

● Paper

## OPERATION SPECIALISTS FOR AUTOMATIC PROGRAMMING AND MONITORING OF ROBOTIC ASSEMBLY CELLS

L. BASAÑEZ, C. TORRAS, J. ILARI and A. SANFELIU

Instituto de Cibernética (UPC-CSIC), Barcelona, Spain

A system allowing both the automatic programming of a robotic cell and the monitoring of assembly task execution is described. The system consists of a *coordinator*, a set of *specialists* and a *task supervisor*. The specialists are each associated with a particular subtask within the assembly and have both off-line and on-line functions. The three specialists which have been incorporated into the system are grasping, displacement and insertion. The *coordinator* works from a task-level description of the assembly to be performed and generates a cell-centered plan based on an ordered call to the *system specialists*. Once the assembly operation has been planned, it is the responsibility of the *supervisor* to monitor the correct execution of the assembly task by handling multisensory integration, correcting local errors and calling—when needed—the on-line functions of the several specialists. The overall operation of the system is illustrated by means of an assembly task example.

### INTRODUCTION

The application of robots in mechanical assembly tasks is a field of growing interest. Nevertheless, several open problems in the programming and execution phases of these tasks currently limit their practical industrial use. This is clear when considering that even the simplest assembly tasks require the coordinated action of different object-oriented planning systems, each of them being associated with a given elementary assembly operation.<sup>4</sup> Grasping,<sup>17</sup> displacement<sup>18</sup> and insertion<sup>11</sup> constitute three examples of such elementary operations arising in most assembly tasks. Each one of these elemental planning systems has been independently studied by many research teams.<sup>9,10,15</sup> The results obtained so far seem to indicate the extreme complexity of this kind of apparently simple operation.

In this paper we introduce an *automated assembly planning system* intended to effectively integrate in a single system three geometric-planning specialists. On the one hand, the aim is to obtain a system able to properly develop a given task-level plan by means of the *coordinated action* of the three specialists. On the other hand, the proposed system is expected to tackle a given assembly task more efficiently than what would result by only spreading among the specialists the constraints introduced by each of them. The proposed system incorporates an on-line supervisory module which makes use of adequate sensory agents

in order to ensure the execution of the planned task despite unexpected events or errors.

### SYSTEM OVERVIEW

The goal of the system is to allow both the automatic programming of a robotic cell and the monitoring of assembly task execution, taking into consideration the uncertainties in the environment.<sup>3</sup>

The system consists of three modules: coordinator, supervisor and specialists. The *specialists* are submodules with off-line and on-line functions related with the planning and monitoring of the kind of operations of their speciality.

Three basic operations on the workpieces in an assembly task have been considered: picking, movement and placement. For each of these operations, a corresponding specialist has been designed. They are: the *grasping* specialist, the *displacement* specialist and the *insertion* specialist.

The input of the system is an object-centered plan generated by an strategic planner or by a human designer. The cell-centered planning is the responsibility of the *coordinator*, which controls the off-line activity of the specialists. It entrusts each kind of operation in the assembly sequence to one of the specialists, which has the responsibility of generating the detailed sequence of actions necessary to accomplish the operation. The coordinator interacts and arbitrates between the specialists, guiding the detailed planning developed by them.

For this function, the coordinator and the specialists make use of the object model and of the robotic workcell features contained in a knowledge base.

In the execution phase, each specialist is called again to perform the operations of its speciality. The

The original version of this paper was presented at the 2nd International Symposium on Robotics and Manufacturing (ISRAM), Albuquerque, New Mexico, 16-18 November 1988. The published proceedings of this meeting may be ordered from: CAD Laboratory for Systems/Robotics, EECE Dept., UNM, Albuquerque, NM 87131, U.S.A.

control of the on-line activity of the specialists is now accomplished by the *supervisor*, according to the plan previously generated. The supervisor is also the manager of the sensors of the cell and has the capacity to deal with some unexpected events or changes in the assembly environment.

## SYSTEM DESCRIPTION

### *The system coordinator*

The *system coordinator* works from a task-level description of the assembly to be performed. Its input consists of an *object-centered plan* (Fig. 1), that is, a description of the task to be performed in terms of relationships between the objects involved in the assembly. This object-centered plan does not take into account the particular features of the assembly cell being considered. This permits the definition of the assembly to be performed without having to provide a thorough description of it and disregarding the particular environment where the assembly will take place.

From the object-centered plan, the system coordinator computes a *cell-centered plan*, that is, an assembly plan incorporating the constraints derived from the particular features of the cell where the assembly task is expected to take place. Obviously, it is possible that the system coordinator concludes that it is not possible to execute the object-centered plan in the assembly cell considered (due, for instance, to reaching limitations of the robot included in the cell) and, thus, that a new object-centered plan has to be proposed either by the user or by a higher-level planning system.

To carry out its task the system coordinator has three operation specialists: a *grasping*, a *displacement* and an *insertion specialist*. These are three planning systems, each of them devoted to a particular subtask within the context of assembly planning. It is the

system coordinator's responsibility to take advantage of the information it gathers from these specialists in order both to achieve a cell-centered plan and to shorten the overall planning time. This is done by adequately propagating and combining the constraints imposed by each one of the several operation specialists. These constraints are derived both from the features of the task, and from the *sensory needs* of the several specialists along the task. The system coordinator's job cannot be reduced, thus, to a simple sequential calling of these specialists, but it involves an appropriate combination of the restrictions imposed by each of them. This is clear if we consider that to execute a pick-and-place operation, for instance, the job of the insertion specialist may impose handling constraints heavily restricting the universe of solutions of the grasping specialist. Similarly, the solution adopted by the grasping specialist may affect the solution path proposed by the path planning specialist. Thus, the solutions proposed by the different operation specialists must be adequately combined by the system coordinator in order to:

- guarantee that a solution leading to task accomplishment will be found if such a solution exists, and,
- take advantage of the several restrictions imposed by each of them to reduce the search universe of the other ones, thus saving search time.

This constraint combination job is undertaken according to a *highest-commitment first basis*: first, the several specialists let the system supervisor know the set of constraints that they will feasibly introduce as a consequence of the features of the task they must accomplish. The system coordinator, then, calls first the specialist imposing the strongest constraints. From the work of this specialist several restrictions

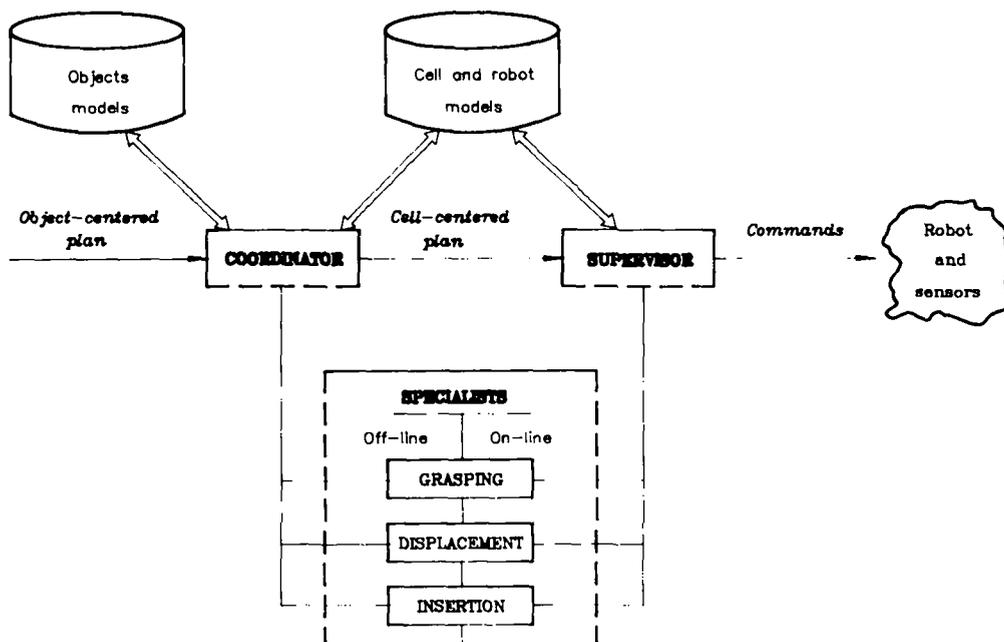


Fig. 1. System structure.

will be derived, with which the other specialists will have to work. Among these specialists, the one imposing most constraints on the task is called afterwards, further reducing the possible solutions of the remaining ones, the cycle ending with the *least committed* specialist.

If it is not possible to combine the constraints introduced by the different specialists, the system coordinator will be unable to find a solution. In this case it will have to try to slightly modify the task in order to make the existence of such a solution possible (for instance, introducing an intermediate position within the gripper from which to reorient the piece handled by the robot). If a solution is still not found, a new task-level description of the assembly will have to be supplied.

#### *The grasping specialist*

*Off-line operation.* From a description of the assembly operation to be performed, in terms of the homogeneous coordinate transformation between the two mating parts together with the approach direction, the off-line *grasping specialist* generates a list of grasping sites of the workpiece to be assembled, ordered according to a measure of their qualities (stability, tolerance to uncertainties, etc.). This measure is based on the criteria proposed by Wolter *et al.*<sup>17</sup> A gripper with two parallel planar fingers is assumed.

The generation process starts by determining the portion of the workpiece surface that will remain accessible after the assembly operation has been completed. This is done by performing some specific subtraction operations on the boundary-based representation of objects provided by the CAD database. Next, a layered search for certain types of grasping sites is carried out on this surface, guided by functions that evaluate the following aspects: face parallelism, distance between faces, and amount of overlap in the projection of one face upon the other. The approach extends that of Laugier<sup>7</sup> in using evaluation functions instead of Boolean filters, and also generalizes to 3D some of the criteria used in 2D by Peshkin and Sanderson.<sup>13</sup>

The position and orientation of the gripper relative to the grasping site is determined through a configuration space (C-space) approach. The C-space of the gripper is three-dimensional, since motion is constrained to the plane between the two fingers. C-obstacle boundaries consist of two types of faces: those arising from *contacts* of the grasped workpiece with either the lateral faces of fingers or the interior of the palm, and those originated by the limiting situation of only *ID overlap* between the gripping surface of a finger and the grasped surface. The problem reduces to that of finding space for a rectangle in a non-convex polygonal environment.

For each grasp, proximity sensor strategies to precisely locate the site are included, together with tactile sensor strategies to detect slippage or rotation of the

workpiece within the gripper. The expected tactile images are automatically generated and recovery procedures to be applied in the case of obtaining some distorted images corresponding to the most common anomalous situations are also included for each site.

*On-line operation.* The output from the off-line process, adequately filtered by the coordinator, constitutes the input to the on-line *grasping specialist*. This specialist can be viewed as a set of procedures that are configured at run-time according to the information computed off-line and the data acquired through the sensors.

A preliminary description of the two grasping specialists, which are at present being implemented, is provided in Ref. 1.

#### *The displacement specialist*

*Off-line operation.* The work of the off-line *displacement specialist* consists of finding a path satisfying the constraints derived from (a) the description of the motion to be undertaken, (b) the existence of obstacles in the workspace, (c) different requirements defined by other specialists, and (d) robot kinematics. All these requirements are supplied to the path planning specialist by the system coordinator. Until now dynamic aspects have not been considered.

Taking into account these constraints, the off-line path planner computes an adequate solution path by means of a search process in *discrete configuration space*. Working with a discretized C-space greatly simplifies the search process which can be reduced to a graph search procedure. As a counterpart, it also makes the algorithm *non-complete* as well as *non-exact*. However, its degree of non-completeness and non-exactness can be precisely tuned by adjusting C-space resolution. This makes the search procedure particularly appropriate for gross motion planning. For fine motion planning, algorithms based on an analytical description of configuration space obstacles are better suited.<sup>2</sup>

The search process in discrete C-space is undertaken by making use of heuristics which are intended to prune the search tree in order to save search time. On the current implementation of this specialist, the proposed heuristics are based on a free-space model which is a generalization to the 3D case of a free-space model previously used for 2D path planning<sup>5,6</sup> (see Fig. 2).

The constraints imposed by these heuristics which require working with a skyscraper universe suggest the convenience of developing in the future more general and powerful methods of cutting the search time.

*On-line operation.* The *displacement specialist* bases its on-line work on (a) the solution path computed off-line, (b) the information gathered from sensory elements, and (c) a set of error recovery procedures.

Clearly, in a perfectly controlled environment, on-

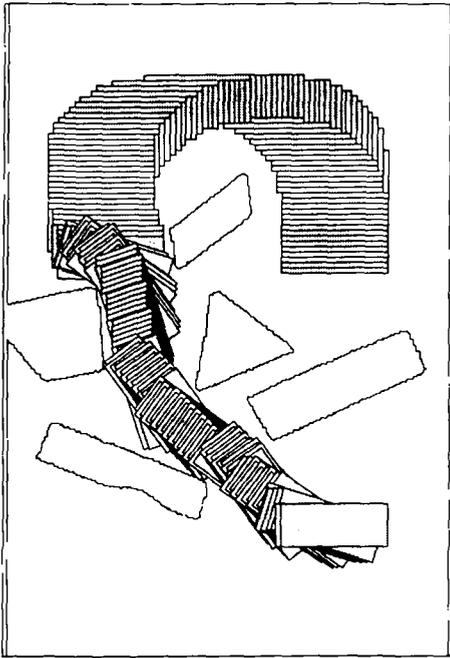


Fig. 2. 2D output of the path planner.

line path-planning should not be necessary. However, real environments are prone to unexpected events and errors. Under real circumstances the solution path computed off-line thus has a great chance of constituting the right solution to the wrong problem. In other words, it is possible that the real initial and final robot configurations do not coincide with the initial and final configurations assumed in the off-line path-planning process. This does not lead, however, to the need to discard the work done off-line and to begin on-line from zero. The solution path obtained off-line can be very effectively used to guide the search for a solution path to the on-line find-path problem. One way to do it is by using configuration space search heuristics based on the knowledge of an approximate or partial C-space solution path. This kind of heuristics has already been used successfully in 2D path planning,<sup>5,6</sup> its use being straightforward in the 3D case.

Once the new solution path has been obtained, the on-line displacement specialist will have to interact with the *system supervisor* in order to verify the correct following of the path. This will be done through the use of infrared proximity sensors to detect unforeseen obstacles as well as force sensors to detect unforeseen collisions. If either some unforeseen obstacle or collision is detected, the supervisor will ask the local path planner to propose local error recovery policies in order to try to *recover* the original path. If none of the local error recovery procedures succeeds, the system supervisor will try to undertake an update of the world model both from vision and infrared sensor readings. Then the system supervisor will call the on-line displacement once more, in order to propose a new solution path and the overall cycle starts again till the goal position is reached or else evidence is gathered that no solution path exists.

### The insertion specialist

**Off-line operation.** The *insertion specialist* is responsible for the assembly operations having the goal of putting together the parts of the final assembled device.

As with the other specialists, it has off-line and on-line functions. The off-line function is to determine the successive position and force set points of the robot servocontrollers, as well as to change conditions for these set points. This specialist must also provide proper test points and alternative sequences to handle deviations at execution time.

The off-line insertion specialist takes from the assembly database the geometrical characteristics of the objects involved in the insertion and the elastic and friction properties of these objects, to generate a position force configuration space (PF-space).<sup>16</sup>

This space is a  $2n$ -dimensional extension of the  $n$ -dimensional C-space<sup>8</sup> obtained by attaching to the C-space points the static reaction forces produced by the contact of the piece in the gripper with other objects.

The insertion specialist takes into account three sources of uncertainty: geometric tolerances in the piece dimensions, inaccuracy in piece position and inaccuracy in force measurement. Combining these uncertainty sources the *worst case uncertainty values* in both position and force parameters are obtained and, then, the set of PF-space points representing possible contact situation in the presence of uncertainty (UCS) is constructed.

The UCS is partitioned into disjoint PF-states which are represented as nodes of a *contiguity graph* (CGraph), whose links connect the contiguous states.

The next step is to determine the *state transition operations*, defined as the commands for the robot control system which produce the pass from a PF-state to another contiguous one. The form of these operators depends on the robot position/force control schema. In our system, a damping control is used and, therefore, the operators are commanded velocities.

Finally, according to given criteria (e.g. minimum number of PF-states in the sequence, minimum PF-states with non-zero force, etc.), a path going from the initial PF-state to that representing the inserted position is selected. The sequence of operators in the selected path will be the plan generated by the insertion specialist, and the sequence of states in this path will provide the test points for the execution phase.

**On-line operation.** The on-line function of the *insertion specialist* is to execute the plan generated by the off-line part. When the specialist is called by the supervisor, it begins reading the position and force in order to verify the state of the robot system. If, as expected, this state matches the initial one, the specialist sends to the robot servocontrollers the desired velocity and force values given by the first operator in the plan. Then, the specialist will monitor the position

and force values until a new state is detected. The specialist verifies if the planned second state has been reached. In this case, the second operator becomes active and the process will continue in an analog form until the insertion has been completed.

If, after applying an operator, the robotic system reaches an adjacent state different from the desired one, the specialist will make use of the CGraph to determine the new path to the goal state from the new situation.

If, in the insertion process, two consecutive states are not adjacent, something wrong has happened and the specialist will transfer the responsibility for handling the anomalous situation to the supervisor.

A mechanism has been designed to reduce the state uncertainty, by propagating the information supplied by the sensors.

#### *The system supervisor*

The objective of the *supervisor* is to distribute and coordinate the assembly operations, to handle multi-sensory integration, and to correct local errors if this can be done without resorting to a global replanning of the task. The first issue implies the need to distribute the operations planned by the three specialists and to coordinate them with the resources allocated, in order to reach the goal while taking into account time and physical constraints. The distribution and coordination of the operations is performed by a *scheduler* which activates the on-line specialists depending on the sequence of the plan and the information of the sensors. A preliminary version can be found in Ref. 14.

The preliminary version of the scheduler was designed as a blackboard control structure, where the control takes the decision of activating some actions depending on the previous tasks and the current conditions. The primitive control structure was based on an agenda where the actions and the conditions were reported. The control uses rules over the agenda to know which is the new action to be fired. If more than one action can be applied, there is a metacontrol which handles this situation. The new control structure which is being built consists of a similar structure, but it will contain some intermediate subcontrols to handle local error recovery and restricted replanning.

The goal of the *multisensory integration module* is two-fold: first to integrate the information of several sensors for each specific task, and second, to fuse the information of several sensors which are getting the same information (for example the distance measure) in order to increase the degree of certainty of the sensor information map.

The *multisensory integration module* is being used in the integration and fusion processes. The integration process is being applied in the grasping process where the torque and force and the tactile sensor are getting complementary information about the workpiece to manipulate. In this case, the data obtained from both sensors is used for the operations of insertion and

translation. The fusion process is being applied for improving the accuracy of the obtained depth data, between the vision sensor (a CCD TV camera) and infrared sensors. Because the camera uses  $256 \times 256$  pixels and the work field is about  $80 \times 60$  cm, the maximum resolution is about 0.31 cm/pixel, which is not accurate enough for doing precise insertion operations. In order to solve this problem we use the camera information to get the approximate location of the assembly module, and then we obtain more precise data with the infrared sensor and the model information.

The third issue is to correct errors in the execution of a task, which has been divided into two subtasks; (a) to correct on-line local errors which require few actions and on-line procedures; and (b) to correct local errors which do not need off-line replanning of the subtask. Both types of errors require three submodules: (1) an evaluation module which evaluates the error taking into account the subtask domain, the used device and the statistics of this type of error; (2) a replanning module which locally modifies the plan in order to overcome the problem; and (3) a local supervisor module which focuses on the correct execution of this subtask, and which is aware of the new sensor data to detect information not previously considered.

The supervisor also governs a distributed network of *intelligent sensors*. These sensors have some knowledge of the task and some strategies to overcome trivial problems. The *multisensory integration module* consists of several submodules which act as experts on the two aforementioned topics.

The structure of the three supervisor modules is being studied as an integration of symbolic knowledge, mainly based on production systems and frames, and procedural knowledge (algorithms), where the statistical, the geometric, the physical, and *ad hoc* knowledge could be integrated to take into account the principal matters of the tasks. An environment which support the structure of the supervisor has already been built as a Multiexpert system called MESYAK.<sup>12</sup>

A preliminary version of the *multisensory integration module* can be seen in Fig. 3. This structure is being implemented on the MESYAK environment. The *multisensory integration module* consists of two submodules, "low cost perception process" and "complex perception process". These two modules take direct decisions in normal situations, but they send the information to the other two if there is something abnormal.

#### SYSTEM OPERATION

The assembly of the components of an oil pump will illustrate the steps of the system based on specialists. Figure 4 shows the oil pump decomposed into its elements and Fig. 5a and b shows the oil pump which briefly consists of two gears (2, 4) which are engaged to pump the oil, and three frames (5, 6, 7), the front (6),

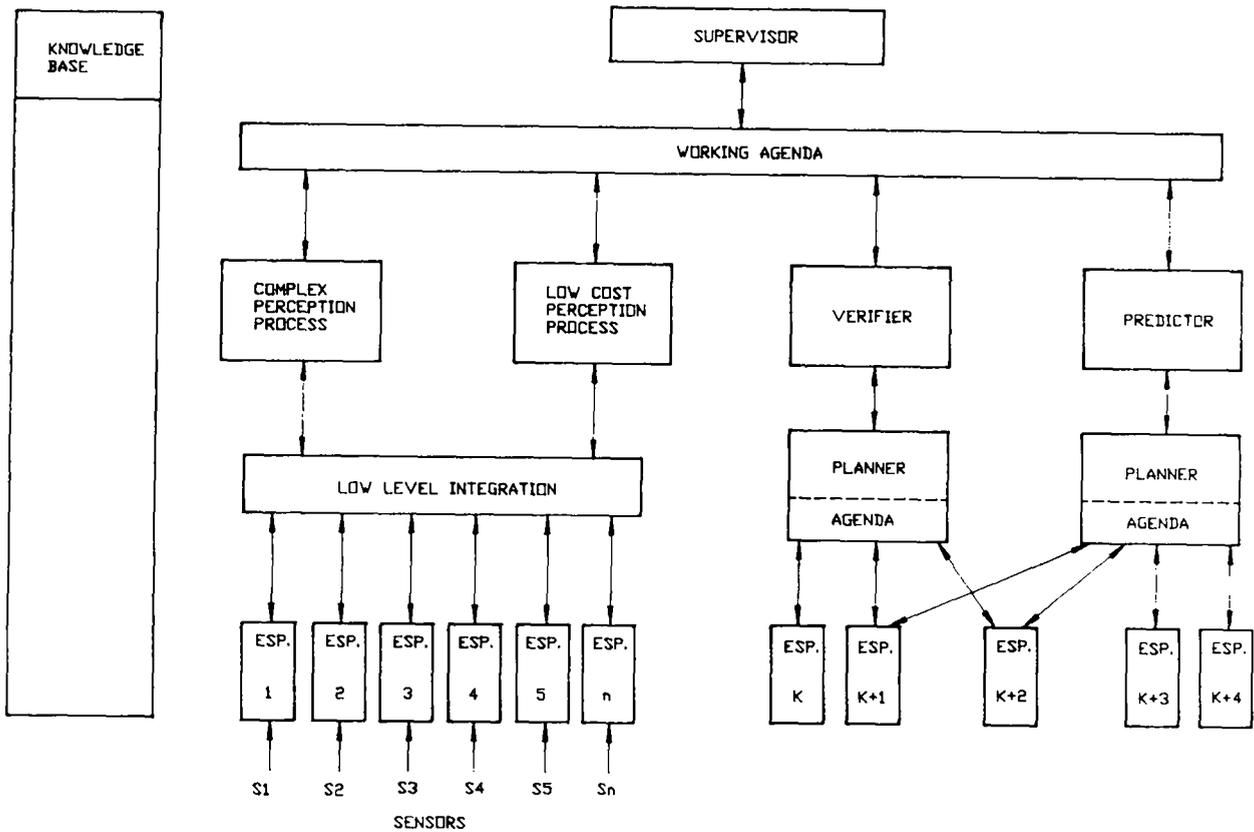


Fig. 3. Diagram of the multisensory integration module.

the middle (5) and the back (7) of the oil pump. Both gears (1, 3) are assembled in an axle bar (2, 4) which includes two axial bearings (12, 13) and (14, 15), one in the front and the other in the back. The motor moving gear (2) is longer than the slave gear and has one additional oil sealed socket (8). The frames are joined together by two pins with setscrews (10, 11). In order to simplify the assembly sequence we consider that the bearings are already inserted in the back and the front frames, as well as the oil sealed socket to the front

frame, and the two oil seals (9) to both the back and front frames. Moreover, we also consider that the gears are inserted into the axle bars.

The assembly process consists of six steps: (a) to leave the front frame in a specified location; (b) to insert the middle frame; (c) to insert the motor moving gear; (d) to insert the slave gear; (e) to insert the back frame; and (f) to insert the two pins and screw the setscrews. In this assembly sequence, several types of insertions are required, for example, the insertion of an

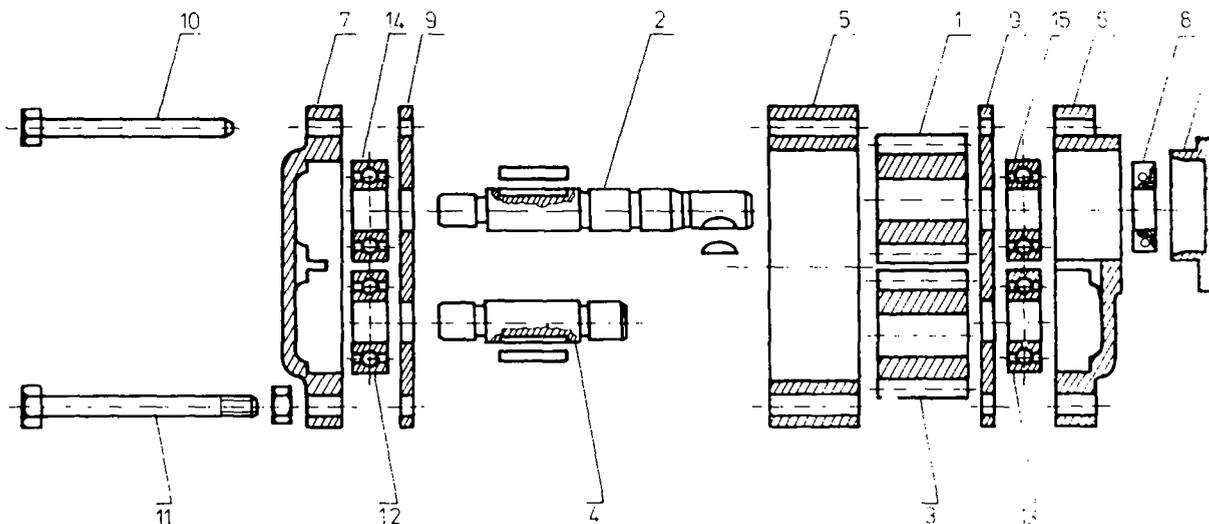


Fig. 4. Component parts of the oil pump.

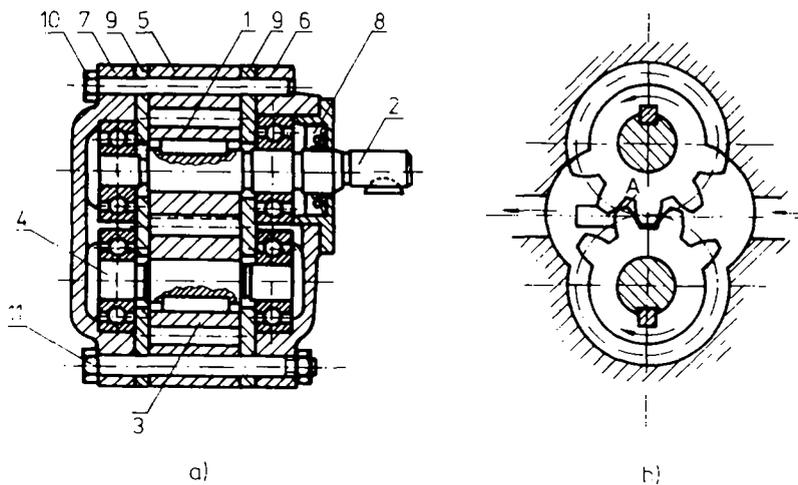


Fig. 5. Oil pump to be assembled.

axle bar in a hole or the engagement of the first gear into the other one. At the same time, there are different types of displacement movements of the workpieces. There is a gross motion for approaching the workpiece near the assembly module, and a fine motion for approaching the workpiece to the start point for insertion task. In this case, the movement of the robot is planned in order to reach the insertion point without touching the other components of the assembly module.

As we described before, in order to do the assembly the system does the off-line planning, where the *grasping*, the *insertion* and the *displacement* specialists compute the best possible solutions to carry out the task. These solutions are sometimes contradictory since each specialist does not know the geometric and physical constraints of the other specialists. The *coordinator* is the module which gathers the specialist solutions and combines them, imposing the constraints (example: the accessible positions required by the perception sensors) in order to obtain the solution jam. Once the off-line plan has been completed, the *supervisor* takes control, and its three modules, the *scheduler*, the *multisensor integrator* and the *local error recover*, are activated to accomplish the assembly task.

The *coordinator* starts the off-line process by activating the insertion specialist which studies the solutions between the front and the middle frame, this subassembly and both gears, the engagement of one gear into the other one, the insertion of both gears and the back frame, and the insertion of the pins into the frames. These solutions are good from the point of view of this specialist, although they do not take into account the physical constraints to reach the insertion point or the movements required to do the task. Then, the *coordinator* activates the *displacement* specialist in order to know if the insertion solutions can be reached by the robot. These solutions include the gross and the fine motion. Finally, the *coordinator* activates the *grasping specialist* which, with the data obtained from the other two specialists, finds the best gripping

points. Sometimes, since the displacement specialist is applied before the grasping specialist, new constraints appear due to the last specialist and the plan has to be modified to include the new conditions. Besides the tasks explained before, the specialists select the best strategies for doing local recovery and this information is included in the supervisor program.

In the oil pump example, the problems come with the insertion of the gears, where there are several uncertainties produced by the first gear, and with the insertion of the front frame due to the axle bar of the motor moving gear. With the first solution, the *coordinator* studies the constraints imposed by the perception sensors (in our case, a TV camera, four infrared proximity sensors, a tactile array and a force-torque sensor) and goes back to the specialists if the plan cannot be performed.

The *supervisor*, with the program generated by the off-line planner, creates the task schedule. In the schedule there are calls to the three specialists which take control of the specific task and which use the multisensory integration module. For example, the multisensory integration module takes control in the location of the front frame, first done by a TV camera, and then corrected by the proximity sensors, or the insertion of the second gear into the back frame. In the last case two sensors are combined in parallel with different strategies, the force and torque sensor and the tactile array sensor. The first one is used to stop the insertion motion and change the strategy to a circular motion, meanwhile the tactile array detects if the axle bar has slided in the gripper. The data of both sensors are used to correct the insertion operation, and to supervise the operation. In case of error, the error recovery module is activated by the *supervisor* and a programmed plan is applied to correct the problem.

## CONCLUSIONS

A system for automating both the programming and monitoring of an assembly cell has been presented and its operation has been illustrated through an example.

The system is currently being implemented. Concerning the *grasping specialist*, some of the computational geometry algorithms and heuristic search procedures required by its off-line part have been programmed, tested and linked with the PADL/2 (© Rochester University) CAD database describing the workpieces. An off-line specialist for 2D displacement has been completed and its on-line counterpart, which makes use of infrared proximity sensors to monitor the robot trajectory, is now being tested; the extension of this prototype to the 3D case is currently under development. In relation to the *insertion specialist*, the algorithms to build the graph of states and to find paths through it, have been implemented for a 2D environment. The general-purpose knowledge-based tool that will be used to implement the *supervisor* is now ready for its application. The sensory information required by each specialist has been assessed and the available sensors (infrared, tactile, force and torque, and vision) have been tuned to supply this information through the use of the appropriate procedures. In sum, a system for intelligent automatic robotic assembly has been designed and some of its modules have been implemented, their integration into a working system constituting the final goal of this research.

#### REFERENCES

1. Avila, F., Torras, C.: Propuesta de un sistema de planificación y monitorización de la aprehensión de piezas por un robot. *Primer Congreso Iberoamericano de Inteligencia Artificial (IBERAMIA' 88)*, Barcelona, Spain, 1988.
2. Avnaim, F., Boissonat, J. D., Faverjon, B.: A practical exact motion planning algorithm for polygonal objects amidst polygonal obstacles. Technical Report, INRIA (centre de Sophia-Antipolis), France, 1988.
3. Basáñez, L., Kelley, R. B., Moed, M. C., Torras, C.: A least-commitment approach to intelligent robotic assembly. *Proceedings 1988 IEEE International Conference on Robotics and Automation*, pp. 1318-1320, Philadelphia, U.S.A., 1988.
4. Doshi, R. S., Desai, R. S., Lam, R., White, J. E.: Integration of artificial intelligence planning and robotic systems with AIROBIC. *Proc. IMACS Conf. on Expert Systems for Numerical Computing*, West Lafayette, Indiana, U.S.A., 1988.
5. Ilari, J., Reyna, J. Ll.: Some experimental results using heuristics for solving the find-path problem in C-space. *Proc. Symposium on Theory of Robots*, Vienna, Austria, 1986.
6. Ilari, J., Torras, C.: The classical 2D find-path problem: improving search efficiency by using orientation heuristics. *SPIE Symposium on Advances in Intelligent Robotic Systems-Mobile Robots II*, Cambridge, MA, U.S.A., 1987.
7. Laugier, C.: A program for automatic grasping of objects with a robot arm. *11th. Int. Symp. on Industrial Robots*, Tokyo, Japan, 1981.
8. Lozano Pérez, T.: Spatial planning: A configuration space approach. *IEEE Trans. on Comput.* C-32(2): 108-120, 1983.
9. Mason, M. T.: Manipulator grasping and pushing operations. Ph.D. Thesis, M.I.T. Artificial Intelligence Laboratory, 1982.
10. Mason, M. T.: Compliance and force control for computer controlled manipulators. In *Robot Motion*, Brady, M., Hollerbach, J. M., Johnson, T. L., Lozano-Pérez, T., Mason, M. T. (Eds). Cambridge, MA, M.I.T. Press, 1982.
11. Nevins, J. L., Whitney, D. E.: Research on advanced assembly automation. *IEEE Comput.* 10(12): 24-38, Dec. 1977.
12. Orteu, I., Sanfeliu, A.: MESYAK: un entorno para la arquitectura del sistema PIRS. *III Symposium Nacional de Reconocimiento de Formas y Análisis de Imágenes*, Oviedo, Spain, 1988.
13. Peshkin, M. A., Sanderson, A. C.: Reachable grasp on a polygon: the convex rope algorithm. *IEEE J. Robotics Automation*, 2, 53-58, 1986.
14. Sanfeliu, A., Torras, C., Font, J., Ruiz, J.: Active-recognition system for the acquisition of overlapping and partially-hidden workpieces. *Proc. 1st. Symposium on Robot Control*, Barcelona, Spain, 1985.
15. Schwartz, J. T., Sharir, M.: On the piano mover's problem: I. The special case of a rigid polygonal body moving amidst polygonal barriers. *Commun. Pure Appl. Math.* 36:345-398, 1983.
16. Suárez, R., Basáñez, L.: Automatic insertion planning for robotic assembly. Technical Report IC-DT-1988.05, Instituto de Cibernética (UPC/CSIC), 1988.
17. Wolter, J. D., Volz, R. A., Woo, A. C.: Automatic generation of gripping positions. *IEEE Trans. Systems, Man Cybernetics* 204, 1985.
18. Yap Chee, K.: Algorithmic motion planning. In *Advances in Robotics*, Yap Chee, K., Schwartz, J. T. (Eds), Vol. 1. Lawrence Elbaum Assoc., 1987.