Constraint-Based Inference of Assembly Configurations * 

Federico Thomas and Carme Torras 
Institut de Cibernètica (UPC-CSIC) 
Diagonal 647, 08028-Barcelona. 
SPAIN.

Abstract 
A system for the automatic synthesis of assembly configurations is presented. It consists of the propagation, combination and satisfaction of three types of constraints: shape-matching constraints, constraints on the degrees-of-freedom and non-intersection constraints.

Given a high-level description of an assembly and the models of the workpieces, the system determines which parts of the workpieces should be mated and produces a set of homogeneous-coordinate transformations defining the relative position and orientation of each workpiece in the final assembly. This system can be seen as a previous step towards a practical and efficient assembly planner.

1 General framework

The method described in this paper for the automatic inference of final assemblies is embedded in the planner described in [1]. The part of the planner described here is confined to obtaining the spatial relationships between workpieces in the final assembly, from an initial description consisting of the models of the workpieces and, if necessary, disambiguating instructions which specify constraints on the degrees of freedom (d.o.f.) between parts of the workpieces.

We have adopted a constraint-based approach to solve the problem, in which the process of synthesizing assembly configurations is conceived as a progressive refinement of an initial description by the application of successive constraints. These constraints are of three types: (1) shape-matching constraints between the mating parts of the workpieces to be assembled (complementary shape and similar parameters); (2) constraints on the degrees-of-freedom (d.o.f.) between the parts of the workpieces; and (3) constraints of non-intersection between workpieces. The first two types reduce the dimensionality of the space of d.o.f. of the workpieces (that is, the space of assembly configurations or configuration space), while the third type eliminates from configuration space the zones that lead to interferences. The remaining configurations after carrying out all the reductions and eliminations implied by the different constraints will be the solutions to the problem.

Fig. 1 (a) Simple assembly problem (b) Associated graph of compatibilities between features.

2 Shape-matching constraints

The first step in the process is the extraction of all relevant features of the involved workpieces, according to a dictionary of possible features [4]. Afterwards, matchings between features are established, leading to a graph of compatibilities between features (GF graph).

The developed system takes into account "graded" shape-matching constraints, that is, those involving concavities and convexities that do not exactly match each other.

Once the GF graph has been created, the control of the system looks for an optimal matching between workpieces. This problem can be conceptualized as a graph matching problem [7]. Obviously, in this case, optimality is not always a guarantee of feasibility. Note that the matchings between features have a direct translation into constraints on the d.o.f. leading to a graph of spatial relationships, or GR graph, and these constraints must be also satisfied. This is best shown through an example.

Fig. 1(a) shows three workpieces to be assembled and fig. 1(b) the graph of compatibilities between their features. Once an optimum is obtained, compatibilities in the matching of the GF graph are translated into a GR graph, whose consistency is checked as described in the next section.

*This work has been partially supported by the grant CA 83/188 of the US-Spain Joint Committee for Scientific and Technological Research.

CH2555-1/88/0000/1304$01.00 © 1988 IEEE
4 Constraints of non-intersection

Two types of interference checking can be clearly distinguished: the detection of intersections between objects in fixed positions and the detection of collisions between objects moving along specified trajectories. The configuration problem tackled here, despite its static nature, has to be viewed as an instance of the second type of checking. The reason for this is the possible existence of d.o.f. for the workpieces, along which we have to find interference-free locations.

A good way to represent these constraints is by using the Configuration Space (C-Space). The main problem concerning C-Spaces is their construction because of the computational effort involved. Thus, it is very important to avoid its explicit representation. Moreover, since clearances between workpieces, in the assembly domain, can be arbitrary small, an exact representation of the boundaries and obstacles in the C-Spaces is required.

Our procedure follows the typification of constraints stated in [6], which is framed within a predicate-based approach as in [2]. Two algorithms have been developed: one concerning a simplified representation of C-Spaces, and the other related to the development of local experts, as in [3], in order to solve some interference problems without explicitly building C-Spaces.

An example using this kind of constraints is given in fig. 3. In this figure, there are three workpieces, two of them still having one d.o.f., which is limited to ranges of values due to constraints of non-intersection. The lower the clearances between them, the greatest the resolution in the C-space must be to find intersection-free configurations.

Conclusions

We have briefly presented a constraint-based method to infer assembly configurations, as well as specialized techniques to deal with each constraint type.

The method has been devised so that it can be very easily extended to deal with new constraint types. Thus, the obvious extension is the addition of a new operator for dealing with sequencing constraints [5]. This would allow the system to distinguish those configurations for which there is a feasible assembly sequence from those for which, despite satisfying all other constraints, there is no such a sequence.

References


