An Expert System for Flexible Robotic Assembly Under Uncertainty Based On Fuzzy Inference

S. G. Tzafestas, G. B. Stamou, S. Roumeliotis, N. K. Papadopoulos
Intelligent Robotics and Automation Laboratory
National Technical University of Athens(NTUA)
Zografou Campus 15773,Athens
GREECE

Abstract: One of the most important robotic applications in manufacturing is automated assembly. The design and implementation of a robotic assembly system is a costly and time consuming problem. Additionally, if techniques that assume a known environment and known constraints are used, the whole system must be redesigned when some of the assumptions are no longer valid. The purpose of this paper is to provide a method for a flexible intelligent assembly system using the hierarchical model. The focus is on the product model abstraction level and an expert system for assembly is designed based on fuzzy logic. This system called OPASSEQ provides the optimum assembly sequence that satisfies the given constraints and goals. An example illustrates how the system works.

1 Introduction

In recent years, an effort was made to automate the process of assembly using robots. The goal is to design a system that can assemble a certain product from its constituent parts, using computer-controlled robots. In the implementation of such an automated robotic assembly system, many recognition sub-problems are involved increasing the complexity of the problem. Therefore, the cost of robotic assembly systems is generally high. The usual need for minor or major changes, either in the product itself or in the operating environment, in conjunction with high cost, makes the need for flexibility obvious..

The ideal solution would be a system that can operate in any environment, for the assembly of any product, with no human interference. Under the current technological standards, such a system is impossible to implement. On the other hand, in several sub-problems sensible progress was made. Many low-level languages for the communication with robots and sensors were implemented. Furthermore high-level languages [6] - [7] have also been implemented, that allow to approach the problem from a higher abstraction level.

Another important sub-problem where good work has already been done, is the representation of knowledge. Two kinds of information must enter the knowledge base, the knowledge describing the environment and the knowledge describing the assembly process. Either the same or different methods can be used to represent these two kinds of information. If the same method is used (as in [4]) there are advantages (all possible assembly sequences are derived from the symbolic representation of the environment) as well as disadvantages. A major disadvantage is that the assembly process cannot be divided into sub-assembly problems of sub-products. This problem (among others) is dealt with in [2] where the assembly graph is used.

Finally, there are other important sub-problems (also referred in the current bibliography) like that of converting the natural language, used by the user, to a symbolic representation of the knowledge (i.e. the assembly graph) as discussed in [5], or that of reducing the overall assembly cost by evaluating all the productive factors as discussed in [3] .

An excellent way of reducing the complexity taking advantage of the existing solutions in sub-problems, is to use an hierarchical model [1], [8]. A general hierarchical model for flexible robotic assembly systems was proposed in [9]. This model (shown in Figure 1) defines five different levels of abstraction. The system is divided into several subsystems. Each subsystem belongs to a specific hierarchical level. The subsystems exchange data in a specific way; data flow top to bottom. A subsystem of one level receives its input from the subsystem of the higher level and transmits its output to the subsystem of the lower level. The only case in which data move from lower to higher level, is when an error occurs. The modules implementing the subsystems are independent. As long as a predefined data exchange protocol is preserved, the modules can be altered or replaced. Notice that the human interference is restricted only to the highest level of abstraction.

The system presented here is an implementation of the "Expert System Determining the Optimum Assembly Sequence" module.

Selecting the assembly sequence is usually assigned to experts since there is no specific algorithm to follow but rather a set of general rules to take into consideration. To automate the selection is more difficult because it requires an evaluation of the variables that cannot be measured.

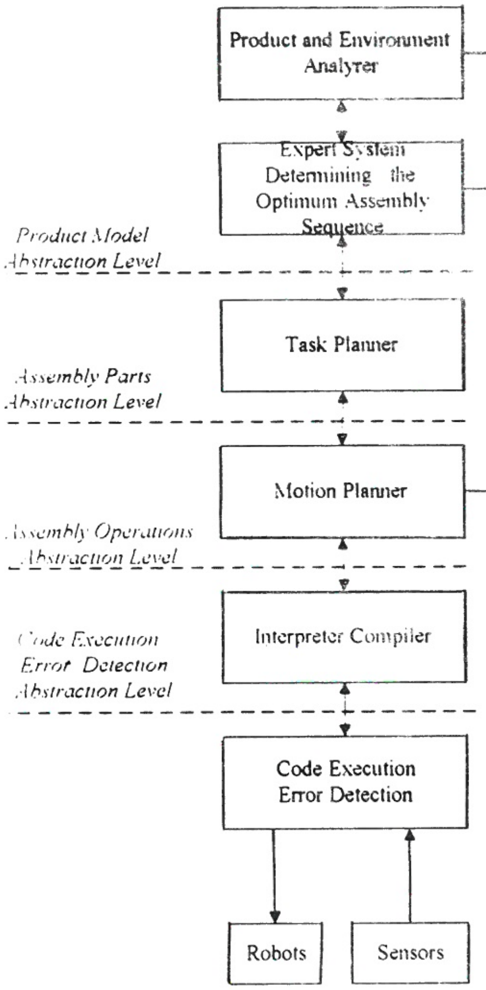

Figure 1

The above two reasons make the use of a fuzzy-logic-based expert system the ideal solution.

The system operates off-line (the assembly sequence is selected for once) and interacts with humans. Therefore speed was not a point during the design of the system, but special effort has been invested in making the user interface as efficient as possible. The user's skill varies from minimum computer operation experience, demanding simplicity and assistance, to programming knowledge, demanding maximum control over the system. The implemented system, is called OPASSEQ (**O**ptimum **As**sembly **Seq**uence) and aims at satisfying both requirements.

2 System Analysis

OPASSEQ is a Windows based application for IBM Compatible Computers. The system is divided into two parts. The first, i.e. the user interface, is responsible for letting the user enter the product model. This part of the system is the only that the user sees. The working environment is similar to that of every MDI (Multi Document Interface) Windows application, giving access to three different windows corresponding to the following functions:

- entering the product's parts
- entering the assembly graph
- entering the variables to be used for determining the optimum assembly sequence.

The second part is a set of *dynamic link libraries* (DLL) that implements the fuzzy-logic-based subsystem for determining the optimum assembly sequence. This part is activated while the variables are entered in the third window mentioned above. The selection of the optimum assembly sequence is made as follows.

First the part to be the base of the product during the assembly process is selected, then having a prefix of the assembly sequence, the next part to be assembled is selected, until no parts are left. Each of these two steps (selecting the base and selecting the next part) is taken using a different set of variables for input and a different set of rules: both steps generate a priority for each candidate part. The part with the maximum priority selected. The variables and the rules for these two steps are discussed later in this Section.

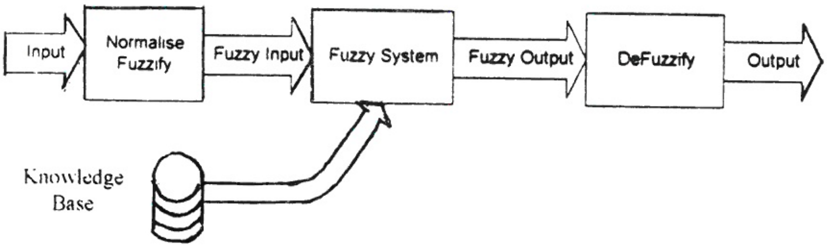

Figure 2

Beside their differences, the two steps are both based on the same fuzzy-logic-based system scheme shown in Figure 2.

Fuzzification, fuzzy reasoning and defuzzification procedures are done within the DLL subsystem. If the input needs to be normalised, this is done within the first part (user-interface) of the system. There are two kinds of variables, those for which the user is asked to enter a value representing a measurement and those for which the user is asked to enter a value representing an evaluation. For example "mass" is a variable representing a measurement, but "potentiality to stabilise" is a variable representing an evaluation. In both cases the user does not have to be precise, but in the second case the user is restricted to giving a value between 0 and 12. In the first case the values entered by the user are linearly normalised within the range 0 and 12.

Let us now present the variables and the rules used for the two steps of the selection process (selecting the base and selecting the next part). For each variable there is a corresponding set of fuzzy-logic rules. The rules have the form:

IF A V1 THEN B V2

where A is an input variable, B is an output variable and V1 and V2 are fuzzy sets of the universe of discourse U={0 1 2 3 4 5 6 7 8 9 10 11 12}. This means that all input and output variables are sets of the same universe of discourse. The fuzzy sets defined for the universe of discourse U are shown in Figure 3 using the abbreviations VL= Very Low, QL=Quite Low, L= Low, M=Medium, H= High, QH= Quite High, VH=Very High.

In the rest of this Section a short analysis of the used variables is made. Either in the selection of the base part or in the selection of the next part to assemble a large (high) value of the variable might encourage or discourage the selection of the part. The first case is noted with [VH→VH] and the second with [VH→VL] during the analysis of the variables.

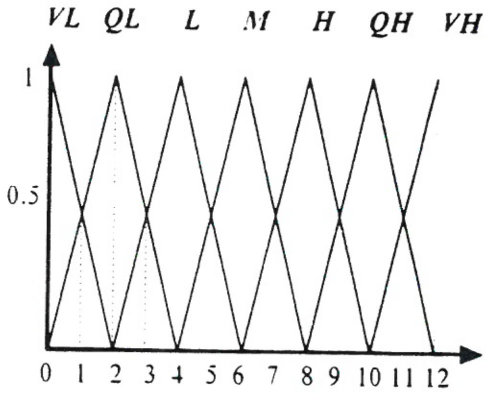

Figure 3

An example of the first case is the "n.i.e." (the number of items enclosed) variable, for which the set of rules is:

IF n.i.e. VL THEN p.b.p. VL

IF n.i.e. QL THEN p.b.p. QL

IF n.i.e. L THEN p.b.p. L

IF n.i.e. M THEN p.b.p. M

IF n.i.e. H THEN p.b.p. H

IF n.i.e. QH THEN p.b.p. QH

IF n.i.e. VH THEN p.b.p. VH

An example of the second case is the d.c.a.d. (degrees of change in assembly direction) variable for which the set of rules is:

IF d.c.a.d. VL THEN p.n.p. VH
IF d.c.a.d. QL THEN p.n.p. QH
IF d.c.a.d. L THEN p.n.p. H
IF d.c.a.d. M THEN p.n.p. M
IF d.c.a.d. H THEN p.n.p. L
IF d.c.a.d. QH THEN p.n.p. QL
IF d.c.a.d. VH THEN p.n.p. VL

The abbreviations p.b.p. and p.n.p. stand for "probability to be base part" and "probability to be next part" respectively. If the variable represents a measurement it is denoted by [M] and if it represents an evaluation it is denoted by [L]. In the analysis that follows, for each variable there is also the name, the abbreviation and a short remark on why a specific variable and set of rules are used.

For the selection of the base part, nine variables are used and the output is the priority-to-be-base-part (p.b.p.).

- *n.i.e.* (number of parts enclosed) [M] [VH → VH].

If a part encloses many other parts it can be treated as a container of a sub-assembly. It is preferable to select such a part as the base part. To do so allows many other parts to be placed in it before assembling the rest of the parts.

- *n.i.s.* (number of parts surrounding it) [M] [VH → VH].

It is preferable to select as the base part a part which many other parts are connected to. If such a part (part A) is not selected as base part and if one (part B) adjacent to it is selected as base part, then the robotic arm will have to track longer paths in order to place many parts not on the base part (part B) but on another (part A). We want to avoid such a choice.

- *n.ph.c.* (number of physical contacts) [M] [VH → VH].

It is preferable to select, as base part, a part with many physical contacts with other parts, because to do so allows the robotic arm to assemble a large number of parts within a specific area (around the base part).

- *d.c.m.* (distance of centre of mass) [M] [VH→ VL].

It is preferable to select as base part the part for which the distance between its centre of mass and the centre of mass of the final product is a minimum one. The smaller the distance is, the smaller is the possibility of a great transposition of centre of mass to occur during the assembly. Such a transposition might cause lack of balance and termination of the assembly process.

- *mass* [M] [VH → VH].

A heavy part is difficult to handle, i.e. difficult for the robotic arm to hold it, move it to the right position and rotate it to right orientation in order to be assembled. Therefore more energy and time are required to assemble a heavy part than to use it as a base part.

- *volume* [M] [VH → VH].

A part with large volume is difficult to handle, i.e. to avoid collisions while moving and rotating it and while other parts have already been assembled. Selecting such a part for base part saves energy and time.

- *n.a.d.* (number of assembly directions) [M] [VH → VL].

Assembling is easier when done in a single direction. Not selecting as base part a part that is assembled in many directions, facilitates the job of the motion planner.

- *p.s.* (potentiality to stabilise) [E] [VH → VH].

It is very important that the base part can be well stabilised. Lack of stability may lead to termination of the assembly process or even to damaging the product.

- *p.v.a.* (potentiality of vertical assembly) [E] [VH → VH].

The robotic assembly devices work better (increased accuracy, simplest motion planning, less energy consumption, etc.) when assembling in the vertical direction.

For the selection of the next part, four variables are used and the output is the "priority-to-be-next-part" (p.n.p.).

- *d.c.a.d.* (degrees of change in assembly direction) [M] [VH → VL].

Any change in the assembly direction requires re-calculation of the new direction from the path planner. This variable and the corresponding set of rules plan to minimise the total number of changes of direction in the assembly process.

- *t.c.m.* (transposition of centre of mass) [M] [VH → VL].

Great transposition of the centre of mass may result in lack of balance and eventually termination of the assembly process or even damaging the product.

- *h.p.a.*(hinders other parts to be assembled) [E] [VH → VL].

Selecting as next part a part that hinders other parts to be assembled results in longer track for the robotic arm, more complicated calculations for the motion planner and therefore increased time and energy cost.

- *f.p.a.* (facilitates other parts to be assembled) [E] [VH → VH].

For example if a part helps in the stabilisation of other parts, it should be assembled before them.

It should be noted here that the variables used for the selection of the base part are part-specific and can be treated as part of the product model. On the contrary the values of the variables used for the selection of the next part have to do not only with the specific part but also with the parts assembled so far.

3 Simulation Example

In this Section we present the results of using OPASSEQ for the selection of the optimum assembly sequence of the gear-shaft shown in Figure 4.

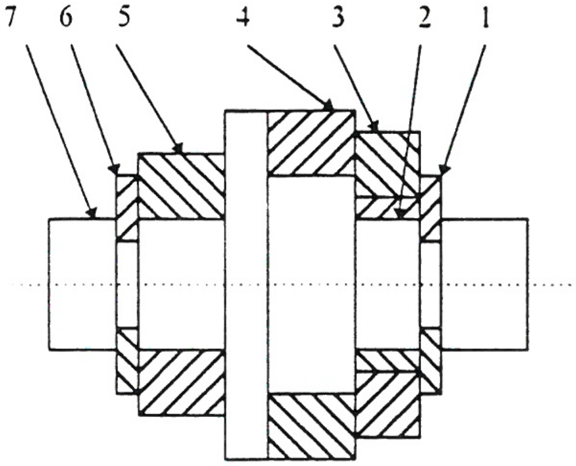

Figure 4

The seven parts the products consist of are:

1. Large Locker Ring
2. Quill
3. Medium Gear
4. Large Gear
5. Small Gear
6. Small Locker Ring
7. Shaft

Figure 5 shows the assembly graph of the product. Each node is a set of boxes. Each box corresponds to a part numbered from left to right and from top to bottom. A black box means that the corresponding part is assembled and a white box means that the corresponding part has not been assembled yet.

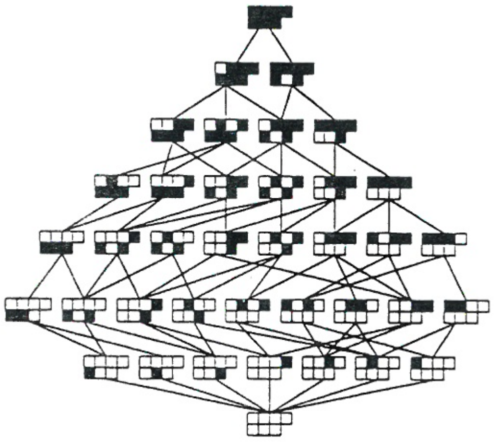

Figure 5

Table 1 summarises the values entered for the selection of the base part and the resulting priorities for each part. As one can see from Table 1 as well as from Figure 5 all the parts are candidates to be base part. In Table 1 and in those that follow, each column corresponds to a candidate part and each row to an input variable. The first row holds the numbers of the parts and the first column the names of the variables. The last row holds the resulting priorities, with bold digits for the selected part. For those variables that represent a measurement, the value entered is followed by a slash character ('/') and the normalised within the range 0 - 12 value.

Table 1

	6	1	4	3	2	7	5
n.i.e	1/6	1/6	1/6	2/12	1/6	0/0	1/6
n.i.s.	0/0	0/0	0/0	0/0	1/2	6/12	0/0
n.ph.c	2/0	3/3	2/0	4/6	4/6	6/12	2/0
d.c.m	24/12	24/12	0/0	16/8	16/8	0/0	18/9
mass	10/0	20/2	50/8	40/6	15/1	70/12	30/4
volume	10/0	30/2	60/5	50/4	20/1	130/12	40/3
n.a.d	1/0	1/0	2/12	2/12	2/12	2/12	2/12
p.s.	2	2	8	8	2	12	8
p.v.a	12	12	6	8	2	12	10
priority	3.97	4.16	5.65	5.58	3.1	**8.96**	4.52

The part selected as base part is "Shaft". Tables 2, 3, 4 and 5 that follow summarise the input and output variables for the selection of the 2nd, 3rd, 4th and 5th parts in the assembly sequence. These Tables are formatted using the same conventions as in Table 1, except that now there are only four, instead of nine, rows with input variables. At each step, the part with the maximum priority is selected as next part. In some cases there are two different parts having equal, yet maximum priority. The part selected (noted by bold digits) depends on the position of the corresponding node in the assembly graph.

Table 2

	4	5	2
d.c.a.d.	180/12	0/0	180/12
t.c.m.	0/0	2/12	1/6
h.p.a.	0	0	0
f.p.a.	0	0	0
priority	**6**	6	4.9

Table 3

	5	2	3
d.c.a.d.	180/12	0/0	0/0
t.c.m.	2/8	0/0	3/12
h.p.a.	0	0	0
f.p.a.	0	0	0
priority	4.13	8.66	6

Table 4

	3	5
d.c.a.d.	0/0	180/12
t.c.m	2/12	1/0
h.p.a.	0	0
f.p.a.	0	0
priority	6	6

Table 5

	5	1
d.c.a.d.	180/12	0/0
t.c.m.	1/12	0/0
h.p.a.	0	0
f.p.a.	0	0
priority	3.33	**8.66**

After selecting the 5th part, according to the assembly graph, there is only one choice for the 6th and 7th parts. The complete optimum assembly sequence as selected by OPASSEQ is:

- SHAFT
- LARGE_GEAR
- QUILL
- MEDIUM_GEAR
- LARGE_LOCKER_RING
- SMALL_GEAR
- SMALL_LOCKER_RING

4 Conclusions

Applying fuzzy-logic-based expert system techniques in designing a flexible system for automatic robotic assembly, effectively helps solve complicated problems and extends the degree of automation and flexibility corresponding to it based on classical methods. In addition, it limits the interference of the human factor lowering the error probability and response time. Finally taking advantage of the capabilities of modern operating systems and programming languages the system can be easily implemented supported by a user-friendly graphical interface.

REFERENCES

1. SARIDIS, N., **Intelligent Robotic Control**, IEEE TRANS. AUTOM. CONTROL, AC 29 (4), 1983, pp.547-557.
2. BINGER, G. and FLEMMING, V., **Graph Oriented Simulation of Assembly Process**, SYSTEM MODEL. SIMUL., 6 (10), ACADEMIC - VERLAG, Berlin, 1989, pp.781-786.
3. JUNG, J. Y. and BILLATOS, S. B., **An Expert System for Assembly Based on Axiomatic Design Principles**, JOURNAL OF INTELLIGENT AND ROBOTIC SYSTEMS, Vol.8, No.2, 1993, pp.245-265.
4. JAIN, A. and DONATH, M., **Knowledge Representation Systems for Assembly Using Robots,** Techn. Report, University of Minessota, Minneapolis, 1992.
5. KAPITANOVSKY, A. and MAIMON, D., **Robot Programming Systems for Assembly: Conceptual Graph-Based Approach**, J. INTELL. ROBOTIC SYSTEMS, Vol.8, No.1, 1993.

6. **SHIMANO, E., GESCHKE, C.C.,** SPALDING, C.H. III, GOLDMAN, R. and SCARBOROUGH, D.W., **AIM: A Task Level Control System for Assembly,** ADEPT TECHNOLOGY INC., IFS Ltd., March 1987.

7. GOODWIN, J. and SANDEWALL, E., University of Linkoping, Sweden & A. Thorne , ABB Corporate Research, Sweden, **ARAMIS: A Robot and Manufacturing Instruction System,** 1988.

8. ALMGREN, R., **On Knowledge Based 8. Planning and Programming Systems for Flexible Automatic Assembly,** LINKOPING STUDIES IN SCIENCE AND TECHNOLOGY, Thesis No.176, May 1989.

9. TZAFESTAS, S.G. and STAMOU, G.B., **A Flexible Robotic Assembly Knowledge - Based System Using Fuzzy Reasoning,** Proc. Intl. Conf. on Recent Advances in Mechatronics, Istanbul, Turkey, August 1995, pp. 684-694.