, −1, 0)	42	43

data fusion.[31] We expect that data fusion from the F/T sensor and the vision system to result in an improved confidence for getting the contact information at the starting of the operation and also to provide important information such as chamfer presence, part geometry and pose information, which will be the input data to a hierarchical task level planner.

Acknowledgements

The authors thank the following organizations for supporting this research through various 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). C programming advice for the membership functions by Reyes Rios is very much appreciated.

References

1. D. Whitney and J. Nevis, "What is the remote center compliance (RCC) and what can it do?" *Proceedings of the 9th International Symposium on Industrial Robots*, Washington, DC, USA (1979) pp. 135–152.
2. S. Joo and F. Miyasaki, "Development of a variable RCC and its applications" *Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS 1998), Victoria, BC, Canada, **2**, pp. 1326–1332.
3. E. A. Erlbacher, *Force Control Basics* (PushCorp, Inc.) http://www.pushcorp.com/Tech%20Papers/Force-Control-Basics.pdf (accessed December 14, 2004).
4. M. T. Mason, "Compliant motion," In: *Robot Motion* (M. Brady *et al*., eds.) (MIT Press, Cambridge, (1983).
5. J. De Schutter and H. V. Brussel, "*Compliant robot motion I, a formalism for specifying compliant motion tasks*," *Int. J. Robot. Res.* **7**(4), 3–17 (1988).
6. T. Lozano-Perez, M. T. Mason and R. H. Taylor, "Automatic synthesis of fine motion strategies," *Int. J. Robot. Res.* **3**(1), 3–24 (1984).
7. I. Lopez-Juarez, M. Howarth and K. Sivayoganathan, "Robotics and skill acquisition," In: *Advances in Manufacturing Technology X* (A. Bramley, T. Mileham and G. Owen, eds.) (1996) pp. 166–170, ISBN 1 85790 031 6.
8. H. Asada, "Teaching and Learning of Compliance Using Neural Nets," *Proceedings of the IEEE International Conference on Robotics and Automation*, Cincinnati, Ohio, USA (1990) pp. 1237–1244.
9. E. Cervera and A. P. del Pobil, "Learning and Classification of Contact States in Robotic Assembly Tasks," *Proceedings of the 9th International Conference IEA/AIE*, Japan (1996) pp. 725–730.
10. E. Cervera and A. P. del Pobil, "Programming and Learning in Real-World Manipulation Tasks," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robot and Systems*, Albuquerque, NM, USA (1997) pp. 471–476.
11. V. Gullapalli, J. A. Franklin and H. Benbrahim, "Acquiring Robot Skills via Reinforcement Learning," *IEEE Control Syst.* 13–24 (1994).
12. M. Howarth, An Investigation of Task Level Programming for Robotic Assembly *Ph.D. Thesis* (The Nottingham Trent University, (1998).
13. E. Cervera and A. P. del Pobil, "Sensor-based learning for practical planning of fine motions in robotics," *Int. J. Inf. Sci.* **145**(1), 147–168 (2002) (special issue on intelligent learning and control of robotics and intelligent machines in unstructured environments).
14. V. Gullapalli, "Skilful control under uncertainty via direct reinforcement learning," *Robot. Auton. Syst.* 237–246 (1995).
15. M. Kaiser and M. R. Dillman, "Building elementary robot skills from human demonstration" *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota (April 1996) pp. 2700–2705.
16. M. Skubic and R. Volz, "Acquiring robust, force-based assembly skills from human demonstration," *IEEE Trans. Robot. Autom.* **16**(6), 772–781 (2000).
17. I. Lopez-Juarez, M. Howarth and K. Sivayoganathan, "An Adaptive Learning Approach to Control Contact Force in Assembly," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS 1998), Victoria, BC, Canada, **3**, pp. 1443–1448.
18. I. Lopez-Juarez and M. Howarth, "Learning Manipulative Skills with ART," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Takamatsu, Japan (IROS 2000) **1**, pp. 578–583.
19. I. Lopez-Juarez and M. Howarth, "Knowledge acquisition and learning in unstructured robotic assembly environments," *Int. J. Inf. Sci.* **145**(1), 89–111 (2002) (special issue on intelligent learning and control of robotics and intelligent machines in unstructured environments).
20. J. Xiao and L. Liu, "Contact States: Representation and Recognizability in the Presence of Uncertainties," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Victoria, BC, Canada (1998).
21. X. Ji and J. Xiao, "Automatic Generation of High-Level Contact State Space," *Proceedings of the International Conference on Robotics and Automation*, Detroit, Michigan, USA (ICRA 1999) pp. 238–244.
22. M. Skubic and R. Volz, "Identifying Contact Formations from Sensory Patterns and its Applicability to Robot Programming

by Demonstration" *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Osaka, Japan (1996).

23. G. G. Towell and J. W. Shavlik, "Knowledge-based artificial neural networks," *Artif. Intell.* **70**(1–2), 119–166 (1994).
24. R. S. Feldman, *Understanding Psychology*, 3rd ed. (McGraw-Hill, Inc., New York, (1993).
25. S. Grossberg, "Adaptive pattern classification and universal recoding II: feedback, expectation, olfaction and illusions," *Biol. Cybern.* **23**, 187–202 (1976).
26. G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Comput. Vis. Graph. Image Process.* 54–115 (1987).
27. G. A. Carpenter, S. Grossberg and J. H. Reynolds, "ARTMAP: supervised real-time learning and classification of nonstationary data by self-organizing neural network," *Neural Netw.* 565–588 (1991).
28. G. A. Carpenter, S. Grossberg, N. Markunzon, J. H. Reynolds and D. B. Rosen, "Fuzzy ARTMAP: a neural network architecture for incremental learning of analog multidimensional maps," *IEEE Trans. Neural Netw.* **3**(5), 698–713 (1992).
29. D. S. Ahn, H. S. Cho, K. I. Ide, F. Miyazaki and S. Arimoto, "Learning task strategies, in robotic assembly systems," *Robotica* **10**, 409–418 (1992).
30. D. Driankov, H. Hellendoorn and M. Reinfrank, *An Introduction to Fuzzy Control*, 2nd ed. (Springer Verlag, USA, 1996).
31. I. Lopez-Juarez, K. Ordaz-Hernandez, M. Pena-Cabrera, J. Corona-Castuera and R. Rios-Cabrera, "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.) (Springer Verlag, Berlin, (2005) pp. 1062–1072.