PROGRAMMING AUTONOMOUS ASSEMBLY AGENTS: FUNCTIONALITY AND ROBUSTNESS.

Lykourgos Petropoulakis, Research Fellow, Robotics
Chris Malcolm, Lecturer, Intelligent Robotics

Department of Artificial Intelligence,
University Of Edinburgh,
5 Forrest Hill,
Edinburgh EH1 2QL,
Scotland.

Abstract

This paper focuses on the problems associated with instructing assembly robots to perform tasks in an autonomous, reliable, and competent way within a real world environment. In the context of achieving these objectives we identify two major limiting factors directly affecting the efficiency, task reliability, and economic viability of robotic systems: programming functionality, and robustness in the face of uncertainty. This paper examines the way these issues have been addressed up to now and explains the present research effort in Edinburgh University. It also reports on our experimental domain which is a complex form of the "blocks world". This naturally generalises to assembling walls from predominantly rectangular components, and is designed to cope with variations in part size and location as a fundamental property of the domain.

1 Introduction

Robotic assembly has been – and still is – the subject of intensive research. The successful and commercially viable use of robots in assembly has so far been restricted to long-running and well-defined tasks. Attempts to diversify and increase the scope of robots within the assembly domain has been met with only moderate success. There appear to be two main reasons for this: the lack of assembly-oriented programming systems, and uncertainty in the assembly domain. Because of the former, the successful completion of assembly processes relies on the detailed "translation" of the assembly tasks into robot motions – an inefficient and time-consuming process. On the other hand, uncertainty in assembly cells directly affects task-reliability. Better robot control systems have not improved matters substantially. This is because reliability in assembly ultimately depends upon the interactions of the assembly components rather than the robot performance. Using sensors to improve reliability has been proved difficult within the existing framework of programming systems.

These issues are generic to autonomous robotic assembly systems and have been the subject of long term investigation in Edinburgh. Our current research program provides an architecture which offers robustness and reliability by the suitable decomposition of the assembly process and the appropriate control of uncertainty in terms of purposefully designed *behavioural modules.*

1.1 Research in Robot Programming Systems

To complete even the simplest of tasks, contemporary robot programming requires detailed descriptions of the robot motions. Such descriptions are not directly related to the objectives of the assembly and this presents difficulties in task identification, program amendments and debugging. In view of the high degree of task and machine dependency of such programs, portability is restricted, and serious reprogramming efforts can result from small changes in the assembly objectives. Moreover, the user is expected to have a good knowledge of the robot system, its programming environment and the intricate demands of the assembly. Under these conditions, very few robot assembly systems can claim true flexibility and efficiency.

Long term research in this area has resulted in systems such as AL [1], AUTOPASS [2], LM-GEO [3], which raise the level of abstraction of the assembly description. RAPT [4], developed in Edinburgh, is an off-line programming system which is independent of assembly, robot, or size of objects and capable of inferring robot positions from the spatial relationships between features of objects (e.g. **against**, bottom_of_block2, top_of_block1). Experiments with RAPT indicated considerable advantages in flexibility, ease of programming, and portability [5]. However, in converting spatial relationships between parts into robot positions in terms of a global coordinate frame, RAPT assumes a distinct and direct relationship between part and robot motions which can only exist in ideal environments. This important issue [6] is directly related to the subject of uncertainty in assembly cells.

1.2 Uncertainty and Use of Sensors

Since uncertainty is a characteristic of real environments and, therefore, constantly present in any process (only the degree and form of uncertainty vary), its effects can combine in complex ways to create serious functionality problems in robot assembly systems. Thus, it is not unusual for the cumulative effect of small errors in part locations and gripping actions to result in assembly failure. Structuring the robot environment to reduce uncertainty can be expensive both in terms of specialised equipment and because it inhibits fast product changeover. It is also contrary to the principle of flexible automation. Attempting to counteract the problem of uncertainty through the use of sensors has proved a non-trivial issue. In general, processing sensory information at run-time to update dynamic world models and reschedule assembly tasks requires qualitative reasoning about highly complex situations. The amount of detail needed, the required reasoning power, and the computational expense involved makes a fully automated reasoning system difficult to construct. Robot assembly systems still rely on human operators to provide a lot of the assembly details while sensors function mainly as means of activating error-recovering procedures, and reducing dependency on start-time knowledge.

Research in the off-line analysis of uncertainty [7, 8] has dealt mainly with the quantitative (geometric) aspects of uncertainty using this analysis as a criterion for assessing potential failure. If failure is anticipated, use of sensors or redesign may be required. There are two main criticisms against these approaches: firstly, they are computationally intensive and this has raised serious doubts about their tractability in less structured environments. Secondly, their predictions are based on the "worst case" principle which, in combination, can lead to gross over-estimates.

2 The Behavioural Approach

The behavioural approach to robotic assemblies was first presented [9] as an alternative to assembly programming methods beset with computational intractability, theoretical complexity, and questionable functionality. The main idea of the method is that the assembly system is composed of a number of elementary units, behavioural modules, which can be combined to perform complex part manipulations and assemblies. There are two fundamental principles about the design of these modules: they are predominantly concerned with the objects of the assembly and their manipulation; also they are purposefully built to incorporate uncertainty handling as an integral part of their design. Hence, behavioural modules must ensure the correct execution of their allocated task, irrespective of the existing uncertainty, provided that the latter is within their design bounds. The general guidelines for the design of behavioural modules have been set out [10] in response to the varying requirements of different robot assembly domains (e.g. manufacturing, construction, etc.), and in accordance with the idea of flexible automation which this approach intends to preserve and enhance. Thus, much in the same way that servo-control systems guarantee, within their design competence, to attain certain output standards, behavioural modules should be able to function in a comparably robust and competent manner.

2.1 Programming in Terms of Behavioural Modules

The successful construction of behaviour-based robot assembly systems relies primarily on decomposing assembly strategies so as to reveal the basic activities required to achieve the various subassemblies. It is assumed that a set, or a number of sets, of elementary modules would exist for the robot system and that they would represent these basic activities. In accomplishing the assembly, higher level behaviours may be constructed by combining behaviours at a lower level. Provided, therefore, that elementary behaviours are robust in their design, the programming system should exhibit a similar degree of robustness. Thus, the problem of off-line programming of robotic assemblies could be reduced to checking whether a particular action lies within the competence of the general behaviours used, rather than be concerned with the detailed description of robot motions. Conceptually the behavioural modules may be thought of as supporting agents for a virtual robot which translates part motions into robot motions.

In building elementary behaviours for a robot system, the user has complete freedom in the design of the modules. The architecture allows for the incorporation of additional modules and the expansion of the system according to individual needs. In this way the designer can utilise the similarities in parts and robot actions which often exist even in different assembly domains – i.e., a "peg-in-hole" is a problem which exists in both manufacturing and construction [11]. In accordance with the philosophy of the approach, higher level modules could also be made sufficiently general to be used on various occasions. Consider for example a wall building robot [12] utilising a "built-wall" module, consisting of several behaviours, being used for different walls merely by specifying parameters such as location, height, and length.

Sensors can increase generality and the behavioural approach is designed to use sensory information in a simple and general way by incorporating sensing as an integral part of behavioural modules e.g., utilising force sensing information during a "part-insertion" behaviour, or vision information in "part-acquisition". Actions involving sensors can also be thought of as stand-alone behaviours, e.g., a "identify-part" module using vision. Using sensors in this way has distinct advantages. For example, both vision and force sensing information can be used at a local level to assist a "part-insertion" behaviour without recourse to high level reasoning or reference to "world models". This approach should normally be sufficient for a wide range of problems as it also allows for the progressive utilisation of more information either from new sensors (e.g. touch sensors) or by redeploying existing resources (e.g. a change in camera angle). This methodology advocates using sensors actively (assist in the task) rather than passively (detect and report errors).

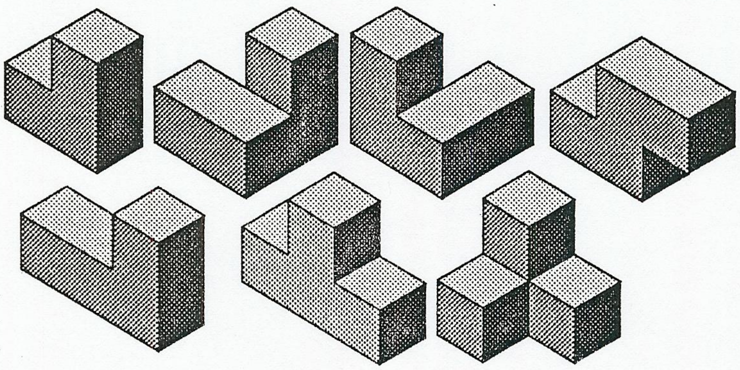

Figure 1: The soma parts

3 The Experimental Domain

To test these ideas, an experimental domain has been used, *the soma world* [1] (figure 1), which contains the general characteristics of assembly systems (i.e. shape-dependency, part-mating requirements), but which also presents a domain suitable for testing new concepts. The soma parts can be arranged to form various assemblies (figure 2) and in various combinations. This is a property that can test the generality of both major subdivisions within our experimental robotic assembly system: the off-line planning system, and the on-line execution system.

3.1 The Off-line Planning System

This is written in PROLOG (POPLOG) and runs in a Sun 3/160 (8Mbyte). It produces assembly plans in VAL2 code which are transferred to the on-line system via an RS232 cable. The system is general enough to deal with other parts (not shown here) and is size independent. To produce an assembly plan the planner goes through hierarchical stages. Initially, the planner is given the shape of the parts and the final assembly and evaluates how the parts are to be disposed.[2] Depending on the assembly, the planner may take a few minutes or several hours to find all possible solutions. In the second stage, a gravitationally stable ordering of the assembly is selected. This may take a few seconds. Next, the planner finds ways of gripping each part so that the gripper and the part do not interfere with the assembly or other parts. In the penultimate stage, the planner tries to match the pick-up and put-down grasps (generally they differ). If matching cannot be achieved, the planner arranges for an intermediate regrasp action. If

---

[1] A mathematical and combinatorial problem [13].

[2] Not really an assembly problem, but it checks the generality and reasoning power of the planner, and creates assemblies effortlessly.

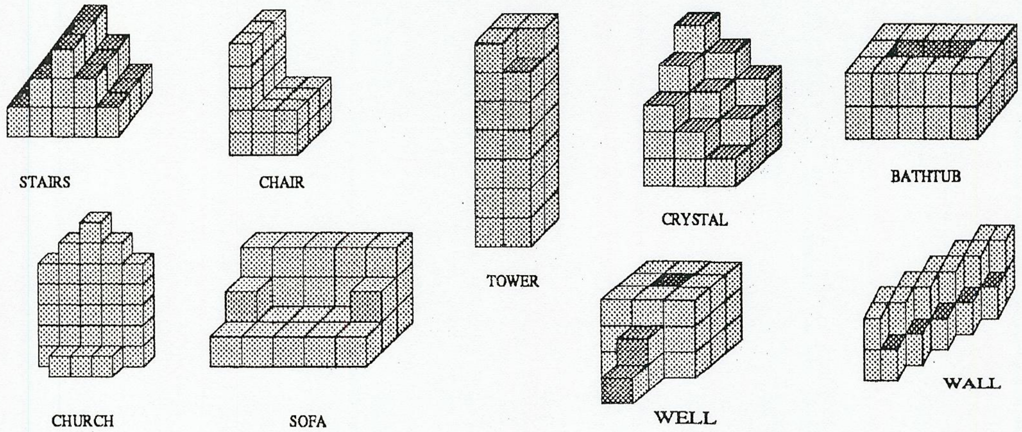

Figure 2: Some of the soma assemblies

any of the above stages fails, the planner normally backtracks to a previous level and tries a different route. Finally, the system is able to generate the detailed manoeuvres for each part in a form suitable for a given robot system (in this case VAL2).

3.2 The On-line Execution System

This comprises a 5 degree-of-freedom Adept1 robot which is programmed in VAL2. This is also the language in which the behaviours are written. There are 9 modules which are used to construct the assemblies. An *initialisation* process instantiates such parameters as the size of the parts and the nominal positions of the objects. Such values are utilised by the behavioural modules to manipulate the parts. Note: there is no on-line representation of the parts and the system is unaware of the precise location of the parts and their shapes. Initial trials with the system did not use any sensors, relying instead in constrained motion to control uncertainty and increase reliability. Reliability trials [6], lasting about 45 hours, showed 12 failures in 529 automatically generated assembly plans.

3.3 Further Development

Currently the powers of the planner are being enhanced so that it can cope with increasing variation in part shape and assembly complexity. At the same time, the on-line system is gradually being improved with the addition of 2D vision using a moving camera (held by an RTX robot), and the implementation of a force sensor unit on the Adept. Successful trials using the vision system [14] have already been carried out. Notably, neither the planner nor the existing behaviours had to be altered in any way with the incorporation of the vision system. Other types of sensors have also been tried (touch, and light switches) in various supportive projects, and these could also be integrated into the system. Wall constructing behaviours have been developed and initial trials performed using rectangular blocks of the scale permitted by the hardware. It is intended that additional tests should be carried out on a larger scale as soon as possible.

4 Conclusions

The behavioural architecture, presents itself as a means for reducing complexity in the assembly domain, by increasing the competence of the assembly agents. It also gives a clear direction to more reliable and efficient programming of robotic assemblies both by providing the essential flexibility and handling of uncertainty, and by removing the need for the user to describe assemblies at the laborious level of robot motions. In addition, it provides a simple and general framework for the incorporation of sensing. To test these ideas, an experimental system has been built capable of planning and executing assemblies in an autonomous manner. The planning stage is completely independent of part size and can deal with a variety of parts and assemblies. The generality of the system is currently being enhanced by the addition of sensors (vision and force sensing). Initial trials with the vision system have been carried out and implementation of force sensing will follow shortly. In addition to the above, it is intended that the approach is further tested using even more rigorous and complex tests. It is believed that this approach will help increase robot task-reliability and programming efficiency so that assemblies become easy to programme and economically attractive.

Acknowledgements

This work was supported by the ACME Directorate of the SERC, grant GR/E 68075: Behavioural modules – A new approach to robotic assembly, with industrial collaboration from GEC Research, Great Baddow, U.K.

References

[1] Finkel, R., Taylor, R., Bolles, R., Paul, R., Feldman, J., " Al, a programming system for automation," Artificial Intelligence Laboratory, Stanford University, AIM-177, November, 1974.

[2] Lieberman, L.I., and Wesley, M.A., "AUTOPASS: an automatic

programming system for computer controlled mechanical assembly", IBM J. Research Development 21, 4(1977), 321-333.

[3] Maser, E., "Geometric programming of assembly robots (LM-GEO)", International Meeting on Advanced Software in Robotics, Liege, May 1983.

[4] Popplestone, R.J., Ambler A.P., and Bellos, I., "RAPT, a language for describing assemblies," Industrial Robot 5, 3(1978), 131-137.

[5] Petropoulakis, L., Malcolm, C., and Howe, J.A.M, "RAPT: An Assessment of An Off-Line System for Programming Robotic Assemblies", Working Paper 219, Department of Artificial Intelligence, University of Edinburgh, Oct 1989, (unpublished internal paper).

[6] Malcolm, C., and Smithers, T., "Programming Assembly Robots in Terms of Task Achieving Behavioural Modules: First Experimental Results", International Advanced Robotics Programme, 2nd Workshop on Manipulators, Sensors, and Steps towards Mobility, October 1988, Manchester, U.K.

[7] Fleming, A.,"Analysis of Uncertainties in a Structure of Parts 1", and "Analysis of Uncertainties in a Structure of Parts 2", DAI Research Papers Nos. 271 and 272, Department of Artificial Intelligence, University of Edinburgh, 1985.

[8] Requicha, A.A.G., and Tilove, R.B., "Towards a Theory of Geometric Tolerancing", Int. Journal of Robotic Research Vol 2, No. 4, 1983.

[9] Malcolm, C., Planning and Performing The Robotic assembly of Soma Cube Constructions, MSc Dissertation Edinburgh University, September 1987, 87 pages.

[10] Malcolm, C., Smithers, T., Hallam, J., "An Emerging Paradigm in Robot Architecture", 2nd Conference on Intelligent Autonomous Systems, Amsterdam, 11-14 December 1989.

[11] Bock, Thomas-Alexander, "Robot-Oriented Design", 5th International Symposium on Robotics in Construction June 6-8, 1988, Tokyo, Japan.

[12] Slocum, A.H., and Schena, B., "Blockbot: A Robot to Automate Construction of Cement Block Walls", Robotics 4, 1988, 111-129.

[13] Carson, G.S., Soma Cubes, Mathematics Teacher, Nov 1973, 583-592.

[14] Chongstitvatana P., and Conkie A., " Behaviour Based Assembly Experiments using Vision Sensing", DAI Research Paper No 410, Department of Artificial Intelligence, University of Edinburgh, 1990.