Planning Motion in Contact
to Achieve Parts Mating

Anastasia Koutsou

Abstract

The research presented in this thesis focuses on planning a parts mating task by formulating a plan for a motion in contact. Specifically, it deals with the problem of moving a 3-dimensional polyhedral object while maintaining contact with a set of stationary obstacles. An algorithm has been developed and implemented which derives a motion plan as as sequence of contacts that have to be established during a motion from some initial to some final contact state.

A configuration space approach to motion planning has been adopted. In planning a motion in contact, a subset of the configuration space, the contact space, is of relevance. The contact space is decomposed into faces of various dimensions and adjacency relations between the faces are determined. For a path-connected contact space, if the 0-dimensional faces (vertices) are connected by 1-dimensional faces (edges), then the motion planning problem is reduced to the problem of searching for a path in the graph of vertices and edges.

The algorithm has two stages. In the first stage, the graph of the surfaces of various dimensions, on which the faces of the contact space lie, is found, and in the second stage, the vertices and edges are determined. The implemented algorithm makes use of a spatial reasoning system for finding the intersections of surfaces and a solid modeller for checking physical interference.

Spatial relationships are used to represent the constraints on the relative location of objects imposed by contacts. Using a spatial reasoning system based on the RAPT inference engine, it is possible to associate a spatial relationship with every contact state. The spatial relationship is arrived at by considering conjunctions of 5 degree of freedom spatial relationships which describe the basic types of contact among polyhedral objects.

The plan for a motion in contact is thus formulated in terms of the interactions between features of objects. A method for transforming the plan into a sequence of motions with sensory involvement could follow naturally from such a formulation of the problem.

Part of the work presented in this thesis is described in [Koutsou 85].

Acknowledgements

I DECLARE THAT THIS THESIS HAS BEEN COMPOSED
BY MYSELF AND DESCRIBES MY OWN WORK

(Anastasia Koutsou)

Table of Contents

List of Tables

List of Figures

List of Symbols

The following is a list of symbols used in this thesis. The third column gives the number of the relevant definition, if a definition exists, and the fourth column gives the page where the symbol first appears.

Chapter 3

Symbol	Description	Def.	Page
$bdry(A)$	boundary of a set A		45
B	moving object	3.2.1	43
$B@Z$	object B at location Z		44
c	clause	3.4.1	59
C	configuration space	3.2.5	44
c^d	a d-cell of a cell complex	3.3.1	54
$cl(A)$	closure of a set A		42
$clauses(D)$	clauses of a descriptor D		96
$CLAUSES$	set of all possible clauses between two objects		69
$CONTACT$	contact space, configurations for which objects are in contact	3.2.8	45
C_s	cell of contact space corresponding to sign-assignment s	3.4.15	88
D	descriptor	3.4.3	62
S_D	configurations satisfying descriptor D	3.4.4	62
$DESCRIPTORS$	set of all possible consistent descriptors		69
E	stationary objects, environment	3.2.2	43
E_D	configurations satisfying descriptor D exactly	3.4.10	66

Chapter 4

Chapter 5

Chapter 1

Introduction

1.1 Motivation

Over the last few years there have been some impressive advances in the design of robots for industrial use. There are now robots in industry capable of performing a large number of diverse tasks such as materials-handling, welding and painting. Yet, with respect to one of the most important of manufacturing processes — assembly operations— the pace of advance has been relatively slow. Currently, only a small proportion of robots in industry are being used for assembly tasks. One of the major problems that remain largely unresolved in the area of robotics is the planning and execution of parts mating operations, that is, actually bringing parts into contact with each other in the desired configuration.

If there were no uncertainty as to the position and shape of the parts, and if the motions of the robot itself were absolutely accurate, then parts mating would be a relatively straightforward operation. In that ideal world, to ensure that a parts mating task would be executed successfully, it would be sufficient to instruct the robot to move through some specified sequence of positions. In the presence of uncertainty, however, describing and executing a task solely in terms of positions is inadequate.

It is generally accepted that what is needed for the successful completion of parts mating operations is a robot system that is capable of taking into account the interactions of the parts, i.e. the contacts, and complying with the constraints imposed by them. A robot capable of performing this function must

be equipped with tactile sensors. This thesis is concerned with the problem of planning parts mating operations and in the approach I propose to follow, the sensors are employed with a view to using the physical constraints of the task to *guide* a part to its destination. Following this line of thought, parts mating is viewed as an operation that can be accomplished by moving the part to be mated while maintaining contact with the rest of the assembly.

1.2 Objective

The research presented in this thesis focuses on planning a parts mating task by means of a plan for a motion in contact. Such a plan should be formulated in terms that would allow it to be transformed into an executable robot program. Having established that for parts mating operations it is not sufficient to consider solely the positions of the objects, it follows that the plan has to be formulated in terms of the interactions between features of the objects. In effect, my objective is to produce a plan for a motion in contact as a sequence of *contact states.* A method for transforming the plan into a sequence of motions with sensory involvement follows naturally from such a formulation of the planning problem. The issue of deriving and executing a sequence of motions falls outside the scope of this research.

In this research, the parts considered are restricted to be three dimensional rigid polyhedral objects with six degrees of freedom of motion. It will also be assumed that only one part can be moved at a time.

After introducing some of the basic concepts used, the developed system for planning a motion in contact will be overviewed.

1.3 Elements of the Approach and Representation of the Problem

In the last few years the issue of motion planning has been considered extensively in robotics research, while the issues of planning a motion in contact and assembly planning have also received some attention. A detailed survey of research in these areas can be found in Chapter 2. In this section I shall outline the basic conceptual tools which I shall be using to tackle the problems of motion in contact, namely the notions of *configuration space*, *contact space* and *spatial relationships*. Before this, I shall illustrate through an example how the notion of degrees of freedom of motion of the objects will be used in this research in order to plan a motion in contact.

1.3.1 Degrees of Freedom

When two objects are brought into contact, their relative motion is constrained and the number of degrees of freedom (d.o.f.) of motion is reduced. The d.o.f. of a contact state are the d.o.f. of motion permitted *with full contact still being maintained*. It has to be emphasised that this is different from the *natural* d.o.f. of motion, i.e. the possible ways in which a part can be moved irrespectively of whether contact is maintained or broken. For example, the situation of a block in a corner is described by a contact state with zero d.o.f. — any motion of the block would result in breaking the contact with the corner. The block, however, is not physically constrained to be in the corner — it could still be moved upwards or sideways. Similarly, the situation of a block on a table is described by a contact state with three d.o.f. — one rotational and two translational d.o.f.

One approach to the construction of the motion plan is to reduce gradually the d.o.f. of the moving part by gradually increasing the number of features that are in contact. For example, a plan for placing a block in a corner would entail the following motions (see Figure 1-1): (a) lowering the block until some

vertex comes into contact with the top surface; (b) rotating about that vertex until an edge comes into contact with the top surface; (c) rotating about this edge until the whole face comes into contact with the top surface; (d) translating along the the top surface until a vertex comes into contact with one of the walls; (e) rotating until the whole face comes into contact with the wall; (f) finally, translating until the block comes into contact with the other wall. In this case, the d.o.f. are reduced one by one until the destination is reached.

In more complex cases, it is more difficult to reach the desired destination. An example is shown in Figure 1–2. In this case, the presence of the obstacle makes it more difficult to move the block to the corner. A plan for accomplishing such a parts mating task would then entail: (a) establishing some initial contact; (b) constraining the part to a zero d.o.f. contact state (Figure 1–2a); (c) moving the part to its destination through a sequence of zero and one d.o.f. contact states (Figure 1–2b). In this way, the transitions between the contact states of the plan would be simple one d.o.f. motions during which the part remains as constrained as possible.

1.3.2   Configuration space

A common conceptual tool for addressing problems related to motion planning is the notion of *configuration space* [Lozano-Perez 83]. A *configuration* of a system is the set of parameters required to specify a system completely. A point in configuration space corresponds to some configuration of the system. In the case of rigid objects, the configurations of the objects can be specified by their locations. In this thesis, the motion of a 3-dimensional rigid object with 6 d.o.f. is considered. A configuration can thus be specified by a 6-tuple and the resulting configuration space is 6-dimensional. Using this approach the problem of moving a 3-dimensional object in a 3-dimensional space is transformed into the problem of moving a point in a 6-dimensional space. As a result, the problem of planning a motion is reduced to the problem of finding a path in configuration space.
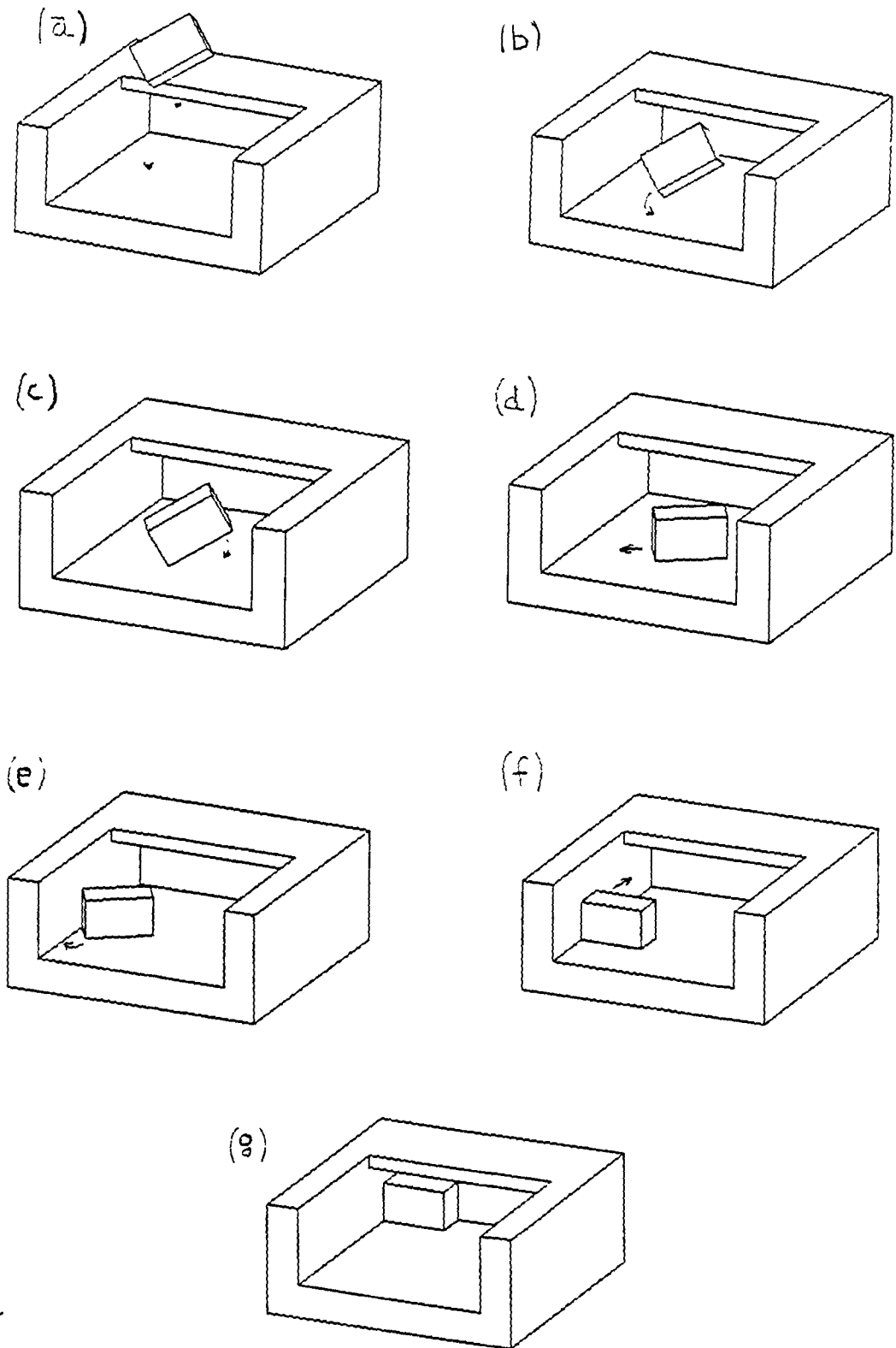
Figure 1–1: Placing the block in a corner by gradually reducing its d.o.f.
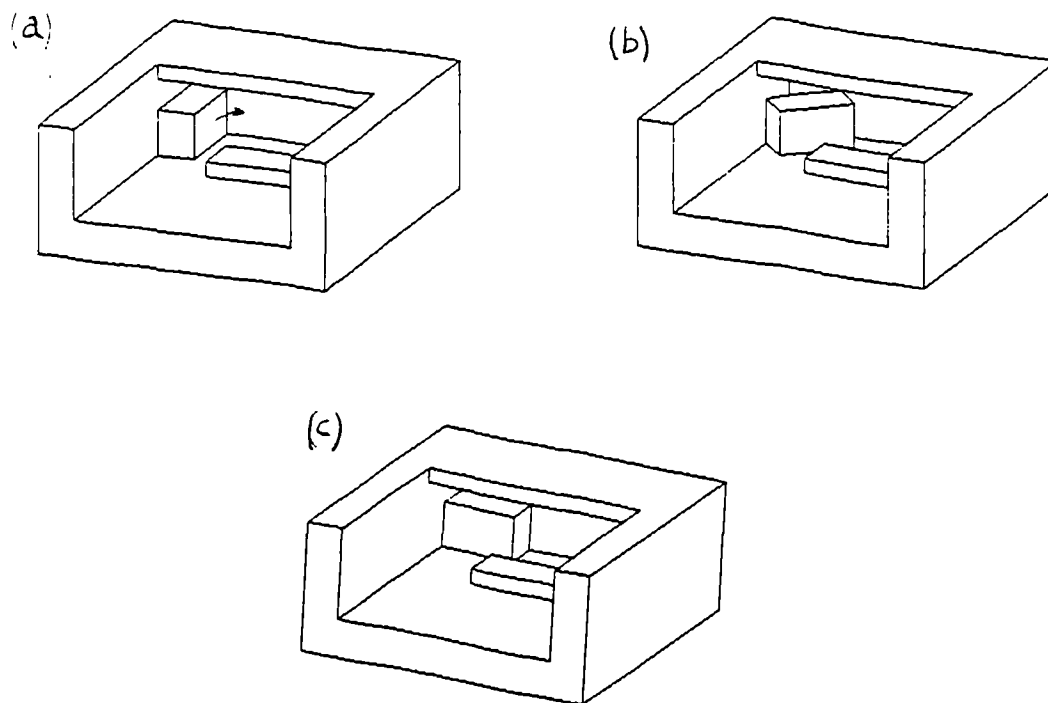
Figure 1–2: Manoeuvring a block around a corner

1.3.3 Contact Space

The notion of *contact space* ([Hopcroft and Wilfong 84a]) is used to denote the set of configurations for which the objects are in contact but do not overlap with each other. The contact space is thus a subspace of the configuration space. Using the notion of the contact space the problem of finding a motion in contact is reduced to the problem of finding a path in the contact space.

The contact space consists of faces of various dimensions. Informally, a face consists of configurations which satisfy some particular contact constraints, that is to say, it consists of configurations belonging to the same contact state. The dimension of a face corresponds to the d.o.f. of motion allowed while the contacts are maintained, i.e. to the d.o.f. of the corresponding contact state.

The approach which will be followed in this thesis is to decompose the contact space into faces of various dimensions, to establish the adjacency relations of the faces and then to find a motion plan as a path on the faces of the contact space.

1.3.4   Spatial relationships

Spatial relationships among features of objects are used as a means of specifying the relative locations of objects ([Popplestone, Ambler, and Bellos 80]). Examples of such spatial relationships would be "the bottom face of the block is against the top of the table" or "the right face of the block is parallel to the wall". Of special interest for the approach I propose to follow are spatial relationships which describe the interactions between features of objects. Spatial relationships of this kind specify the constraints on the location of the objects imposed by the contact situation.

A system which reasons about spatial relationships has been used in this research in order to decompose the contact space. The spatial reasoning system makes inferences related only to the locations of features and objects but not to their physical extent. In order to reason about contacts and physical interference it is necessary, however, to be able to make inferences related to the space occupancy of the parts. In this research I shall propose a method for decomposing the contact space in such a way that the issue of the constraints on the locations of the objects can be considered separately from the issue of physical occupancy. A spatial reasoning system will be employed for dealing with the first problem, i.e. the 'kinematics' of a task, and a solid modeller will be employed in order to deal with the second problem, i.e. the physical interference.

1.4   Planning a Motion in Contact

In this research I have developed and implemented an algorithm for finding a path for a motion in contact along the 1-dimensional faces of the contact space. The algorithm is based on a decomposition of the contact space into faces of various dimensions. The proposed decomposition is an enlargement of the decomposition in [Hopcroft and Wilfong 84b] for the case of 3-dimensional objects which can rotate and translate. This alternative decomposition makes a clear distinction between the 'kinematics' of contacts and body occupancy. In

addition, by introducing some additional 1-dimensional faces, it overcomes, in most cases, the problem of 0-dimensional faces of the contact space which are not connected by 1-dimensional faces.

The algorithm for the decomposition of the contact space makes use of a spatial reasoning system which handles constraints on the locations of objects imposed by physical contact, that is to say, it deals with the 'kinematics' of contacts. Issues related to body occupancy are handled by a solid modeller. The developed spatial reasoning system is an extension of the RAPT inference engine ([Corner, Ambler, and Popplestone 83]). With the use of the particular reasoning system, it is not necessary to resort to the solution of algebraic equations while planning a motion. General solutions of the algebraic equations have been formulated in terms of geometric relations between coordinate systems embedded in objects and their features. This approach differs significantly from the approach in [Donald 84], where an algebra system is employed.

Having decomposed the contact space into faces, the problem of planning a motion in contact is transformed into the problem of finding a sequence of faces that have to be traversed, given some initial and final contact state. In the case when the initial and final states are contact states with 0 degrees of freedom, i.e. 0-dimensional states, a path can be found along the 1-dimensional faces. A graph searching algorithm is used for searching the graph of 0- and 1-dimensional faces and for finding, therefore, a plan for a motion in contact as a sequence of 1-dimensional contact states.

The developed system has been tested for an example similar to the one shown in Figure 1-2.

1.5 Contents

In Chapter 2 the problem of planning parts mating is placed in the perspective of assembly planning, motion planning and fine motion planning. Research in these areas is reviewed and the relation of this thesis to other work is described.

Chapter 3 addresses the theoretical issues related to the motion of a 3-dimensional polyhedral object while maintaining contact with a set of stationary polyhedral objects. A mathematical framework is presented for describing the structure of the contact space — its faces and their connectivity. A decomposition of the contact space into faces is formulated in terms of the interactions among the features of the objects. Finally, an algorithm is outlined for achieving such a decomposition.

In Chapter 4 a system which reasons about spatial relationships between features of objects is presented. The spatial reasoning system is based on the RAPT inference engine ([Corner, Ambler, and Popplestone 83]). Given a pair of spatial relationships between two objects, the system is able to decide whether the constraints on the locations of the objects implied by the two relationships can be satisfied at the same time. If this is the case, then the system infers a more constrained relationship, which is equivalent to the original pair of relationships.

In Chapter 5 the issue of developing a plan for a motion in contact is considered. By bringing together the theoretical themes developed in Chapters 3 and 4, a method is developed for using the spatial reasoning system in order to construct a model of the contact space. The algorithm outlined in Chapter 3 for the decomposition of the space is described in more detail and its implementation is discussed. Using the constructed graph of the space, the development of a plan for a motion in contact is examined. Finally, ways in which the plan can be transformed into a sequence of motions are discussed.

Finally, in Chapter 6 the contributions and the limitations of the presented research are discussed and suggestions for future research are made.

Chapter 2

Review of Related Work

The issue of parts mating is one of the problems which have to be addressed for the purposes of automated assembly. The approach taken in this thesis is to bring the parts together through a sequence of motions such that the parts are first brought into contact and then their positions are further constrained by establishing more contacts until the desired relative positions are attained. Therefore, planning parts mating operations becomes an instance of the general motion planning problem, that of planning a motion of an object while maintaining contact with a set of stationary objects. When dealing with parts mating, apart from the issue of avoiding collisions with obstacles, problems arising from the presence of uncertainty in the environment become of major importance. For this reason, parts mating can be regarded as an instance of what is usually called fine-motion planning.

In this chapter the problem of planning parts mating operations will be placed in the perspective of assembly planning, motion planning and fine-motion planning. Research related to these areas will be reviewed and the relation of this thesis to other work will be outlined.

2.1   Assembly planning

Industrial robots currently in use are mostly programmed on-line by guiding them through a sequence of positions (teach mode). Research has been carried out for a number of years now towards higher level robot programming and off-line programming and there have been experimental systems developed exploring the issues involved in higher level programming such as AL [Mujtaba and Goldman 79], RAPT [Popplestone, Ambler, and Bellos 78], LM [Latombe and Mazer 81], AML [Taylor, Summers, and Meyer 82]. For the purposes of automated assembly some of the problems that have to be solved are:

- how are the robot and the environment to be modelled;

- at what level and how can a task be specified;

- where and how should each part be grasped;

- how is a part to be transferred without colliding with other objects;

- how is a part to be brought to some specified relation to the assembly;

- how to deal with parts tolerances and uncertainties ;

- how to recover from failure;

- how can sensors be used?

As we are moving to higher level systems, more of the above tasks are performed automatically and less has to be programmed explicitly. All of the above issues are currently being dealt with in robotics research and although proposals have been made for systems which are able to perform all these functions automatically (AUTOPASS [Lieberman and Wesley 77], LAMA [Lozano-Perez 76]), no integrated system exists at the moment.

[Lozano-Perez 82] divides task planning into three phases: modelling, task specification and manipulator program synthesis. The objective of an assembly planner is to transform a task specification into a manipulator program.

A task is specified by a number of goals (states) and perhaps by information on how they can be achieved. The higher the level of programming the fewer states that would have to be described. These states are described in terms of the positions of all the objects in the environment. The positions of the objects can be specified either explicitly, in terms of the position of the manipulator, or implicitly, in terms of the effects on the objects. The latter approach is referred to as object-level programming or task-level programming (RAPT [Popplestone, Ambler, and Bellos 78], LAMA [Lozano-Perez 76], LM-GEO [Mazer 83]). Within the framework of task-level programming, the task is specified using spatial relationships among the features of the objects. For example, 'Face1 AGAINST Face2'. These spatial relationships are then used for obtaining position constraints.

A system which derives position information from spatial relationships is the RAPT system described in [Ambler and Popplestone 75], [Popplestone, Ambler, and Bellos 80]. In that system, statements about spatial relationships are transformed into position equations and these equations are solved in order to obtain real values for the positions. An alternative system described in [Popplestone and Ambler 83], [Corner, Ambler, and Popplestone 83] is capable of recognising standard combinations of relationships between two objects and applying standard solutions. As a result, this system is more efficient by comparison to its precursor.

[Taylor 76] also considers the issue of spatial relationships. He extends the contact relationships treated in the RAPT system to non-contact relationships, resulting in inequality constraints on the positions. He restricts himself to cases involving one rotation and linearises the resulting equations by dividing up the range of the rotation.

A system of reasoning about spatial relationships based on [Popplestone and Ambler 83] will be described in detail in Chapter 4.

Motion planning lies at the core of manipulator program synthesis. During the assembly process, or any other robotics process, the manipulator has to pick up objects (grasping), transfer them and place them at appropriate positions (parts mating). In the case of transfer movements, the main difficulty is to avoid collisions with obstacles. In the case of grasping and parts mating, apart from the issue of avoiding collisions with obstacles, other issues also have to be considered, arising mainly from the need to have some objects coming into contact. Then, the issues of forces, friction, errors in the models and the control system of the robot have to be taken into account. We therefore make the distinction between motion planning and fine-motion planning. Issues related specifically with the problem of grasp planning ([Wingham 77], [Lozano-Perez 81], [Mason 82b], [Brady 82b]) will not be considered in this review.

Although I will be dealing with the problem of parts mating, some of the general approaches to motion planning are relevant and applicable. In Section 2.2 the general area of motion planning will be considered as a geometrical problem and forces, friction and uncertainty will be ignored. In Section 2.3 the issue of fine-motion planning will be examined.

2.2 Motion Planning

2.2.1 Formulation of the Problem

The general motion planning problem can be formulated as follows: given the description of the robot and the environment, find a path for the robot from some initial to some final location such that the robot avoids collisions with the various obstacles in the environment.

The term 'path' in the above definition is used to denote a 'physical' path which can be described in terms of positions. When the path is described in terms of positions, velocities and accelerations it is usually referred to as trajectory, although the two terms have been used interchangeably in the literature. A

review of the work on trajectory planning can be found in [Brady 82a]. In this section issues related to control theory, kinematics and dynamics will be ignored. What will concern us here is the issue of motion planning as a geometrical issue.

The general motion planning problem can be divided into various classes by varying all of the following characteristics ([Yap ss]):

- Dimension of the space: Although the underlying space of the motion planning problem is three dimensional, sometimes it can be sufficiently realistic to consider two dimensional problems.

- Geometry of the robot: Various cases can be characterised according to whether the robot is: (a) rigid; (b) it consists of a number of joints (Cartesian, revolute etc) or (c) there are more than one independent robots, in which case we talk about 'coordinated motion'.

- Boundaries of the objects: Various approximations can be used to model the objects and the robot, the most common being polyhedral approximations. A larger class of objects can be modelled using algebraic surfaces.

- Motion objective: Usually, the objective is to reach some specific location. Alternatively, the objective could consist of a set of locations with a common property. For example, the objective could be to place a box anywhere on a table.

- Optimality criteria: Most common criteria are shortest path, maximum clearance and minimum time. Usually, however, no optimality criteria are used.

- Degrees of Freedom: In the general case, our aim is to solve the motion planning problem for an arbitrary robot system which implies that we are dealing with an arbitrary number of degrees of freedom. Usually, however, we are concerned with a particular robot system and hence a fixed number of d.o.f. In this case, the motion planning problem is referred to as the basic motion planning problem.

In the next section I will review the various approaches and algorithms for the general motion planning problem and for specific instances of the problem according to the above classification.

2.2.2 Approaches to Motion Planning

The motion planning problem has been considered within the fields of robotics, A.I., computer science and, accordingly, various approaches have been taken. The problem has also been given various names such as mover's problem, piano mover's problem, collision avoidance, obstacle avoidance and find-path. The first discussion started with the work on autonomous vehicles. Later on, research began with respect to manipulator arms, and more recently there has been an interest in algorithmic motion planning.

Hypothesize/test

This is the earliest method for dealing with motion planning as introduced by [Pieper 68]. In general the algorithm consists of three basic steps: (a) generate a path (b) test the path for collisions; (c) if a collision is found, examine the objects colliding and propose a modified motion. The process is repeated for the modified motion. The two operations required for the algorithm are first, detection of potential collisions and second, modification of the path. The first operation is called, in the literature, collision detection or clash detection or *dynamic* interference detection. *Static* interference detection is the ability to detect non-null intersections between objects, and is now part of the repertory of most geometric modelling systems. The problem of potential collisions is handled by one of the following methods: (a) multiple interference detection [Meyer 81]; (b) sweeping volumes [Boyse 79]; (c) four dimensional interference detection [Cameron 84]; (d) constraints [Canny 85]. The main problems of this method are related to the second operation, the modification of the path, and they are a result of a lack of a global view of the problem. Path searching suffers from the drawbacks of the hill-climbing method. It can result in unnecessarily

complicated paths, or it can fail to find a path since it relies on approximations of objects. The main advantage of this method lies in its simplicity and efficiency. Collision detection can be used to test user defined paths and to help the user modify the paths.

Explicit representation of free space

Almost all the research on motion planning which falls outside the generate/test paradigm has concentrated on the problem of representing explicitly the set of locations which are collision free — the free space — and examining its connectivity. In most cases, the free space, or a subset of the free space, is represented as a graph with nodes representing connected regions of the space and edges representing adjacency between the regions. Path planning is thus reduced to graph searching. The various methods differ in the way they partition the free space and the degree of approximations they use. The fundamental drawback of the method is that explicit construction of the free space is computationally expensive. First, some of the implemented or partially implemented algorithms will be reviewed.

Grown obstacles

Widdoes [Widdoes 74] and Udupa [Udupa 77a], [Udupa 77b] developed free-space algorithms for the Stanford arm, an arm consisting of a 'boom' and a 'forearm', both approximated by cylinders. Udupa first formulated the motion planning problem in terms of an obstacle transformation. Intuitively, motion planning for a robot among obstacles is reduced to the problem of motion planning of a robot shrunken by some amount to a point among obstacles grown by the same amount into 'grown obstacles'. Grown obstacles represent the positions of the robot at which it collides and the free space is thus the space outside the grown obstacles. Using such a transformation Udupa computed an approximation of the free space for the boom as a set of adjacent rectangles of varying

dimensions. Then he used heuristic methods to determine the path for the rest of the manipulator.

The method of grown obstacles was generalised by Lozano-Perez and Wesley [Lozano-Perez and Wesley 79]. They considered the motion of a polyhedral object amidst polyhedral obstacles. They present an algorithm for the planar case with fixed orientation and heuristics for generalising to more complex cases. The general algorithm restricts the rotational motion of the polyhedron so that it can only rotate at some discrete positions.

Other algorithms related to the grown obstacle idea are described in [Nillson 69], [Moravec 80] and [Chatila 82].

Configuration Space

The grown obstacles method is stated formally in [Lozano-Perez 81], [Lozano-Perez 83], where the term configuration space was introduced in motion planning. A point in configuration space corresponds to the position/orientation (configuration) of the moving object/s. The grown obstacles, now called configuration space obstacles, correspond to sets of configurations for which the moving object overlaps one or more obstacles. [Lozano-Perez 81] calculates the exact configuration space obstacles for a cartesian manipulator under pure translational motion. One rotational d.o.f. is handled by splitting the rotation range into a number of slices and bounding the grown obstacle within each slice by a polyhedron. Free space is represented as a tree of polyhedral cells at varying resolution. The main drawbacks of the algorithm is that it is limited to Cartesian manipulators and that it uses approximations of constraints on rotations.

The method in [Brooks and Lozano-Perez 83] is based on the configuration space approach but it presents a more adequate treatment of rotations. Brooks and Lozano-Perez implemented an algorithm for moving a polygonal object in the plane with two translational and one rotational d.o.f. The algorithm uses a hierarchical subdivision of the 3-dimensional configuration space into cells which are defined to be filled, empty or mixed. The algorithm can be quite expensive

and can not be easily generalised. [Gouzenes 84] also uses the configuration space approach for planning the motion of a 2 d.o.f. revolute manipulator.

The main idea in the grown obstacles/configuration space approach is that the free space can be computed as the complement of the grown obstacles. Free space is tesselated into a number of cells and represented as a graph. Different methods use different techniques for the tesselation of the space and different approximations.

The main advantage of the configuration space method over the *local* (hypothesize/test) technique is that it guarantees to find a path in the calculated subset of the free space if such a path exists. The configuration space based algorithms described so far compute only approximations of the free space, and therefore, there is no guarantee that a path can be found even if one exists. The main problem is related to the computation of the grown obstacles in the case of rotation, because then the grown obstacles have to be embedded in a higher dimensional space and they have non-planar surfaces. All the algorithms described so far have *approximated* the grown obstacles in order to be able to deal with rotations and even then it has not been easy to consider more than one rotation. The problem of planning with 6 d.o.f. is treated in [Donald 84]. He has implemented an algorithm for planning paths using operators that slide parallel to five dimensional surfaces in configuration space and parallel to the intersection of such surfaces.

The configuration space approach and related issues are further examined in chapter 3.

Freeways

[Brooks 83b] has implemented an algorithm for a two-dimensional planner. Free space is modelled as the overlapping union of generalised cones, called freeways. Each cone has a spine and left and right width functions. A motion consists of translations along the spines of the cones and re-orientations at the points where the spines intersect.

[Brooks 83a] extended the above method for the case of 4 d.o.f. motion, three translational d.o.f. and one rotational d.o.f. Each cone is swept vertically in order to build prisms at horizontal slices through the workspace. The algorithm is reported to be fast and to run well when the space is relatively uncluttered. The main problem seems to be that it can not be easily extended to non-convex moving objects or to more than one rotational d.o.f.

Algorithmic motion planning

The characteristics of algorithmic motion planning are: (a) the algorithms are non-numeric and exact (non-heuristic); (b) the basic tools are computational geometry and asymptotically efficient techniques. It should also be noted that, in contrast to the research reviewed so far, the algorithms which will be presented next have not been implemented.

The first paper to appear from the view point of complexity theory is [Reif 79], where he sketches a polynomial time algorithm for moving a polyhedral body. The paper [Schwartz and Sharir 83b] put the idea of configuration space and grown obstacles on general mathematical foundations. Schwartz and Sharir provided a theoretical formulation for the general motion planning problem with an arbitrary number of d.o.f. and they described an algorithm for the decomposition of the free space based on Collins decomposition of semi-algebraic cells. The algorithm is polynomial time in the number of obstacle surfaces but exponential time in the number of degrees of freedom. The algorithm is only of theoretical interest because of its high time complexity ($O(n^{1024})$ for six degrees of freedom). It mainly serves as an existence proof for a polynomial time algorithm for the motion planning problem with a fixed number of degrees of freedom.

A detailed presentation and discussion about algorithmic motion planning can be found in [Yap ss]. He distinguishes between two general techniques that can be used for motion planning : decomposition and retraction. With reference to research reviewed so far, these two approaches correspond, loosely speaking, to the grown obstacle and the freeways methods respectively.

Decomposition method

The algorithm for the general motion planning problem described in [Schwartz and Sharir 83b] was entirely impractical because of its high time complexity. For this reason, variants of the decomposition method described in that paper have been considered and applied to more specific instances of the motion planning problem (lower dimensions, simpler objects , etc.) yielding much more efficient algorithms. [Schwartz and Sharir 83a] and [Sharir and Leven 85] examine the case of a straight segment moving along polygons, [Schwartz and Sharir 83c] examine the coordinated motion of discs among polygons, [Sharir and Ariel-Sheffi 84] examine the motion of a two dimensional robot consisting of several arms jointed at a common endpoint and [Schwartz and Sharir 84] the case of a rod moving amidst polyhedral walls. All these algorithms are based on the idea of recursive decomposition of the free space using *critical curves:* by considering one or more of the degrees of freedom fixed, the free space is projected into a space of lower dimension. This space is then partitioned into maximal connected regions by considering 'critical' positions, that is positions where there are multiple contacts established. The above algorithms are shown to be computationally efficient but have not been implemented. The main difference of this approach to the one in [Lozano-Perez 81], [Lozano-Perez 83] and [Brooks and Lozano-Perez 83], is that, while in the former the free space is quantised into cells,in the latter it is decomposed into maximal connected cells. The limitation of the quantisation method lies in the fact that a collision free motion may not be found, even if one exists, if the parameters chosen for the quantisation are do not provide sufficient details.

Voronoi diagrams − Retractions

The use of Voronoi diagrams for motion planning first appeared in [Rowat 79]. For a finite set of points P on the plane, the Voronoi diagram partitions the plane into regions such that all points in a region are closer to some particular point in P than to another. The points on the edges of the diagram are equidistant from

two points in P, while the points on the vertices of the diagram are equidistant from three points in P. The Generalised Voronoi diagram (GVD) [Drysdale 82] is an extension of the Voronoi diagram to deal with a set of polygons instead of a set of points: points on the GVD are equidistant from two or three polygons. The concept can be extented to Voronoi diagrams in configuration space [Donald 84], so that points on the diagram are equidistant from two or more obstacles. Planning a path on a Voronoi diagram corresponds, thus, to finding a path with maximal clearance from the obstacles.

Definitions and algorithms for the use of Voronoi diagrams in motion planning can be found in [O'Dunlaing and Yap 85], [O'Dunlaing, Sharir, and Yap 83], [O'Dunlaing, Sharir, and Yap 84], [Yap 84a], [Yap 84b], [Yap ss], [Donald 84] and [Canny 85], but an algorithm for the 6 d.o.f. motion planning problem has not been developed yet. This approach is also called the retraction approach ([Yap ss]), since motion planning is done in a lower dimensional space than the original problem, that is, the edges of the Voronoi diagram. A more detailed account of retractions is given in chapter 3.

Motion in Contact

The problem of moving one or more objects while maintaining contact with the stationary objects is examined in [Hopcroft and Wilfong 84a] where they prove a theorem concerning the existence of motion in contact. The theorem states that if there exists a free motion between two configurations where the objects are in contact, then there exists a motion such that the objects remain in contact throughout the motion. In configuration space, the space where the objects are in contact is a subspace of the free space of lower dimension. It consists of faces which intersect in lower dimensional faces, which in turn intersect in still lower dimensional faces. [Hopcroft and Wilfong 84b] considered the 2-dimensional case of rectangles which are allowed only to translate. They proved that, in this case, if there is a motion in contact between two 0-dimensional faces, then there is a motion along 1-dimensional faces. The two papers are examined in more detail in chapter 3.

Computational Complexity

The goal of computational complexity theory is to classify computational problems according to their inherent complexity. There are three important classes of problems: (a) those that can be solved in polynomial time (P); (b) those that can be solved nondeterministically (NP); (c) those that can be solved in polynomial space $(PSPACE)$ $(P \subset NP \subset PSPACE)$.

A problem is *hard* for a class if every problem in that class is reducible to it. So, if a problem is NP-hard it means that it is at least as hard as any other problem in NP. Proving that a problem is hard in a class means that we derive a lower bound for its inherent complexity. An upper bound on its complexity can be placed if we can prove that a problem is *in* a class. If we can place both lower and upper bounds, then the problem is *complete* for that class. For example, a problem is NP-complete if it is in NP and it is NP-hard.

A deterministic polynomial time algorithm is said to be efficient while a deterministic exponential time algorithm is inefficient. If a problem can only be solved in exponential time then we say that it is computationally intractable.

$PSPACE$ contains many problems for which no efficient solutions are known. Therefore a $PSPACE$-hard problem is computationally intractable for the case of an algorithm that will solve all problem instances.

With respect to the general motion planning problem with an arbitrary number of d.o.f. the following lower bounds have been found for the reachability problem, that is the decision whether a motion exists from some initial to some final position: [Reif 79] showed $PSPACE$-hard lower bound for a certain many-jointed three-dimensional motion planning problem. [Hopcroft, Schwartz, and Sharir 84] showed $PSPACE$-hardness for the case of moving rectangles. [Hopcroft and Wilfong 84b] showed that the problem is in $PSPACE$ for the case of rectangles which can only translate. [Hopcroft, Joseph, and Whitesides 84] showed that the problem is NP-hard for a planar many-linked arm. Finally, [Spirakis and Yap 84] showed NP-hardness for moving many discs.

With respect to the problem with fixed degrees of freedom, sometimes called the basic motion planning problem, it is known ([Schwartz and Sharir 83b]) that a polynomial time algorithm exists. As an indication of the complexity of some of the algorithms for specific motion planning instances, the following results are summarised, where n is the number of 'walls':

- 2D,discs, polygonal obstacles: $O(nlogn)$ for one moving body $O(n^2)$ for two moving bodies [Yap 84a]

- 2D, straight segment, polygonal obstacles : $O(n^2logn)$ [Sharir and Leven 85]

- 2D, 'k-spiders', polygonal obstacles: $O(n^{k+4})$ [Sharir and Ariel-Sheffi 84]

2.2.3 Summary and Conclusions

While reviewing research in the field of motion planning, it was indicated that the following distinctions can be made among the various approaches: (a) explicit representation of free space vs. hypothesis/test; (b) decomposition vs. retraction; (c) critical curves vs. grown obstacles. An algorithm can be also characterised according to whether it is heuristic or non-heuristic (complete), what kind of approximation it uses, either for modelling the objects or for computing the free space, and according to whether it is local or global. The concept of local motion planning has been used in different ways. [Lozano-Perez 82] makes the distinction local vs. explicit free space, local implying the absence of global information about the free space. [Donald 84] makes the distinction local vs. global to differentiate between the decomposition technique and the retraction technique. In the latter case, global information on the connectivity of the free space is available. Finally, [Yap ss] uses the term local planning to imply local experts which are able to find a path in specific situations, such as moving through a door.

Research has been carried out on both theoretical and practical algorithms. The general motion planning problem has been proven to be computationally

intractable. However, a polynomial time algorithm has been described for the basic motion planning problem, that is the problem of planning for a fixed robot system [Schwartz and Sharir 83b]. The complexity of that algorithm makes it inefficient for any practical use. In order to reduce the complexity of the problem and to produce efficient and practical algorithms, researchers have concentrated on more specific instances of motion planning. The complexity of the problem can be reduced if (a) a 2-dimensional world is considered; (b) the number of degrees of freedom is reduced, either by considering manipulators with less than six d.o.f. or by considering some of the d.o.f. fixed; (c) the geometry of the objects is simplified (discs, rectangles, rods, polyhedra etc). Although some of these simplifications may seem unrealistic, they can be useful, firstly, because they provide insight into some of the issues of the problem, and secondly, because they yield efficient algorithms which can be of practical use. For example, enclosing the objects in spheres could be a useful approximation in the case of relatively uncluttered environments.

It has been indicated that a lot of research on motion planning has adopted the method of considering a generally higher dimensional space, called configuration space, where each point corresponds to a particular configuration of the objects. A path in configuration space corresponds to some motion of the objects. The regions in configuration space representing configurations which are forbidden are called configuration space obstacles. The motion planning problem is, then, reformulated as the problem of finding a path in configuration space outside the obstacles. This approach to motion planning has been particularly helpful in formalising the motion planning problem and developing the necessary conceptual and mathematical tools. The main difficulties with this approach are associated with the explicit computation of the configuration space obstacles, especially when more than one rotational degree of freedom is allowed.

We can conclude that algorithms for various instances of the motion planning problem have been considered both from the theoretical and practical point of view. Relatively efficient algorithms exist for some specific classes of problems, mainly for motion planning in two dimensions or motion planning with only one

rotational degree of freedom. At the moment, because of the inefficiency of the algorithms which employ an explicit representation of the free space, it is the hypothesize/test method, if any, that is used in practice.

2.3 Fine-Motion Planning

2.3.1 Uncertainty

One of the main problems in robot planning in general and motion planning in particular is to enable the robot to achieve a desired goal despite the presence of uncertainty in the environment. The main sources of uncertainty are:

- Errors in the input model (modelling errors): imperfect parts, positioning errors.

- Errors in control: position and sensor errors.

- Computational inaccuracy.

The above description of uncertainty covers the cases when the actual state of the world differs from the model or the computed state within some known bounds. For example, parts are usually manufactured within stated tolerances. However, there could be defective parts outside the specified tolerances or there could be events which could not have been foreseen during planning. In the rest of this thesis, we will concentrate on the issue of 'expected' uncertainty, where 'expected' implies that it can be bounded. In the case of unexpected events, the problem is usually referred to as error recovery [Gini, Gini, and Somalvico 80].

Motion planning is, therefore, a problem of planning with incomplete information. Research in A.I. has considered the problem of planning with uncertainty. In the field of robotics the issue has been more neglected. [Taylor 76]

and [Brooks 82] have proposed and implemented schemes for dealing with uncertainty. These are reviewed in section 2.3.3. Uncertainty resulting from tolerances of parts has also been considered ([Requicha 83], [Fleming 85]).

When dealing with gross motion, uncertainty can be dealt with by enlarging the obstacles and the robot by a given amount. This method can be satisfactory if the environment is not too cluttered. Unfortunately, we can not use this technique when dealing with grasping or parts mating, since the objective is to bring some objects into contact. In this case, uncertainty can be reduced by introducing force sensing.

2.3.2 Compliance and Force Control

Most robots are position controlled. Such robots can successfully perform tasks which can be adequately expressed as a sequence of positions. Spot welding, machine loading and spray painting fall in this category. Moreover, the positional accuracy required for these tasks is generally lower than the robot's accuracy. There are, however, manipulator tasks which cannot be adequately described as a sequence of positions. Examples of such tasks are inserting a peg in a hole, sliding along a surface, closing a door and many others. Parts mating falls within this class of tasks. The common characteristic of the above examples is that they require motions which have to comply with certain physical constraints. Motions which are constrained by external constraints imposed by the geometry of the task are called *compliant motions*. Consider as an example the task of sliding along a surface from some initial to some final position. This task could be described as a sequence of positions, but it could only be accomplished successfully if the model, the trajectory planner and the controller are perfectly accurate. In the presence of uncertainty the task could only be performed successfully if the trajectory is constantly modified by tactile information. The ability of the manipulator to perform compliant motions is, therefore, closely related to the introduction of sensing into the manipulator program and the controller.

In compliant motion the manipulator has to maintain contact between some surfaces. Another class of motions which are also constrained by external constraints, are *guarded motions* [Will and Grossman 75]. These are motions used when the manipulator is about to establish some contact, for example when placing a box on a table. A *compliant-guarded motion* is a motion during which some contacts have to be maintained while others are about to be established or broken. As an example, consider the task of inserting a peg in a hole until the tip of the peg touches the bottom of the hole.

There are two primary methods for producing compliant motion. The first one is using passive mechanical compliance built in the manipulator. An example of passive compliance is the Remote Centre Compliance (RCC) device used for insertion tasks [Whitney and Nevins 79]. The second method is the use of active compliance in the control loop of the manipulator. This method is called *force control*. Its main advantage over passive compliance is its programmability. There are two different approaches to force control: explicit feedback and hybrid (position/force) control. The explicit feedback scheme is based on the idea of generalised stiffness or generalised damping. Sensed forces are sent back to the position controller which corrects the position accordingly [Nevins and Whitney 74], [Whitney 82], [Salisbury 80], [Hanafusa and Asada 77]. The hybrid control scheme controls positions along some specified d.o.f. and independently controls forces along the remaining d.o.f. ([Paul and Shimano 76], [Mason 81], [Raibert and Craig 81]).

A thorough review on the issues of compliance and force control can be found in [Mason 82a]. The point that we would like to make here is that pure position control is not adequate for fine motions because of uncertainty. Compliant and guarded motions require force control and the use of force sensors. Using compliant and guarded motions, we make use of the geometry of the environment to guide the motions. In the previous section the goal of motion planning was to find a path as a sequence of positions. In this chapter, where we are dealing with fine-motion planning, the objective is to describe a path not only in terms of positions but also in terms of forces.

2.3.3 Approaches to Parts Mating

The various methods for dealing with the parts mating problem can be categorised in five different classes: static analysis, information approach, skeleton programs, learning strategies and configuration space approach. Almost all the research has concentrated on the peg-in-hole insertion problem and all comments will be made with reference to that problem.

Static Analysis

The first method concentrates on the analysis of the geometry and statics of the task in detail ([Simunovic 75], [Drake 77], [Ohwovoriole 80], [Whitney 82]). Through this analysis, the conditions under which 'jamming' and 'wedging' can occur are formulated. The Remote Center Compliance (RCC, [Whitney 82])has been built as a result of this analysis. Similar analysis is described in [Inoue 74], [Goto, Inoyama, and Takeyasu 80] and led to the development of heuristic strategies for the peg-in-hole insertion.

Information Approach

Simunovic [Simunovic 79] uses what he calls an information approach to parts mating. Having defined the assembly problem as the problem of accurately positioning the parts with respect to each other, he then proceeds to develop a method for correcting the errors in the relative position of the parts being assembled. The method is based on the gathering of information generated during the assembly process. Both positional information and force information resulting from the interactions of the objects is used. This information is analysed and fed back into a positioning device.

This approach can be seen as an extension of the static analysis method, since it also relies on the analysis of the geometry of a task and is, therefore, highly dependent on the particular task being examined. Its fundamental limitation

is that it views the assembly task as a positioning problem and pure position control is assumed.

Skeleton Programs

Taylor [Taylor 76] and Lozano-Perez [Lozano-Perez 76] were the first to explore the area of automatic synthesis of fine motions, the *fine-motion planning* problem. The key idea is that partially specified strategies, known as skeletons, are used. Skeletons are parameterised robot programs for particular tasks. They include motions, error tests and computations, but many parameters for motion and tests remain unspecified.

Taylor [Taylor 76] developed an algebra system which deals with position constraints, which model the effect of error and uncertainty. These error estimates are propagated and they are used to make decisions for choosing a strategy and filling in the values of the parameters.

The method of propagating constraints was further extended by [Brooks 82]. This system is able to handle symbolic constraints and it can also propagate constraints backwards, so that besides error estimates the value of plan variables can be also specified. The system developed by Brooks deals in general with the problem of robot planning with uncertainty and it is not specific to the fine-motion planning problem.

[Lozano-Perez 76] proposed a method for selecting the parameters in a strategy by computing the range of positions that the relationship among the parts entails, as specified in the strategy.

Learning Strategies

The approach taken by Dufay and Latombe [Dufay and Latombe 84] is also based on strategies. It differs significantly from the two previously described methods ([Taylor 76] and [Lozano-Perez 76]) in the fact that strategies are expanded into programs through a sequence of experiments. Specifically, partial

local strategies, described by a collection of rules, are extended to full strate-
gies by considering the multiple traces of executed plans. The process consists
of two phases: the training phase, which generates execution traces, and the
induction phase which transforms traces into programs. The system has been
implemented.

By contrast to Taylor's method ([Taylor 76]), which is also based on error es-
timates, this system *explicitly* tests for different motion outcomes resulting from
uncertainty. For this reason, it is not constrained by the worst case uncertainty
behaviour which might lead to overcautious plans. One possible limitation of
the system is that it requires knowledge of the actual contacts achieved by each
motion. Such information could be ambiguous because of errors in measuring.

The last two approaches, skeletons and learning strategies share some com-
mon characteristics. The most important assumption that they make is that
there is a basic repertory of common operations which can be described in some
abstract way. The synthesis of a fine-motion program would then require the se-
lection of an appropriate strategy and of the values of the parameters. The main
weakness of these approaches lies precisely in this assumption. Small changes in
the geometry can affect drastically the structure of the strategy. Lozano-Perez
([Lozano-Perez 82]) presents examples of various peg-in-hole insertions where a
slight change in the geometry of the parts would require a different strategy.

Configuration Space Approach

[Lozano-Perez, Mason, and Taylor 84] present a formal approach to the problem
of planning fine-motions, based on the configuration space approach to motion
planning. The task is transformed into a task in configuration space. In the
transformed problem, surfaces represent set of positions which are constrained by
external constraints (contacts). Motion along these surfaces, called C-surfaces,
corresponds to compliant motion, while motion from one surface to another cor-
responds to compliant- guarded motion as described in [Mason 81]. Given a goal,
described as a set of positions on some C-surface, and bounds on errors in position

and sensing, they propose a method for computing sets of positions from which the goal can be successfully reached using a *single* compliant-guarded motion. Such a set of positions is called the pre-image of a goal. Mason in [Mason 83] addresses some of the theoretical issues of this formalism. The formalism has been partially implemented as described in [Erdmann 84]. The main weakness of this approach lies in the difficulties inherent for a full implementation.

2.4 Relation of this Thesis to Reviewed Work

The problem that this thesis focuses upon is that of planning parts mating operations. It has been established that in planning parts mating operations the forces arising from contact and the presence of uncertainty are of considerable importance and need to be taken into account. It is, nevertheless, possible to approach the problem in a manner such that the issues of forces and uncertainty are considered only indirectly. Specifically, insofar as the direction of forces can be inferred from the geometry of the environment, it is possible to treat the problem of planning parts mating operations as a problem in *geometric planning*.

The approach that is taken in this thesis is to bring the parts together through a sequence of motions such that the parts are first brought into contact and then their positions are further constrained by establishing more contacts until the desired relative positions are attained. Therefore, planning parts mating operations becomes an instance of the general motion planning problem, that of planning the motion of an object while maintaining contact with a set of stationary objects. This alternative approach underlies the specific emphasis that I have chosen to place in the review of related work. In this Section, I shall first relate the work in this thesis to work in the field of motion planning by specifying what is the particular problem that concerns my work and how the approach relates to the basic approaches to motion planning outlined in Section 2.2.2. Then, I shall consider the relation of this work to other approaches to parts mating.

In Section 2.2.1 a formulation of the *general* motion planning problem was presented and various of the characteristics according to which different families of problems arise were indicated. It was also indicated that there is strong evidence that the solution to the general motion planning problem is computationally intractable. On the other hand though, it was shown that the *basic* motion planning problem — i.e. the problem of planning with a fixed robot system — is tractable.

The observations made in Section 2.2 with regard to the general motion planning problem also hold true in the case of the motion in contact problem, that is, the problem of moving one or more objects while maintaining contact with obstacle surfaces. That is to say, in the case of the motion in contact problem as well, there exist different families of problems while the evidence strongly suggests that the problem in its general form is computationally hard.

In view of these observations there is a strong case for narrowing the problem down. Specifically, I shall consider the problem of the motion in contact of an arbitrary, rigid, three dimensional polyhedral object amidst polyhedral obstacles. The moving object then, will have three translational and three rotational degrees of freedom.

In the relevant literature, the problem of planning in three dimensions remains open. Moreover, the problem of motion planning with three rotational degrees of freedom is far from having been considered extensively. In effect, the only detailed examination of this problem is to be found in [Donald 84]. With the exception of [Schwartz and Sharir 83b], where the general motion planning problem is considered, in all the work that has been done in this area, the objects are taken to be polygonal or discs, in the two dimensional case, or polyhedral or spheres, in the three dimensional case, or more specific instances of these. For example, some of the classes of objects which have been considered are discs, straight segments, rectangles and spheres. In this thesis, the restriction to polyhedral objects will be made, which is not an unreasonable one insofar as a large class of objects can be modelled as polyhedral.

Having, thus, specified the particular problem that will be considered, I shall

next outline the approach in relation to motion planning. As it has been indicated, almost all the approaches to motion planning are based on the transformation from physical space to configuration space [Lozano-Perez and Wesley 79]. In configuration space, every point represents the position and orientation (configuration) of the moving object/s. Those points in configuration space representing positions for which one or more objects overlap form the configuration space obstacles. The problem of moving objects among obstacles can then be reformulated as the problem of moving a point among obstacles in configuration space. In this thesis, the configuration space approach to motion planning is followed. Since we are interested in motion in contact, we are interested in that subspace of the configuration space which corresponds to the set of configurations for which the objects are in contact, the contact space. The contact space corresponds to the surface of some generalised configuration space obstacle.

[Lozano-Perez, Mason, and Taylor 84] proposed a method for planning fine-motions based on the configuration space approach. They indicated that fine-motions correspond to motions on the surfaces of the configuration space obstacles. My approach differs significantly from [Lozano-Perez, Mason, and Taylor 84] and [Erdmann 84] in the ways listed below:

1. Representation of the space: the method employed by Lozano-Perez et al. is based on an explicit representation of the configuration space obstacles. However, when rotational degrees of freedom of motion are permitted, only approximations of the grown obstacles are computed. This thesis presents an alternative representation for the configuration space obstacles.

2. Forces and uncertainty: by contrast to the approach followed in [Lozano-Perez, Mason, and Taylor 84], [Erdmann 84] in this thesis these issues are only treated implicitly.

3. Objective: while [Lozano-Perez, Mason, and Taylor 84] and [Erdmann 84] compute a set of initial positions for which a *given single* motion will guarantee the attainment of the desired state, I will aim in *deriving a*

sequence of motions which will achieve the transition from some initial to some desired state.

[Donald 84] also considers the problem of sliding along the surfaces of the configuration space obstacles, called C-surfaces. The principal difference between the approach taken by Donald and that employed in this thesis can be located at the level of representation of the C-surfaces and for the computation of their intersections. Specifically, Donald's approach is based on the use of 'C-functions' in order to model the C-surfaces and to compute, algebraically, their intersection. By contrast, the approach taken in this thesis is premissed upon the use of spatial relationships both for the representation of the surfaces and the computation of their intersections. By the use of a spatial reasoning system the computation, by algebraic means, of the intersections of the C-surfaces is no longer required. To these differences I shall return in Chapters 3 and 4.

The problem of motion in contact has also been considered in [Hopcroft and Wilfong 84a], [Hopcroft and Wilfong 84b]. The approach that is followed in this thesis is similar to the approach in [Hopcroft and Wilfong 84b] insofar as the contact space is decomposed into faces of various dimensions and the connectivity of these faces is considered. The work by Hopcroft and Wilfong considers the two-dimensional case of rectangles which are only allowed to translate. In this thesis, the approach will be generalised and modified to account for the problems resulting from the introduction of rotations. In addition, an implementation of the algorithm for the decomposition of the space will be presented based on a geometrical reasoning system which reasons about spatial relationships among the features of the objects.

In the beginning of this Section it was argued that the parts mating problem can be regarded as a problem in geometric planning. Geometric planning is premissed upon information about the geometry of the environment. In this thesis, this information is formulated in terms of spatial relationships among features of objects. Such a formulation of the problem is the major distinction between the work described in this thesis and the reviewed work. The reasoning system

developed in this thesis is an extension of the inference engine of the RAPT system [Popplestone and Ambler 83], [Corner, Ambler, and Popplestone 83]. The system is extended to deal with a large number of spatial relationships among polyhedral objects. Spatial relationships have been used so far in order to describe an assembly task. The objective of the reasoning system has been to deduce the relative positions of the objects. Here, spatial relationships are used in a different way. They are used firstly, to decompose and model the search space during the formulation of the plan and secondly, to extract information about positions, forces and motions during execution of the plan.

Chapter 3

Theory of Motion in Contact

3.1 Introduction

In this chapter I shall consider the theoretical issues related to the problem of moving a 3-dimensional polyhedral object while maintaining contact with a set of stationary polyhedral objects. To address this problem, the notion of *contact space* will be considered. The contact space is the set of locations of the moving object for which it is in contact with, but does not overlap, one or more stationary objects. The objective is to decompose this space into a number of *components* and to build the connectivity graph of the space, that is a graph where vertices represent the components and edges represent adjacency between components. In this way, the motion planning problem is reduced to a graph searching problem.

The approach taken is based on the configuration space approach to motion planning. In configuration space the moving object is shrunk to a point, representing the position of some reference vertex of the object, and the stationary objects are grown into configuration space obstacles, representing the locations where the moving object overlaps one or more stationary objects. The problem of moving an object among obstacles is, thus, transformed into the problem of moving a point among the grown obstacles in a higher dimensional space. Informally, the contact space is the boundary of the grown obstacles and, therefore, a surface of dimension less than n, where n is the dimension of the configuration space.
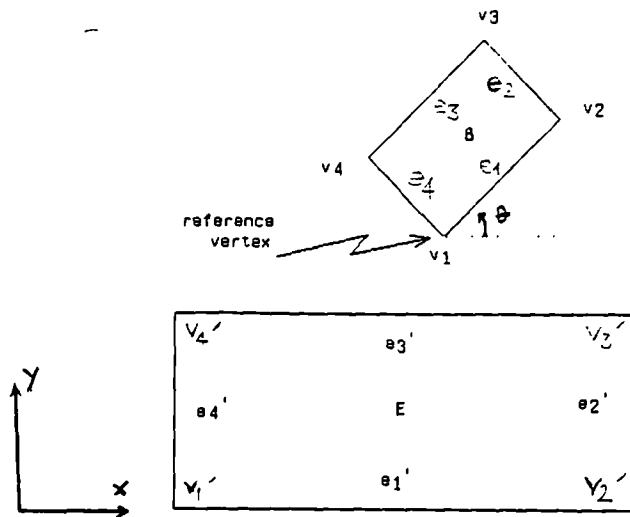
Object B is the moving object, v_i are its vertices and e_i its edges. Its location is specified by the coordinates of vertex v_1 and the angle θ. Object E is the stationary object, v_i' are its vertices and e_i' are its edges.

Figure 3-1: A 2-dimensional example
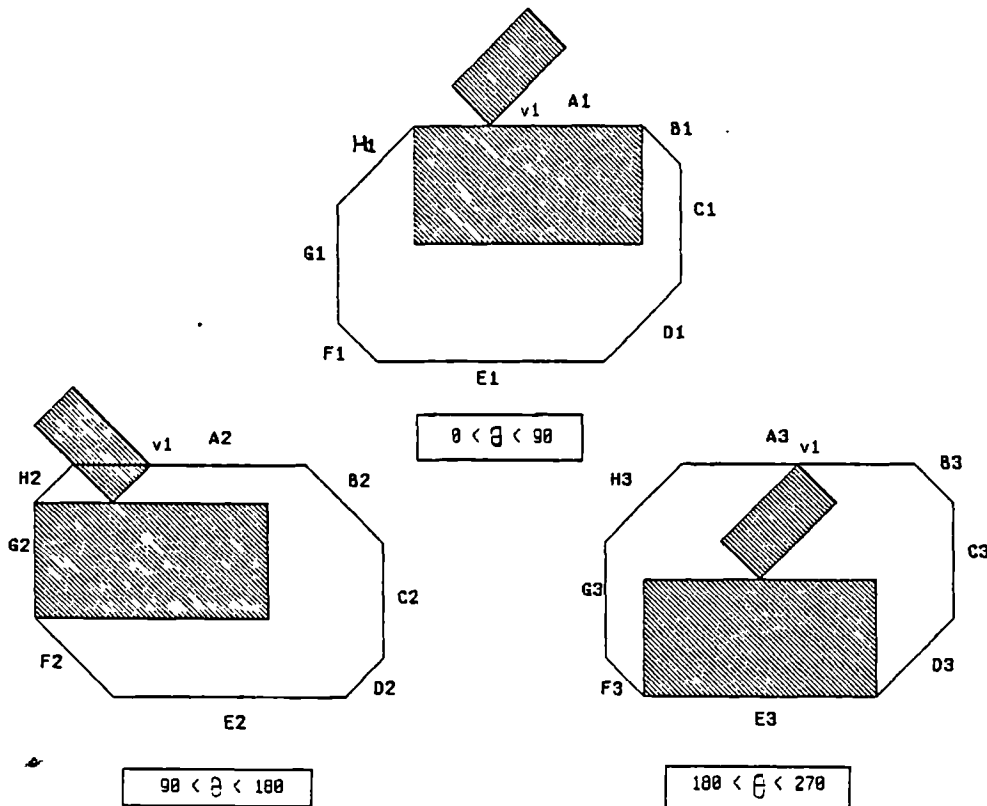
For a given orientation θ the configuration of B is specified by the coordinates of v_1. The grown obstacle consists of configurations for which B and E overlap. For a given orientation, the grown obstacle is obtained by sliding B around E and marking the coordinates of v_1 ([Lozano-Perez 83]). A_1 consists of locations for which v_1 touches e_3', B_1 of locations for which v_3' touches e_4, A_2 of locations for which v_4 touches e_3' etc.

Figure 3-2: Slices of configuration space obstacle
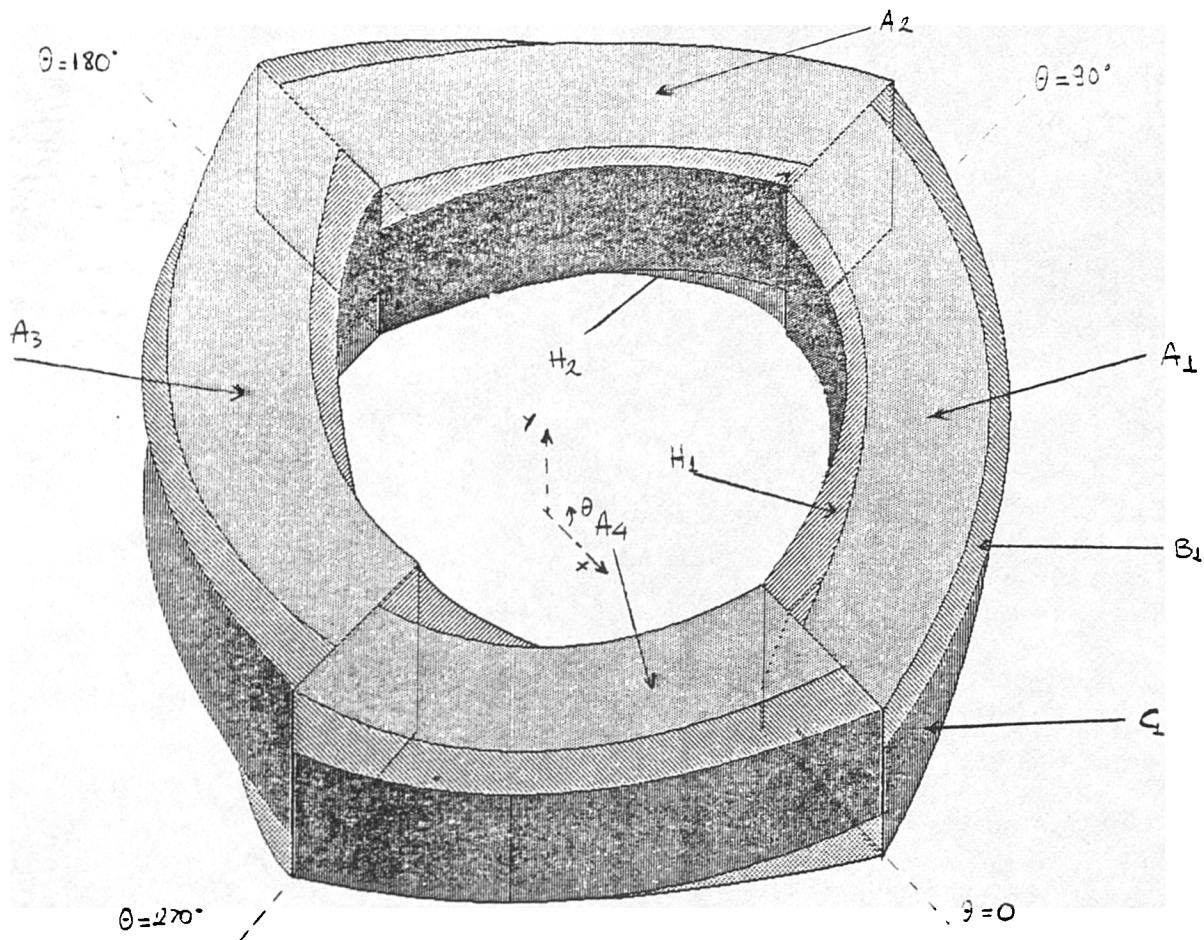
A point on the surface of the grown obstacle represents a configuration for which B and E are in contact. The x and y coordinates represent the location of v_1. The angle θ represents the orientation of B. Slices of the grown obstacle for various values of θ are shown in Figure 3-2.

Figure 3-3: Configuration space obstacle

Let us illustrate the approach by considering a two-dimensional example. In Figure 3-1 the moving object B and one stationary object E are shown. There are three degrees of freedom of motion, two translational, x and y, and one rotational θ. The configuration space is, therefore, three dimensional. Figure 3-2 shows slices of the grown obstacle for various values of rotation angle θ. The whole grown obstacle is shown in Figure 3-3. Informally, the boundary of the grown obstacle is the *contact space*.

It can be seen from Figure 3-3 that the contact space consists of faces of various dimensions: 0-dimensional faces, which will be referred to as vertices,

1-dimensional faces (edges), and 2-dimensional faces. Each face consists of configurations which satisfy some specific contact constraint among the features of the objects. For example, the face labeled A_1 consists of configurations which satisfy the constraint "vertex v_1 of B is on edge e'_3 of E" [1], while for the face A_4 the constraint is "vertex v_2 of B is on edge e'_3 of E". The dimension of a face is equal to the degrees of freedom of motion imposed by the contact constraint. For example, in the case of vertex to edge contact in two dimensions, there are two degrees of freedom, one rotational and one translational. Informally, faces meet in faces of lower dimensions and configurations lying on the intersections of faces satisfy the conjunction of the constraints. For example, the intersection of faces A_1 and A_4 consists of configurations for which both v_1 and v_2 are on the edge e'_3, as shown in Figure 3-4. This set of configurations lies on a 1-dimensional face of the space, which belongs to the boundary of A_1 and to the boundary of A_4.

The 'topological' structure of the contact space can be represented as a graph, where the nodes in the graph represent the faces of the space and arcs represent adjacency relationships among faces. A fragment of the graph for the above example is shown in Figure 3-5.

Having constructed such a graph of the space, we can plan a motion in contact as a path in this graph, where the only type of motion permitted is a motion from a face to another face which is on its boundary, or vice versa. In particular, it will be shown how a path can be found along the edges of the contact space. In order to be able to derive such a plan we have to ensure that the vertices of the space are edge-connected.

As can be seen from the above example, even for the case of two very simple 2-dimensional objects, the 'geometry' of the contact space is complicated, since rotations result in curved surfaces. The approach taken is to use a system which

---

[1] In the rest of this thesis, features of the stationary object(s) will be marked by a dash, e.g. e'_1.
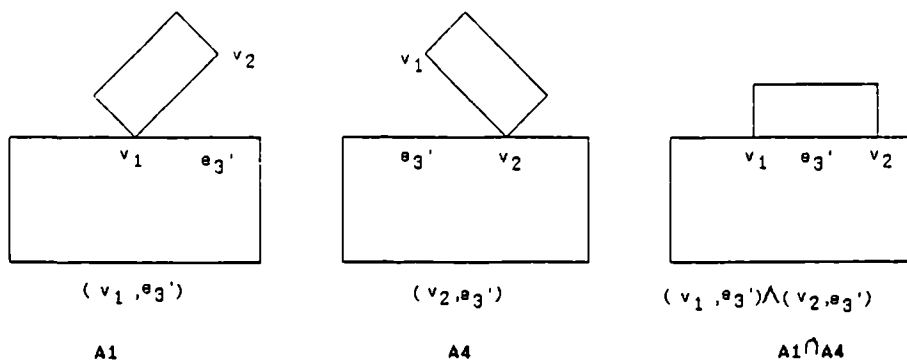
A configuration for which both v_1 and v_2 are on e'_3 lies on the intersection of the faces A_1 and A_4 (see Figure 3-3).

Figure 3–4: Intersection of the faces of grown obstacle
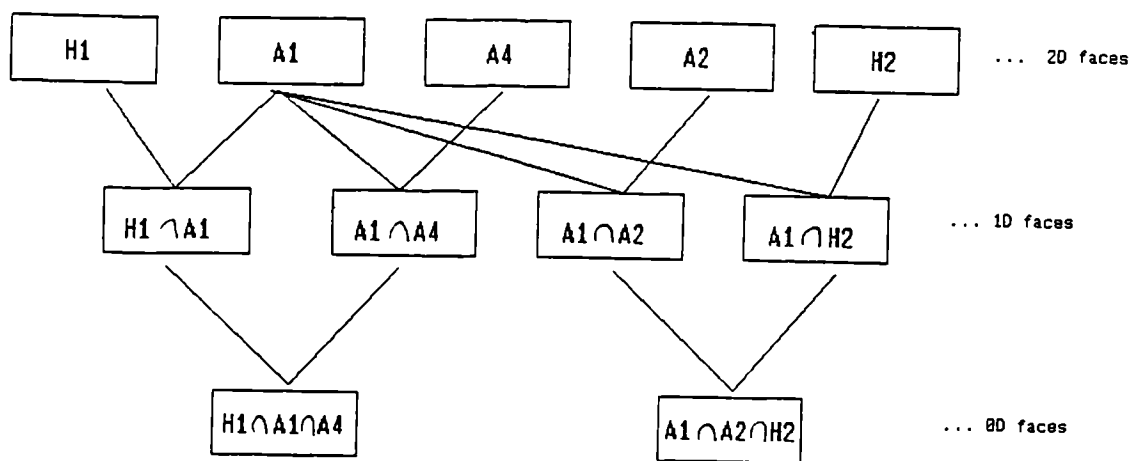
Fragment of the graph of the faces of the contact space for the objects in Figure 3-1. A_1, A_2, H_1 etc are 2-dimensional faces. Their intersections, $A_1 \cap A_2$ etc, are 1-dimensional faces (edges) and the intersections of the edges are the vertices of the contact space.

Figure 3–5: Topological structure of contact space

can handle interactions among features of objects in order to derive the 'topology' of the contact space, that is its faces and their connectivity.

Some of the questions that will be addressed in this chapter are the following:

- How can we use interactions among features of objects to decompose the contact space into various dimensional faces?

- Is there a decomposition such that the vertices of the space are connected by edges?

- How do we find intersections of faces?

The chapter is structured as follows: in Section 3.2 some basic concepts are defined. In Section 3.3 the concept of decomposition of a topological space is formally introduced and a decomposition with specific properties is considered (cell complexes). In Section 3.4 two slightly different methods for decomposing the contact space are developed and, finally, in Section 3.5 an algorithm for the decomposition is presented.

3.2 Basic Concepts

3.2.1 Models of objects

A rigid solid may be modelled as a subset of the 3-dimensional Euclidean space, $\mathcal{E}^3$. However, not all subsets of $\mathcal{E}^3$ model physical solids. In [Requicha 77] the properties of a 3-dimensional physical solid are identified, and the mathematical implications of these properties are considered. It is argued that suitable models of solids are subsets of $\mathcal{E}^3$ that are bounded, closed, regular, and semi-analytic or semi-algebraic. These sets are called R-sets. In the rest of this section a brief overview of R-sets is presented. A more thorough account can be found in [Requicha 77] and [Brown 81]. Following this, the definition of the models of the objects considered in this research is presented.

A subset X of $\mathcal{E}^3$ is a *closed regular set* if it equals the closure of its interior, i.e. $X = cl(int(X)$, where $int(X)$ denotes the interior of a set X, and $cl(X)$ its closure. The *regularised union* of two sets, denoted by the symbol $\cup*$, is defined by

$$X \cup *Y = cl(int(X \cup Y)).$$

The regularised intersection and difference are similarly defined. The class of compact (closed and bounded) regular sets is closed under the regularised set operators. It can be used to model solids which are of finite extent. However, there is no guarantee that the solids are finitely describable or that their boundaries are 'well-behaved'. For this reason, some sub-class of compact regular sets has to be considered.

Semi-algebraic sets are set-theoretical combinations of regions of $\mathcal{E}^3$ whose points satisfy inequalities of the form $f_i(x, y, z) \geq 0$, where f_i is any polynomial function on $\mathcal{E}^3$. A *semi-analytic set* is similarly defined. The only difference is that f_i is any analytic function. Semi-algebraic sets are triangulable and, therefore, finitely describable. Also, they are closed under the regularised

set operations. Compact, regular semi-algebraic sets (R-sets) are finite sub-polyhedra of $\mathcal{E}^3$, that is, they are polyhedrally embedded in $\mathcal{E}^3$, and, therefore, their boundaries are 'well-behaved'. Intuitively, R-sets are curved polyhedra with 'well-behaved' boundaries. They are not necessarily connected and they may have holes.

In this thesis, the problem considered is the motion of a 3-dimensional polyhedral object in contact with a set of 3-dimensional polyhedral obstacles. The models of objects which will be considered are subsets of R-sets. First of all, instead of semi-algebraic sets, I will consider sets which are regularised set-theoretical combinations of regions of $\mathcal{E}^3$ whose points satisfy inequalities $f_i(x, y, z) \geq 0$, where f_i are *linear* functions. Secondly, I will consider connected subsets of $\mathcal{E}^3$.

Definition 3.2.1 *An object is a compact, regular, connected 3-dimensional polyhedron with planar faces.*

Only one object is allowed to move, and it is called the *moving object*, B. Stationary objects are called *obstacles*, A_i. It will be assumed that all obstacles are supported by some other obstacle, that is to say, the obstacles are connected, i.e. $\forall i\ \exists j : A_i \cap A_j \neq \emptyset$. As it will be shown in future sections, by making this assumption, we are able to guarantee the existence of a motion in contact between any two locations for which the moving object is in contact with the obstacles.

Definition 3.2.2 *The environment E is the regularised union of the stationary objects (obstacles),* $E = \bigcup_j *A_j$

Since the obstacles are connected, and the class of compact regular sets is closed under the regularised set operations, it follows that E is also an object.

3.2.2 Transformations, Configurations, Configuration Space

Transformations are used to define the locations of the objects in space.

Definition 3.2.3 *A rigid transformation, Z, in $\mathcal{E}^3$ is a mapping of $\mathcal{E}^3$ onto itself which preserves distances and signed angles.*

A transformation Z can be specified by the 6-tuple $(x, y, z, \theta, \phi, \psi)$. The space of rigid transformations in $\mathcal{E}^3$ is thus 6-dimensional. A common method for representing transformations is by means of a 4×4 matrix which is the product of a rotation and a translation. Transformations are presented in detail in Section 4.2.1.

Each object has a coordinate system affixed to it. The point at which the origin of the coordinate system is affixed is called the *origin* of the object. The *configuration* or *location* of the object is specified by the position of its origin and the orientation of the affixed coordinate system with respect to the reference coordinate system in $\mathcal{E}^3$. ·

Definition 3.2.4 *The* configuration *or* location *Z of an object is a rigid transformation from the reference coordinate system to the coordinate system of the object. The configuration of an object is specified by a vector*

$$Z = (x, y, z, \theta, \phi, \psi).$$

Definition 3.2.5 *The* configuration space *C is the space of all configurations of the moving object.*

Thus in our case the configuration space is 6-dimensional.

Let Z be a point in configuration space. Then for any point p in $\mathcal{E}^3$, $p@Z$ will denote the Z transformation of p. For any object B, or any subset of an object, $B@Z$ denotes the region in $\mathcal{E}^3$ occupied by object B at configuration Z, that is

$$B@Z = \{p@Z | p \in B\}.$$

3.2.3 Contact Space

In this section various subsets of the configuration space will be classified according to body interference. The following definitions are derived from the definitions presented in [Hopcroft and Wilfong 84a].

Definition 3.2.6 *The moving object B at configuration Z and the environment E* **overlap** *if the intersection of their interiors is non-empty. Let OVERLAP denote the set of configurations at which B overlaps E, i.e.*

$$OVERLAP = \{Z | int(B@Z) \cap int(E) \neq \emptyset\}.$$

Note that *OVERLAP* is open in configuration space.

Definition 3.2.7 *A configuration Z is* **legal** *if B@Z does not overlap the environment. Let LEGAL denote the set of all legal configurations,*

$$LEGAL = \{Z | int(B@Z) \cap int(E) = \emptyset\}.$$

Note that *LEGAL* is the complement of *OVERLAP* in configuration space and hence closed.

Definition 3.2.8 *The moving object B at configuration Z is* **in contact** *with the environment if the intersection of B@Z and E is nonempty but B does not overlap E. The contact space, CONTACT, is the set of all such configurations,*

$$CONTACT = \{Z | int(B@Z) \cap int(E) = \emptyset \wedge B@Z \cap E \neq \emptyset\}.$$

It has be shown [Hopcroft and Wilfong 84a] that

$$
\begin{aligned}
CONTACT &= cl(OVERLAP) - OVERLAP \\
&= cl(OVERLAP) \cap LEGAL \\
&= bdry(OVERLAP),
\end{aligned}
\tag{3.1}
$$
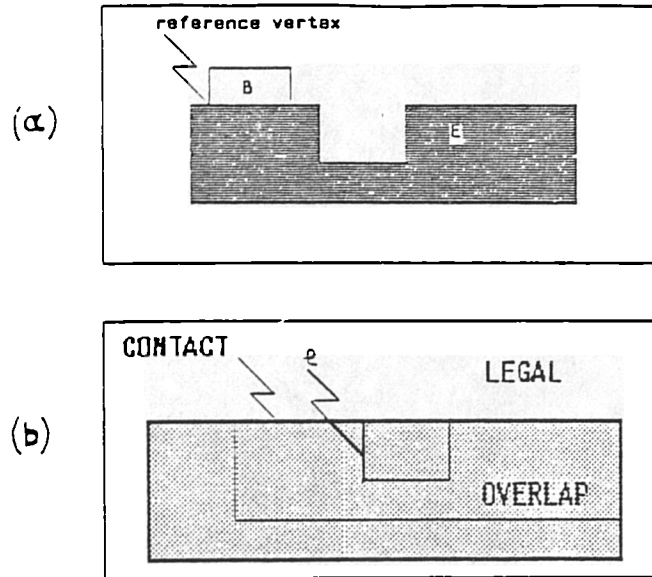
B is 2-dimensional object which is allowed only translational motion and *E* is stationary. The dimensions of *B* are such that it fits exactly into the hole of *E*. In (b), the various subsets of the configuration space are shown. The configurations on *e* are in *CONTACT* but not in *cl(OVERLAP)* [Hopcroft and Wilfong 84a].

Figure 3–6: The subsets of the configuration space

where *bdry(A)* denotes the boundary of a set *A*.

The set *cl(OVERLAP)* is what is sometimes called the grown obstacle [Lozano-Perez 81]. It has to be noted that the contact space is the boundary of *OVERLAP* (Equation 3.1) and this is different from the boundary of *cl(OVERLAP)*. The contact space is not, therefore, the boundary of the grown obstacle. This observation is illustrated in Figure 3–6. Figure3–6a shows a two dimensional object *B*, which is allowed to translate but not to rotate, and a two dimensional stationary object *E*. The dimension of *B* is such that it can exactly slide in the hole. In Figure 3–6b the corresponding subsets of configuration space are shown. While configurations where *B* is in the opening of *E* are in *CONTACT*, they are not in the boundary of *cl(OVERLAP)*.

3.2.4   The Existence of Motion in Contact

In Section 2.2.2 it was mentioned that [Hopcroft and Wilfong 84a] have considered the problem of motion in contact and have proved a theorem concerning the

existence of motion in contact. In its simplest form the theorem states that if two objects in contact can be moved to another configuration for which they are in contact, then there is a way to move them such that they remain in contact throughout the motion.

Let us first give the formal definition of a motion:

Definition 3.2.9 *A motion or path m in configuration space between two configurations Z_1 and Z_2 is a continuous function from the unit interval to the configuration space C,*

$$m : [0,1] \mapsto C$$

such that $m(0) = Z_1$ and $m(1) = Z_2$. A motion is legal if $m(t) \in LEGAL$ for $0 \leq t \leq 1$. A motion is in contact if $m(t) \in CONTACT$ for $0 \leq t \leq 1$.

The theorem concerning the motion of objects in contact can be restated as it applies to the restricted case of one moving object, and connected environment as follows:

Theorem 3.1 *If there is a legal motion between two configurations in the contact space, then there is a motion in contact [Hopcroft and Wilfong 84a].*

In the proof of this theorem the Meyer-Vietoris sequence [Massey 78] is used in order to prove that the number of path-connected components of CONTACT is equal to the number of path-connected components of LEGAL. In order to use this sequence, a certain subspace of the configuration space had to be contractible to a point. For this to be the case, a rotation of 2π can not be identified with no rotation at all. Furthermore, in order for that subspace to be path-connected, each parameter corresponding to an orientation can only vary within some closed and bounded interval. The resulting restricted configuration space for which the theorem is valid does not impose any problems in its use.

A second point to be noted with respect to that theorem is that it is not valid if only rotations are allowed. As an example of a situation for which the theorem
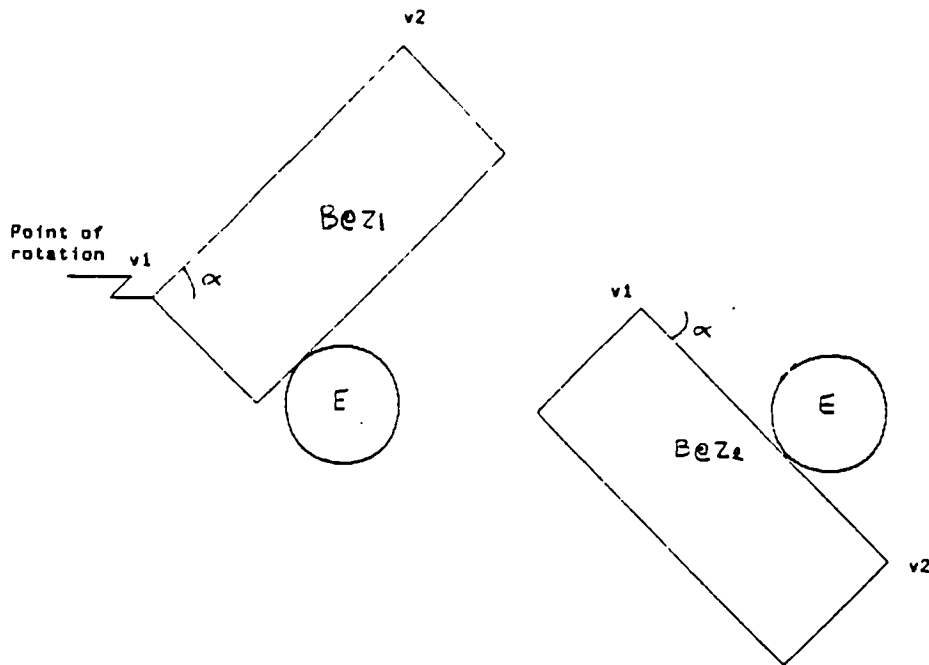
Object B is allowed to rotate about v_1 but not to translate, and object E is stationary. Although there is a legal motion from $\theta = \alpha$ to $\theta = 2\pi - \alpha$, there is not a motion which keeps the objects in contact [Hopcroft and Wilfong 84a].

Figure 3–7: Pure rotational motion

does not hold is shown in Figure 3–7. In this case, when only rotational motion is permitted, there is no motion in contact between the two configurations shown, although there is a legal motion. This problem arises because, if only rotations are allowed, the resulting configuration space is not contractible to a point.

3.2.5 Interactions among features of objects

The boundary of a polyhedral object is composed of three types of geometric entities: plane faces, edges, and vertices. These entities will be referred to as *features* . There are three *basic types of contact* between the features of a polyhedral object and a polyhedral obstacle:

1. a vertex of the object lies on a plane of the obstacle (Figure 3–8);

2. a vertex of the obstacle lies on a plane of the object (Figure 3–9);

3. an edge of the object intersects an edge of the obstacle (Figure 3–10).

It is easy to see that any other type of contact in the case of polyhedra can be expressed as a conjunction of contacts of the above form. For example, an edge on a plane contact will either be expressed as two vertices on a plane, if the whole edge lies on the face, or a vertex on a plane and an edge intersecting an edge, if only part of the edge lies on the face (Figure 3–11).

In Section 3.4 the basic types of contact and the ways of expressing any contact between polyhedral objects are formally defined. A contact between the moving object and its environment imposes a constraint on the configuration of the object. In Chapter 4 the concept of *spatial relationships* between features of objects will be introduced in order to define the constraints on the relative locations of the objects imposed by each specific type of contact. For the moment let us just assume that if the moving object at configuration Z is in contact with the environment, then Z satisfies some equation of the form $f(Z) = 0$, where f is a 'smooth' real-valued function on the configuration space, and its form depends on the type of contact. In Section 4.3.2 the form of these functions will be examined.

3.2.6   Surfaces of the Contact Space

If the object is in one of the basic types of contact with the environment then its configuration is constrained to lie on a 5-dimensional manifold in the configuration space. That is, each basic type of contact among the features of polyhedral objects defines a 5-dimensional manifold (surface)[2] in configuration space. It follows from [Lozano-Perez 83,Canny 84b] that these manifolds form the boundary of the $OVERLAP$ and thus the Contact Space.

---

[2] An *n-dimensional manifold* is a Hausdorff space such that each point has an open neighbourhood homeomorphic to the open *n*-dimensional disc [Massey 67].
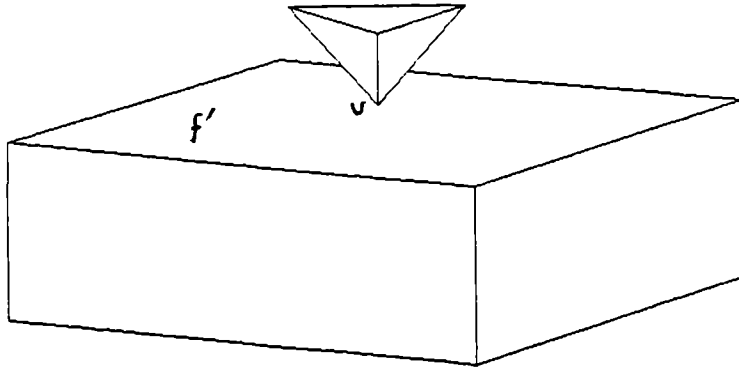
Vertex *v* of moving object lies on plane *f'* of obstacle.

Figure 3–8: Vertex-plane contact
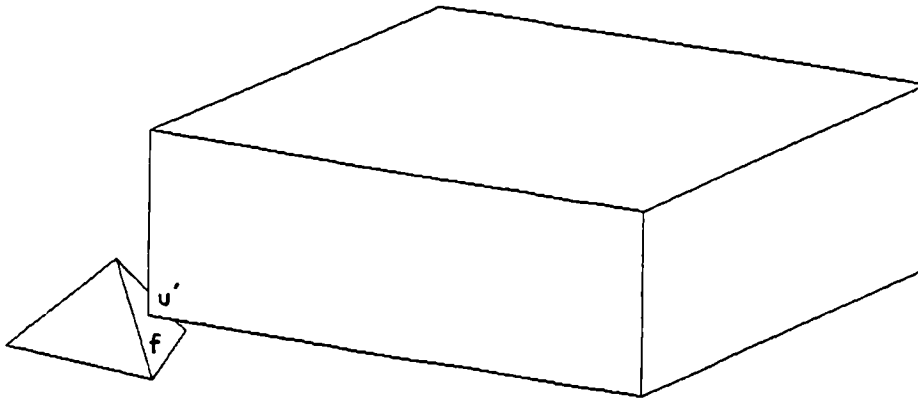
Vertex *v'* of obstacle lies on plane *f* of moving object.

Figure 3–9: Plane-vertex contact
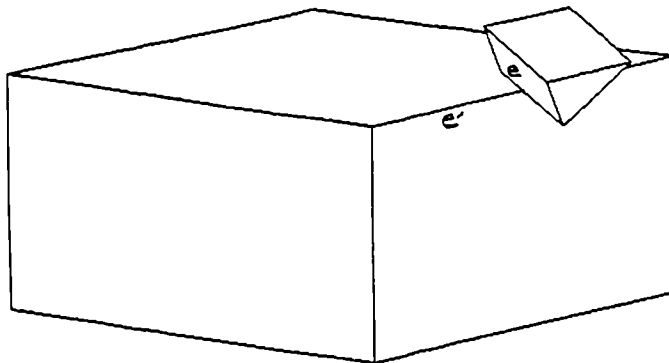
Edge *e* of moving object intersects edge *e'* of obstacle.

Figure 3–10: Edge-edge contact
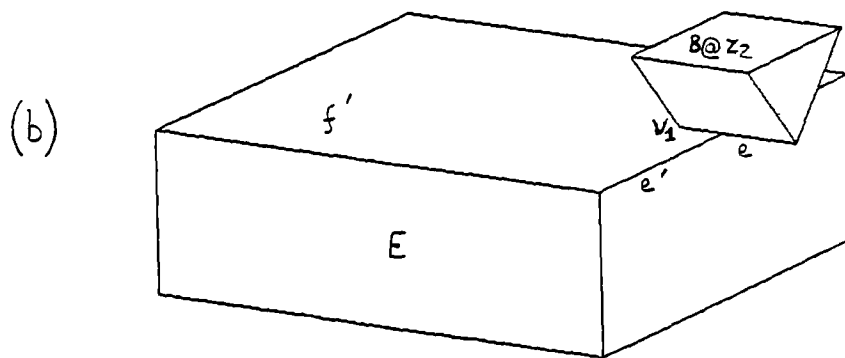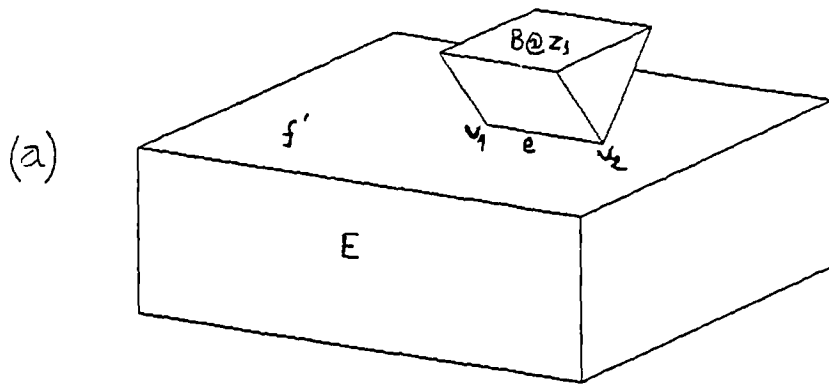
In (a) The edge e to plane f' contact can be expressed as "v_1 on plane f' and v_2 on plane f'''.

In (b) the contact can be expressed as "v_1 on plane f' and e intersects e'''.

Figure 3–11: Two cases of edge-plane contact

Since every type of contact can be described as a conjunction of the three basic types of contact, every configuration at which the object is in contact lies on the intersection of some 5-dimensional manifolds. The intersection of two or more 5-dimensional manifolds lies on manifolds with dimension $n = 0, \ldots, 5$. The dimension of the manifold is equal to the degrees of freedom of motion of the object, if its configuration is constrained to lie on the manifold, that is if its motion is constrained so that certain features are kept in contact.

The manifolds (surfaces) of the contact space and their intersections are examined at various points in this thesis. In Section 3.5.2 the intersections of the surfaces are considered. In Section 4.3.2 the equations of the surfaces are established by means of spatial relationships. Finally, in Chapter 5 an algorithm is presented for constructing the surfaces of the space.

3.3   Decompositions and Retractions of Topological Spaces

In Section 2.2.2 two general techniques for motion planning were identified: the decomposition approach and the retraction approach. In the first case, the space in which a path has to be found is decomposed into connected regions, where each region is represented by a vertex in a graph. If two regions are adjacent then there is an edge in the graph connecting the vertices which correspond to the regions. The path-planning problem is transformed, thus, into a graph searching problem.

Using the retraction approach, the problem of searching for a path in a space is reduced into the problem of searching for a path in a lower dimensional subspace. For this to be possible, there must be a specific relation between the path-connected components of the space and its subspace. Retractions exhibit this property.

In this section, the concepts of decompositions and retractions will be defined. Furthermore, a special case of decomposition, *cell complexes,* will be considered.

3.3.1   Cell Decompositions

A *cell decomposition* of a compact topological space S is a finite collection K of disjoint connected subsets c of S, called cells, whose union is S [Massey 78]. Two cells b, c are adjacent if

$$(cl(b) \cap c) \cup (b \cap cl(c)) \neq \emptyset.$$

The *adjacency graph* of K is the graph $G(K)$ with vertices corresponding to the cells and with edges connecting two vertices when the cells are adjacent. Then, a path between two points p_1, p_2 exists if and only if there is a path in $G(K)$ between c_1, c_2, where c_1 contains p_1 and c_2 contains p_2. Thus, the path planning problem is reduced to a graph searching problem.

A decomposition method for path planning requires: (a) a decomposition algorithm to find the cells; (b) an adjacency test to construct the adjacency graph; (c) a cell location algorithm to find the cell in which a point is contained; (d) a graph searching algorithm.

A special case of cell decomposition is a cell complex.

3.3.2   Cell Complexes

The definition of a cell complex is given below [Cooke and Finney 67]:

Definition 3.3.1 *A* finite *cell complex consists of a compact topological space* $|K|$ *and a sequence of subspaces called* skeletons *and denoted by* $|K_d|$, $d = -1, 0, 1, \ldots, n$, *which satisfy:*

1. $\emptyset = |K_{-1}|, |K_{-1}| \subset |K_0| \subset |K_1| \ldots \subset |K_n|, |K_n| = |K|$.

2. *Each* $|K_d|$ *is closed in* $|K|$.

3. $|K| = \bigcup_{0 \le d \le n} |K_d|$.

4. *For each* $d \ge 0$, *the components* $c_1, c_2 \ldots c_n$ *of* $|K_d| - |K_{d-1}|$ *are open cells in the relative topology of* $|K_d|$. *They are referred to as the* d-cells *of* $|K|$.

5. *For each d-cell, c of* $|K|$, *there exists a continuous map f from the closed unit ball B of dimension d to the closure of c,* $cl(c)$, *where f when restricted to the interior of B, is a homeomorphism onto c.*

A cell complex is shown in Figure 3–12.

Definition 3.3.2 *A cell complex is* regular *if there exists a homeomorphism from the closed unit ball onto the closure of c,* $cl(c)$.

The cell complex in Figure 3–12 is regular, while the cell complex in Figure 3–13 is irregular.
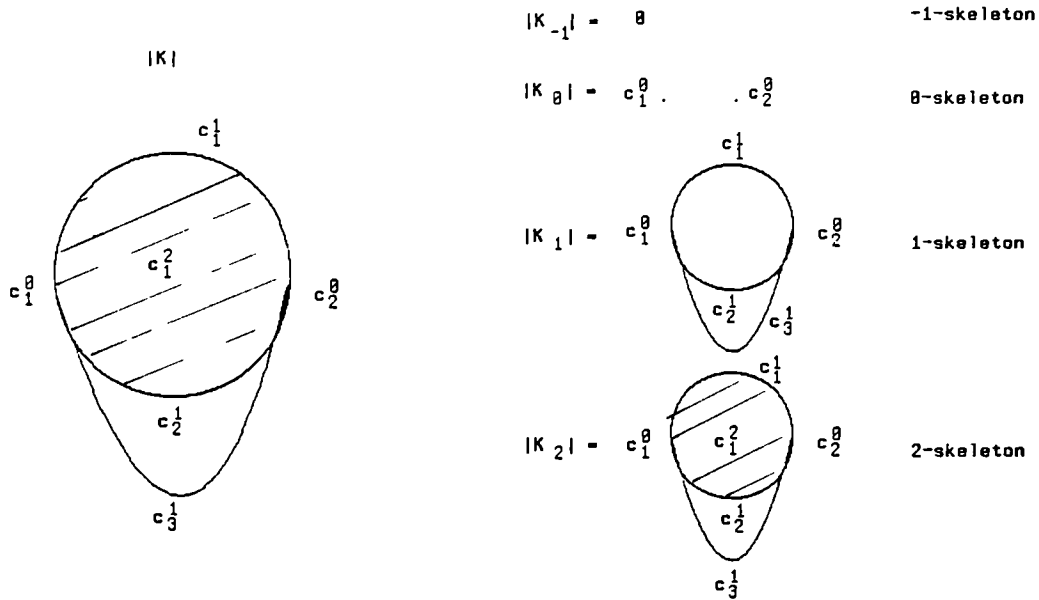
A regular cell complex consisting of one 2-cell $\left(c_1^2\right)$, three 1-cells $\left(c_1^1, c_2^1, c_3^1\right)$ and two 0-cells $\left(c_1^0, c_2^0\right)$.

$|K_d|$ denotes a d-skeleton, consisting of cells of dimension less than or equal to d ([Agoston 76]).

Figure 3–12: A regular cell complex
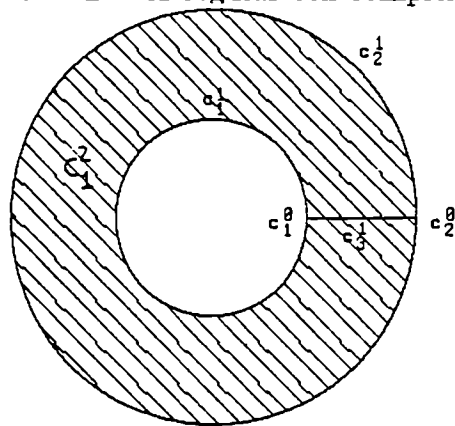
An irregular cell complex consisting of one 2-cell $\left(c_1^2\right)$, three 1-cells $\left(c_1^1, c_2^1, c_3^1\right)$ and two 0-cells $\left(c_1^0, c_2^0\right)$ (see [Cooke and Finney 67]).

Figure 3–13: An irregular cell complex

Definition 3.3.3 *The boundary of a cell, bdry(c) is defined to be:*

$$bdry(c) = cl(c) - c$$

The following properties of cell complexes either follow from the definition or can be proved (see [Cooke and Finney 67]):

1. $|K|$ is called the *underlying space* of the complex and it is the union of all cells: $|K| = \bigcup_i c_i = \bigcup_i cl(c_i)$.

2. The *d-skeleton* $|K_d|$ of $|K|$ is the union of cells of dimension less or equal to d: $|K_d| = \bigcup_{i,q} c_i^q$, for $q \le d$.

3. The *d-cells* are the components of $|K_d| - |K_{d-1}|$.

4. The cells c are *disjoint*

5. If c is a d-cell, then $cl(c) \subset |K_d|$.

6. The *boundary* of a d-cell c, lies in the union of cells of dimension lower than the dimension of the cell, $bdry(c) = cl(c) - c = cl(c) \cap |K_{d-1}|$.

7. The boundary of a d-cell is connected if $d \ge 2$.

8. If K is a cell decomposition of a compact topological space into disjoint d-dimensional cells and for each cell:

    (a) there exists a continuous mapping from the closed unit ball $\mathcal{B}$ to the closure of c which maps the interior of $\mathcal{B}$ homeomorphically onto c,

    (b) the boundary of a d-cell, c lies in the union of cells of dimension lower than that of c,

    then K is a cell complex.

9. If c is a cell of a *regular* cell complex then $cl(c)$ contains at least one vertex.

10. For a *regular* cell complex, if c_1 is a d-cell and c_2 is a $(d-2)$-cell in the boundary of c_1, then there are precisely two $(d-1)$-cells, c_3, c_3' on the boundary of c_1, such that c_2 is on the boundary of both c_3 and c_3'.

From the above properties it follows that if a space is decomposed into cells c which form a regular cell complex, then there is a path in the space between points p_1 in c_1 and p_n in c_n if and only if there is a chain $c_1 \rightarrow c_2 \rightarrow \ldots c_i \ldots \rightarrow c_n$, such that c_{i+1} belongs to the boundary of c_i or vice versa. Therefore, in order to find a path we need: (a) a decomposition algorithm to find the cells; (b) a test to find whether a d-cell belongs to the boundary of a $(d+1)$-cell; (c) a cell location algorithm; (d) a graph searching algorithm.

In comparison with a general cell decomposition, cell complexes have more structure, and the homology groups of the space can be found by examining the structure of the complex. A cell complex would have, normally, fewer cells than a simplicial complex, but more cells than a general decomposition where the cells need not be homeomorphic to open balls. Therefore, the properties of the complex have been introduced at the expense of adding more cells.

3.3.3 Retractions

Definition 3.3.4 *Let $A \subset X$. Then A is said to be a retract of X if there exists a continuous map $r : X \mapsto A$ such that the restriction of r to A is the identity. r is called a retraction [Massey 78].*

Using retractions, the problem of path-planning in a space can be reduced to the problem of path planning in a lower dimensional subspace, that is in a retract of the space:

Let $A \subset X$ and A be a retract of X. Then if there exists a path $p(t)$ in X between two points t_1 and t_2 in A, then there is path in A.

Yap [Yap ss] defines a *retraction-like* map and then proceeds to show that for cell-complexes $|K_{d-1}|$ is a retract of $|K_d|$. Then he shows that:

Theorem 3.2 *The vertices of a cell complex are edge-connected iff the underlying space of the cell complex is path-connected.*

Proof

See [Yap ss]. □

Using Voronoi diagrams for path-planning is an example of the retraction approach to motion planning. Intuitively, the points on the 1-dimensional skeleton of a Voronoi diagram represent configurations where the objects have *maximum* clearance. In this research, we are interested in the 1-dimensional skeleton of the Contact space, that is, the configurations where the objects have *minimum* clearance.

3.4   Decomposition of the Contact Space

In this section a decomposition of the contact space will be described, which is an extension to three dimensions and rotations of the decomposition in [Hopcroft and Wilfong 84b].

3.4.1   Clauses

A feature (an edge, face, or vertex) of the moving object is in contact with a feature of the environment at some configuration if their intersection is nonempty. The concept of a clause is introduced to describe the basic types of contact. In the next section, conjunction of clauses will be used to describe all possible types of feature interactions.

Definition 3.4.1 *A clause c is a pair (f_1, f_2), where either*

1. f_1 *is a vertex of moving object B, f_2 is a face of environment E or*

2. f_1 *is a face of moving object B, f_2 is a vertex of environment E or*

3. f_1 *is an edge of moving object B, f_2 is an edge of environment E*

By convention f_1 always belongs to the moving object.

Definition 3.4.2 *A configuration Z satisfies a clause if the features in the clause are in contact, that is*

$$f_1 @ Z \cap f_2 \neq \emptyset.$$

With each clause c there is an associated 'smooth'[3] real valued function on configuration space C:

---

[3]continuously differentiable

$$f_c : C \longmapsto \Re.$$

At the moment I will not be concerned with the form of the functions but only with their geometric significance. The form of the functions is discussed in Section 4.3.2. It has to be noted though that the form of the functions depends heavily on the representation of rotations. If rotations are represented by Euler angles, then the functions are not algebraic, since they include cos and sin. [Canny 84b,Canny 85] uses quaternion representation of rotations to derive algebraic functions.

If a configuration Z satisfies the clause c, then the associated function is zero-valued, $f_c(Z) = 0$, that is it lies within the kernel of the function $Ker(f_c)$,

$$Ker(f_c) = \{Z | f_c(Z) = 0\}.$$

However the function can be zero-valued and yet the clause may not be satisfied. In the case of vertex-face contact, that would mean that the vertex lies on the plane containing the face but not on the face itself. Similarly, in the edge-edge case, the lines would intersect but not the edges. This is illustrated in Figure 3–14.

The sign of the function at a configuration is of particular importance. In the case of vertex-face contact it indicates whether the vertex lies above or below the plane of the face. In Section 3.4.6 the sign of the functions will be used in the decomposition of the space.

The sign of a function can also be used to find out if the objects overlap [Canny 84a], [Lozano-Perez 83]: the half-space (or half-hyperspace more precisely) associated with a clause c is the set of all configurations where $f_c \leq 0$,

$$h_c = \{Z | f_c(Z) \leq 0\}$$

The union of intersections of a number of such half-spaces define the region $cl(OVERLAP)$, that is, the grown obstacles (see [Canny 84a]).
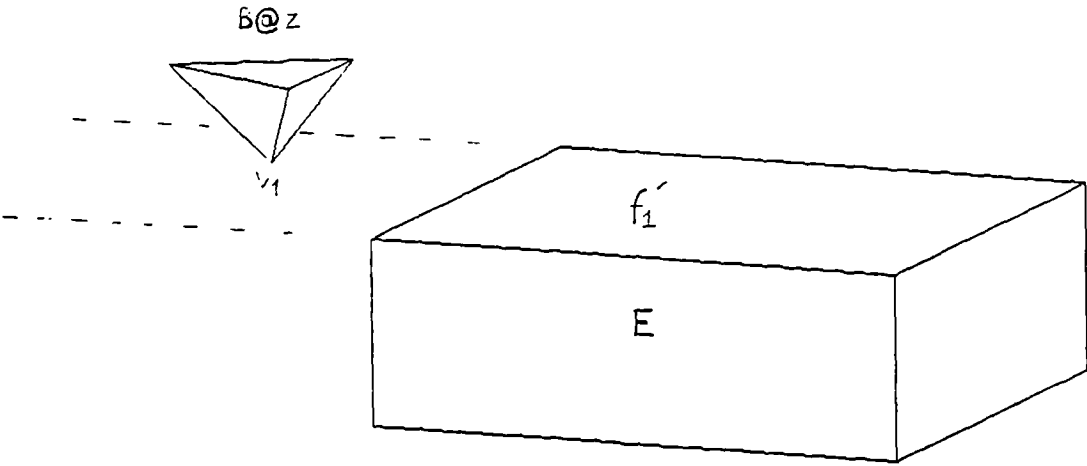
The vertex v_1 of B lies on the plane f_1' but not on the face f_1'. In this case $f_c(Z) = 0$, where $c = (v_1, f_1')$, but Z does not satisfy the clause c.

Figure 3–14: A configuration which does not satisfy a clause

3.4.2   Descriptors

In the previous section a clause was defined in terms of the three basic types of contact. All other types of contact between polyhedral objects can be expressed in terms of clauses.

Definition 3.4.3 *A descriptor D is a predicate that is a formal conjunction of clauses c_i,*

$$D = \bigwedge_i c_i.$$

Each c_i is said to be in D.

Definition 3.4.4 *A configuration satisfies a descriptor if it satisfies all the clauses of the descriptor. Let S_D be the set of all configurations which satisfy the descriptor D,*

$$S_D = \{Z | Z satisfies\ c, \forall\ c\ in\ D\}.$$

Since for each clause there is an associated function f_c, and for all configurations which satisfy the clause, $f_c(Z) = 0$, then with each descriptor there is an associated system of equations. If a configuration satisfies a descriptor then it satisfies the system of equations associated with the descriptor. The converse is not true.

Definition 3.4.5 *H_D is the set of configurations which satisfy the system of equations associated with the descriptor D,*

$$H_D = \{Z | f_c(Z) = 0, \forall\ c\ in\ D\}.$$

Note that $S_D \subseteq H_D$. The variables in the system of equations are the translational and the rotational degrees of freedom of motion. The dimension of the solution space of the system of equations is equal to to the number of degrees of freedom of the motion of the object.
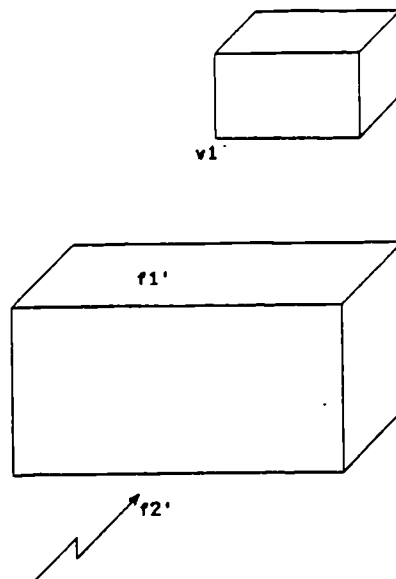
The descriptor $D = (v_1, f_1') \wedge (v_1, f_2')$ describes a contact situation which is unattainable.

Figure 3–15: An inconsistent descriptor

Definition 3.4.6 *The* dimension *of a descriptor is equal to dimension of the solution space of the system of equations of the descriptor.*

From the above we can conclude that a configuration satisfies a descriptor if

1. the configuration satisfies the system of equations associated with the descriptor

2. the features in each clause of the descriptor have non-empty intersection at the configuration

While (2) is a necessary and sufficient condition, (1) is only a necessary condition.

If the system of equations of a descriptor has no solutions, then the descriptor describes a contact situation which is unattainable. In this case the descriptor is called inconsistent. If the system of equations of a descriptors has at least one solution, then the descriptor is called consistent.

Definition 3.4.7 *A descriptor D is* consistent *if $H_D \neq \emptyset$.*
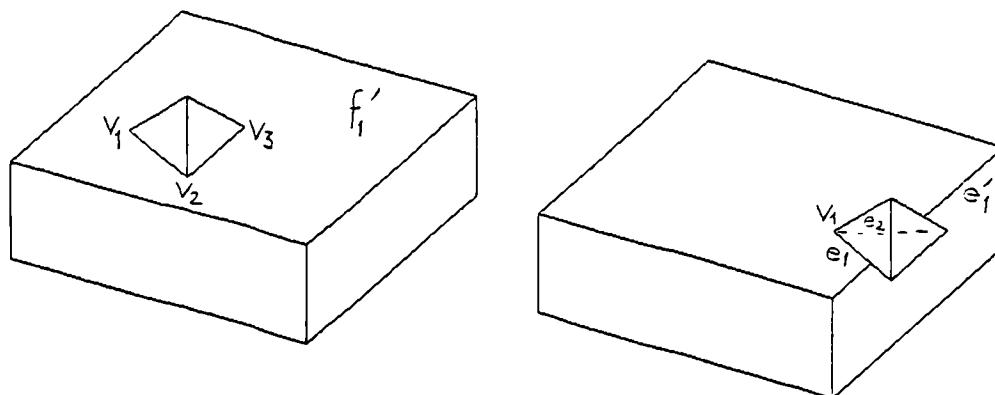
(a) $D_1 = (v_1, f_1') \wedge (v_2, f_1') \wedge (v_3, f_1')$ (b) $D_2 = (v_1, f_1') \wedge (e_1, e_1') \wedge (e_2, e_1')$

The systems of equations of the two descriptors have the same sets of solutions: the descriptors are kinematically equivalent.

Figure 3–16: Equivalent descriptors

Figure 3–15 shows a case of an inconsistent descriptor. Clearly a vertex can not be in contact with two parallel planes simultaneously. The system of equations of the descriptor $D = (v_1, f_1') \wedge (v_1, f_2')$ has no solutions. Whether or not the system of equations has any solution does not depend at all on the physical extent of the objects. The system of equations can be seen as a definition of some kinematic mechanism. Distinguishing between a configuration which satisfies a descriptor and a configuration which satisfies the system of equations implies distinguishing between a configuration which satisfies a descriptor in terms of physical contact and a configuration which satisfies a descriptor in the 'kinematic' sense. Therefore, a system which handles transformations can be used to deal with the equations, while a system with knowledge about space occupancy can be used to deal with the task of physical contact.

It can also be observed that two different descriptors may describe two contact situations which are kinematically equivalent. In Figure 3–16, $D_1 = (v_1, f_1') \wedge (v_2, f_1') \wedge (v_3, f_1')$ and $D_2 = (v_1, f_1') \wedge (e_1, e_1') \wedge (e_2, e_1')$. Both descriptors describe the same planar mechanism: the plane of f_1 must be against the plane of f_1'.
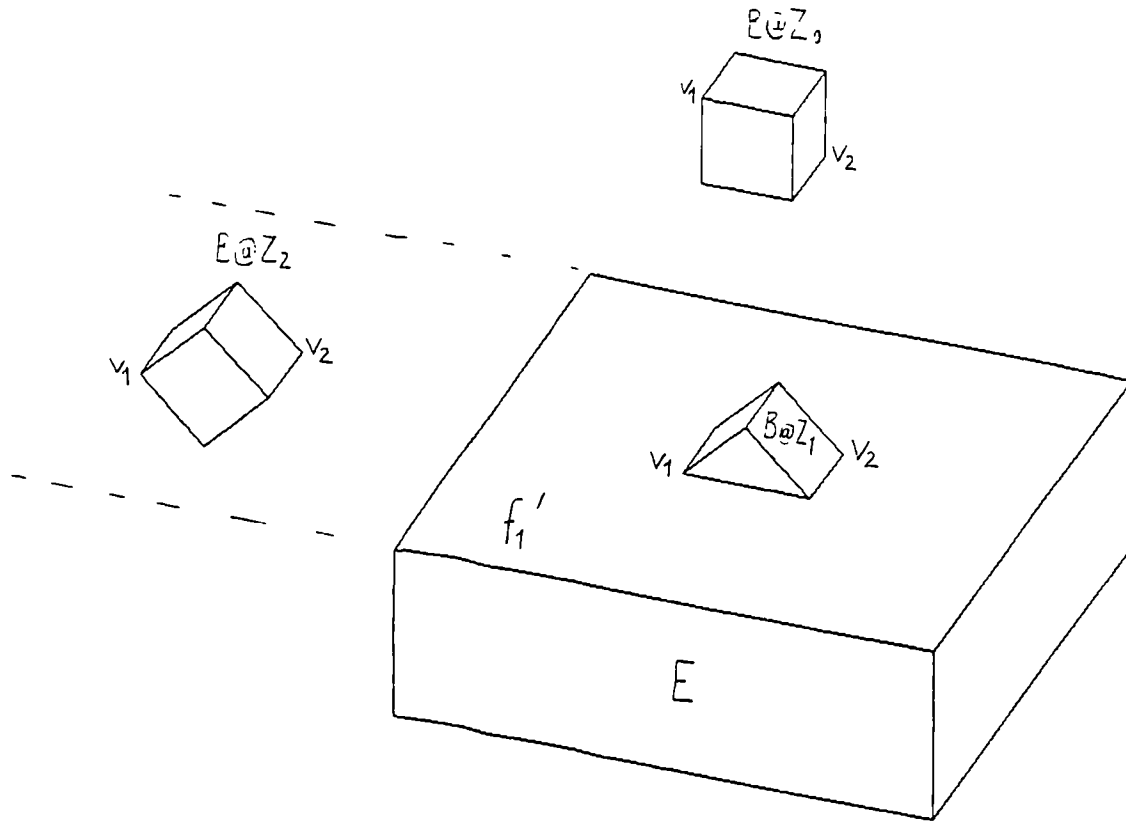
Z_1 satisfies the descriptor $D = (v_1, f'_1) \wedge (v_2, f'_1)$ but is illegal, since $B@Z1$ overlaps E. Z_2 satisfies the system of equations of D, since v_1 and v_2 are on the plane of f'_1, but Z_2 does not satisfy D. The descriptor $D = (v_1, f'_1) \wedge (v_2, f'_1)$ is illegal: it is not satisfied by any legal configurations.

Figure 3-17: An illegal descriptor

In this case we say that D_1 and D_2 are equivalent, meaning equivalent in the kinematic sense.

Definition 3.4.8 *The descriptors D_1 and D_2 are* equivalent, $D_1 \sim D_2$, *if* $H_{D_1} = H_{D_2}$.

Let us now consider physical interference between objects. In Figure 3-17 configuration Z_1 satisfies the descriptor $D = (v_1, f'_1) \wedge (v_2, f'_1)$ but it is physically impossible. It can also be seen that in this case there are no legal configurations that satisfy the descriptor, although there are legal configurations which satisfy the system of equations (e.g. Z_2).

Definition 3.4.9 *A descriptor is* legal *if it is satisfied by at last one legal configuration.*
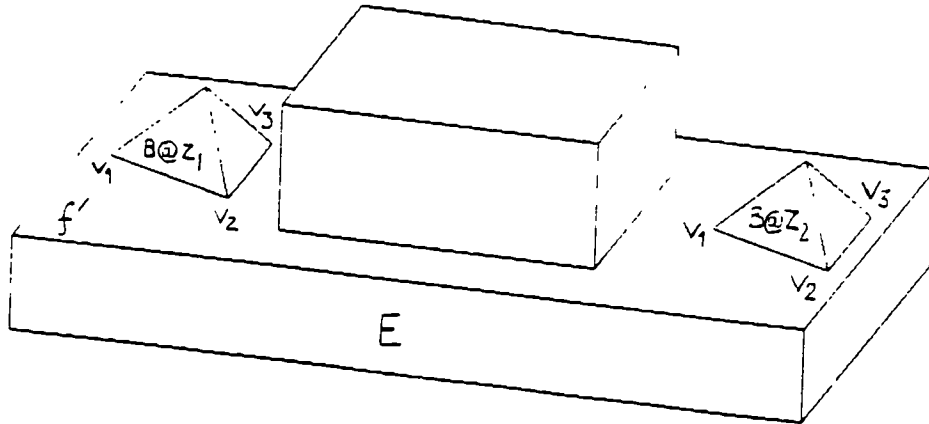
Both configurations Z_1 and Z_2 satisfy exactly the descriptor $D = (v_1, f') \wedge (v_2, f') \wedge (v3, f')$ but they belong to different path-connected components of E_D.

Figure 3–18: Path-connected components of E_D

As is shown in Figure 3-11 (page 51), a configuration can satisfy more than one descriptor. Configuration Z_1 satisfies $D = (v_1, f_1')$, $D_2 = (v_2, f_1')$ and $D_3 = (v_1, f_1') \wedge (v_2, f_1')$.

The last two observations are captured by the following definition:

Definition 3.4.10 *A configuration Z exactly satisfies a descriptor D if Z is legal and Z does not satisfy any clauses not included in the descriptor. Let E_D be the set of legal configurations which satisfy D exactly,*

$$E_D = \{Z | Z \in S_D \cap LEGAL \bigwedge \forall c = (f_1, f_2) \notin D : f_1@Z \cap f_2 = \emptyset \}.$$

Finally, it can be noted that the set E_D is not necessarily path-connected. In Figure 3-18, both Z_1 and Z_2 satisfy exactly the descriptor $D = (v_1, f') \wedge (v_2, f') \wedge (v3, f')$ and both are legal. Therefore $Z_1, Z_2 \in E_D$. There is not a path however in E_D that connects Z_1 and Z_2. Since we are interested in path-connected components of CONTACT, E_D should be partitioned into its path-connected components.

Definition 3.4.11 $P_D(i)$ *is defined as a path connected component of* E_D.

Let's summarise what has been achieved so far:

1. Every type of contact has been described in terms of the three basic types of contact by means of clauses and descriptors.

2. The set H_D of configurations which satisfy the system of equations of the descriptor has been considered.

3. Various subsets of H_D have been classified: $P_D(i) \subseteq E_D \subseteq S_D \subseteq H_D$

   - S_D : set of configurations which satisfy the descriptor.

   - E_D : set of configurations which exactly satisfy the descriptor and are legal.

   - $P_D(i)$: set of configurations which exactly satisfy the descriptor, are legal and path connected.

The definitions are illustrated collectively in the example shown in Figure 3–19.

In the next section it will be shown that the path-connected components of all possible consistent descriptors form a decomposition of the Contact Space.
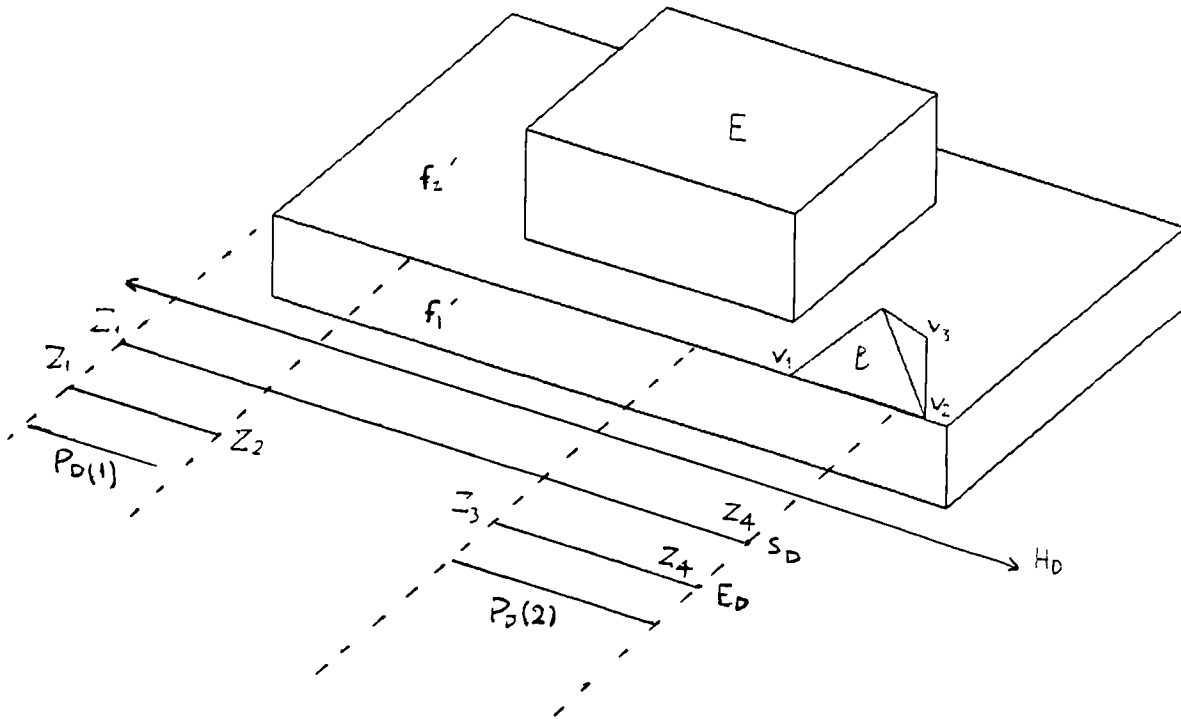
Let the origin of B be on v_1, and D be the descriptor specifying that v_1 and v_2 lie on both faces f_1' and f_2', and v_3 lies on f_2'. H_D is an infinite line in configuration space. S_D is a closed interval in H_D and its end points are defined by configurations Z_1 and Z_4. At Z_1 the vertex v_1 reaches the left end of the face f_1', while at Z_4 v_2 reaches the right end of f_1'. E_D consists only of the legal configurations of S_D which satisfy D exactly. Therefore, E_D does not include Z_1, Z_2, Z_3 and Z_4. Finally, $P_D(1)$ and $P_D(2)$ are the two path-connected components of E_D.

Figure 3–19: The subsets of H_D

3.4.3 Decomposition of the Contact Space Using Descriptors

Consider the set of all possible clauses for the two polyhedral objects B and E,

$$
\begin{aligned}
CLAUSES \;=\; & \{vert(B) \;\times\; faces(E)\} \\
& \cup \{faces(B) \;\times\; vert(E)\} \\
& \cup \{edges(B) \;\times\; edges(E)\}
\end{aligned}
$$

A descriptor is a conjunction of the elements of a non-empty subset of CLAUSES. Let DESCRIPTORS be the set of all possible consistent descriptors. It will be shown that the collection of all the sets $P_{D_a}(i)$, where D_a is a consistent descriptor, and $P_{D_a}(i)$ is the ith path-connected component of the set of legal configurations which satisfy D_a exactly, constitutes a decomposition of the contact space.

Lemma 3.4.1 *For every configuration Z in CONTACT there is exactly one descriptor D, such that Z satisfies D exactly.*

Proof

Since any contact between polyhedral objects can be expressed in terms of the three basic types of contact, any configuration in CONTACT satisfies some conjunction of clauses and, hence, satisfies at least one descriptor *exactly*. By the definition of E_D (page 66), a configuration cannot satisfy two or more distinct descriptors exactly. Therefore there is exactly one descriptor which it satisfies exactly. $\Box$

Lemma 3.4.2 $CONTACT = \bigcup_{i,D} P_D(i)$, $D \in DESCRIPTORS$.

Proof

Let $Z \in P_D(i)$. Then Z satisfies descriptor D. Let $c = (f_1, f_2)$ be a clause of D. Now $f_1@Z \cap f_2 \neq \emptyset \Rightarrow B@Z \cap E \neq \emptyset$. Also Z is legal and hence

$int(B@Z) \cap int(E) = \emptyset$. By the definition of CONTACT (page 45) we have that $Z \in CONTACT$.

Let $Z \in CONTACT$. Then Z is legal and by Lemma 3.4.1 there is a descriptor that Z exactly satisfies. Therefore, $Z \in E_D$ and hence Z belongs to some path-connected component of E_D, i.e. $Z \in P_D(i)$. $\square$

Lemma 3.4.3 $P_{D_a}(i) \cap P_{D_b}(j) = \emptyset$,

Proof

Let $D_a = D_b = D$, that is $P_D(i)$ and $P_D(j)$ are path connected components of E_D and hence disjoint.

Let $D_a \neq D_b$ and let $Z \in P_{D_a}(i) \cap P_{D_b}(j)$. Then Z would satisfy exactly both D_i and D_j which contradicts Lemma 3.4.1. Therefore, $P_{D_a}(i)$ and $P_{D_b}(j)$ are disjoint. $\square$

Theorem 3.3 *The collection of the sets $P_{D_a}(i)$, where $P_{D_a}(i)$ is the ith path-connected component of a set of legal configurations which satisfy a consistent descriptor D_a exactly, constitutes a cell decomposition of the Contact Space.*

Proof

The sets $P_{D_a}(i)$ are path-connected (by definition), disjoint (Lemma 3.4.3) subsets of the Contact Space whose union is the Contact Space (Lemma 3.4.2). Therefore, they constitute a cell decomposition of the contact space (see definition page 53). $\square$

3.4.4 Faces of the Contact Space and their Intersections

Having defined a decomposition of the contact space into the cells $P_{D_a}(i)$, I shall now proceed to examine the properties of this decomposition. Working towards this objective, the notion of faces and their boundaries will be introduced. A face of the contact space is defined as the closure of a cell of the decomposition:

Definition 3.4.12 *A face F_D of the Contact Space is defined to be the closure of a path-connected component P_D in H_D,*

$$F_D(i) = cl(P_D(i)).$$

The dimension of a face F_D is equal to the dimension of the descriptor D. 0-dimensional faces are called vertices and 1-dimensional faces are called edges. Let $|K_d|$ be the collection of faces of the Contact Space with dimension less or equal to d. $|K_d|$ is called the **d-skeleton** *of the Contact Space.*

In order to decide how the various faces are related, the concept of the boundary of a face will be introduced.

Definition 3.4.13 *The boundary of face F_D is defined by*

$$bdry(F_D(i)) = F_D(i) - P_D(i).$$

Intuitively, the boundary of a face corresponds to configurations which lie on the face but at which the motion of the object is further restricted by one or more additional contacts, that is configurations which don't satisfy the descriptor exactly. In Figure 3-20, for example, $D_1 = (v_1, f_1')$ and F_{D_1} is the unique face of the descriptor and is of dimension 5. While Z_1 satisfies the descriptor D_1 exactly, Z_2 and Z_3 don't. Z_2 satisfies the descriptors D_1, $D_2 = (v_2, f_1')$ and $D_3 = (v_1, f_1') \wedge (v_2, f_1')$. Z_3 satisfies the descriptors D_1, $D_4 = (v_1, f_2')$ and $D_5 = (v_1, f_1') \wedge (v_1, f_2')$. In this case Z_2 and Z_3 belong to the boundary of the face F_{D_1}. It can also be observed that configurations Z_2 and Z_3 lie in 4-dimensional
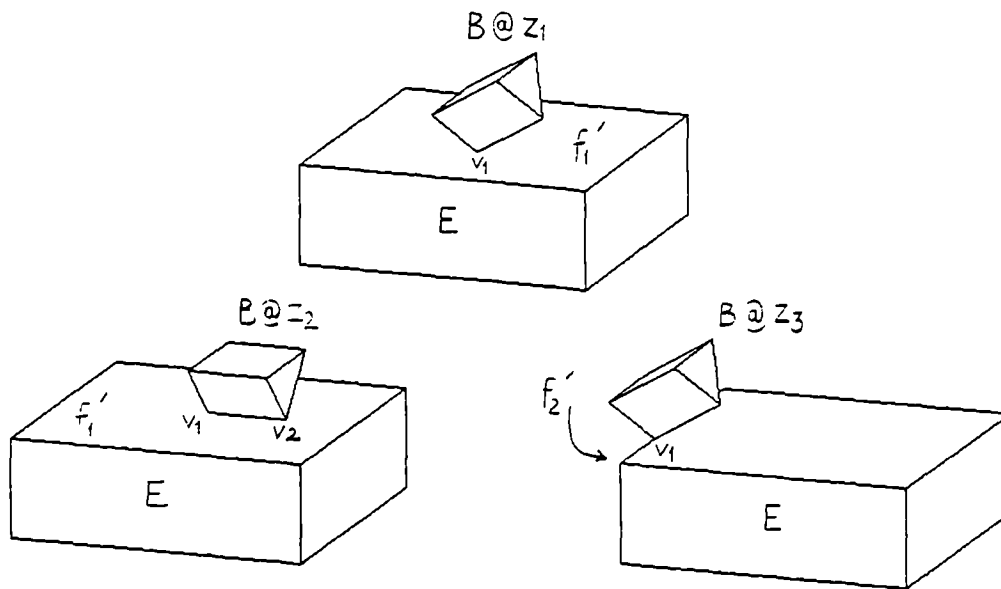
(a) Z_1 exactly satisfies the descriptor $D_1 = (v_1, f_1')$

(b) Z_2 satisfies the descriptors $D_1 = (v_1, f_1')$, $D_2 = (v_2, f_1')$ and $D_3 = (v_1, f_1') \wedge (v_2, f_1')$

(c) Z_3 satisfies the descriptors $D_1 = (v_1, f_1')$, $D_4 = (v_1, f_2')$ and $D_5 = (v_1, f_1') \wedge (v_1, f_2')$

Z_2 and Z_3 belong to the boundary of F_{D_1}

Figure 3–20: Boundary of a face

faces (Z_2 has 2 rotational and 2 translational d.o.f., Z_3 has 3 rotational and 1 translational d.o.f.). In addition, Z_2 lies on the intersection of F_{D_1} and F_{D_2} while Z_3 lies on the intersection of F_{D_1} and F_{D_4}. In this section, we will try to see if the observations that have been made with reference to Figure 3–20 are valid in general. In particular, the following questions will be examined:

Q1: Do all faces, apart from 0-dimensional faces, have non-empty boundaries?

Q2: Does the boundary of a d-dimensional face $(d > 0)$ consist of faces of dimension less than n?

Q3: Does the intersection of two faces consist of faces of dimension less than the minimum of the dimensions of the two?

Q4: Do all d-dimensional faces lie on the boundary of some $(d + 1)$-dimensional face $(0 \le d \le 4)$

Q5: For every face of dimension greater than one, is there a path on the boundary of the face between any two configurations on its boundary?

Q6: Are the vertices of the space edge-connected?

Q7: Finally, is the decomposition a cell complex?

By answering these questions, we would be able to examine some of the properties of the decomposition. In particular, the property we are more interested to establish is the edge-connectedness property.

If the answer to questions Q1–Q2 is yes, then the contact space can be regarded as a polytope. The structure of polytopes is well studied, and therefore, it will be useful if the Contact Space could be regarded as a polytope. Polytopes in general, however, do not exhibit the edge-connectedness property. If the decomposition is a regular cell complex then the answer to all questions Q1–Q7 would be yes.

Hopcroft and Wilfong [Hopcroft and Wilfong 84b] have examined the structure of the Contact Space in the case of 2-dimensional polyhedra and pure translational motion. Defining a decomposition of the space similar to the one presented in the previous sections, they proved that, even in the case of more than one moving object, the vertices of the space are edge-connected. As will be shown by some counter examples, the results do not generalise to the case of 3-dimensional objects, even if only translations are allowed.

The problems of the proposed decomposition will be illustrated by means of some examples. In a later section, the same examples will be used to to propose a slightly different method for decomposing the contact space which overcomes most of the problems encountered.

Problem 1: An edge with no vertices

Consider the case shown in Figure 3–21. Object B is free to rotate about the axis defined by vertices v_1 and v_2. Since there is one degree of freedom of motion, the face of the contact space which corresponds to this contact situation is 1-dimensional, i.e. an edge. It can be noticed that the motion of B can not be further constrained by additional contacts. In other words, there isn't a configuration which satisfies D and also satisfies a clause not in D. More formally, all configurations which satisfy the system of equations of D, satisfy D exactly and are path-connected, i.e.

$$P_D = E_D = S_D = H_D.$$

This implies that $P_D = H_D$. H_D is both open and closed in H_D, hence P_D is also both open and closed, and, therefore, $bdry(P_D) = \emptyset$.

It can be argued that the situation of faces with empty boundaries arises only in the case of 1-dimensional faces which correspond to one rotational d.o.f. or 2-dimensional faces which correspond to two rotational d.o.f. If there are translational d.o.f. then, eventually, some new contact constraint would be attained, either because the object would 'hit' something, or because the boundary
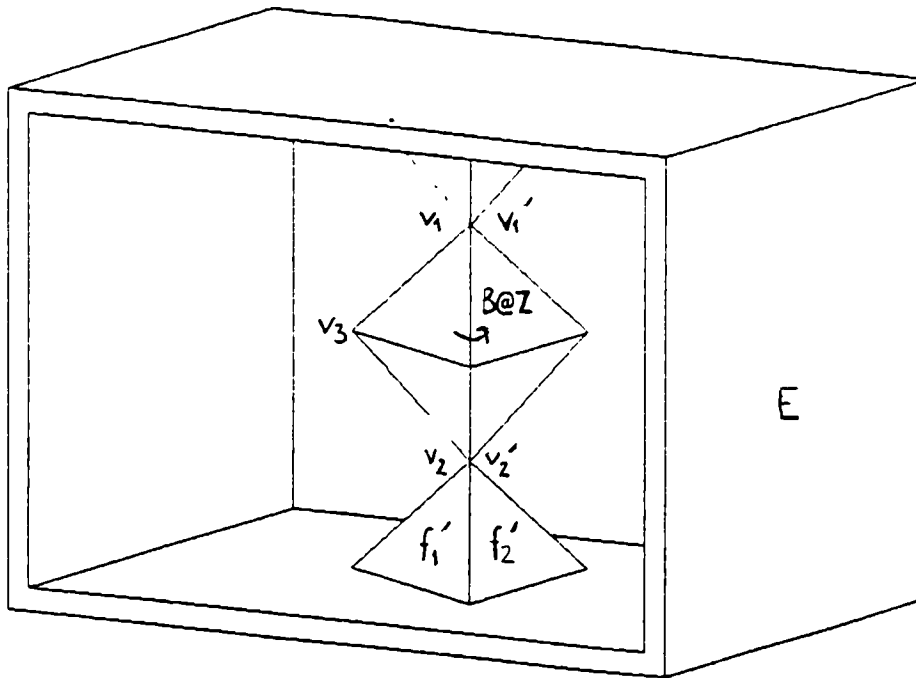
The descriptor D specifies that v_1 coincides with v_1' and v_2 with v_2'. Therefore, B is free to rotate about the axis $v_1 v_2$. Z belongs to a 1-dimensional face F_D with empty boundary

Figure 3–21: An edge of the contact space with empty boundary

of some feature in some clause would be reached. Similarly, if there are three rotational d.o.f., clearly the object can not rotate without 'hitting' some obstacle (Figure 3-22).

From this example, we can conclude that the answer to the question (Q1) is no, that is, not all faces have non-empty boundaries. As a result, there are faces which do not contain any vertices. This is a violation of a property of regular cell complexes (see item 9, page 56). Therefore, we can conclude that the decomposition is not a regular cell complex.
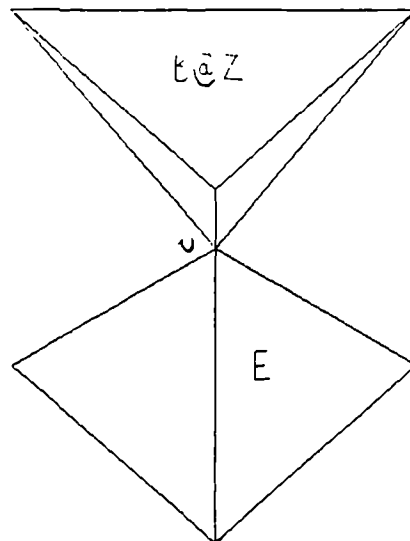
B can rotate about vertex *v*. *Z* is a configuration in a 3-dimensional face F_D. At some configuration in the face, *B* comes into contact with the environment. Therefore, the boundary of F_D is not empty.

Figure 3–22: A vertex to vertex contact

Problem 2: Faces with 'holes'

Consider the situation shown in Figure 3–23. A 3-dimensional object *B* is allowed only translational motion. Figure 3–23b shows the resulting contact space, which is the boundary of the *OVERLAP* region. Configurations Z_1 and Z_2 are vertices of the contact space. They lie on the boundary of a 2-dimensional face F_D, where *D* is a descriptor specifying that the vertices of face *f* of *B* lie on face *f'* of *E*. As can be seen from the figure, configurations Z_1 and Z_2 cannot be connected by a path in the boundary on the face. They can, however be connected by a path along edges of the contact space.

A similar situation is shown in Figure 3–24. In this case though, not only vertices Z_1 and Z_2 are not connected by a path in the boundary of F_D, but they are not edge-connected at all.

From the above two examples (Figures 3–23 and 3–24) we can conclude that the answer to question (Q5) is no, that is to say, there isn't always a path on the boundary of a *d*-dimensional face between two configurations on its boundary
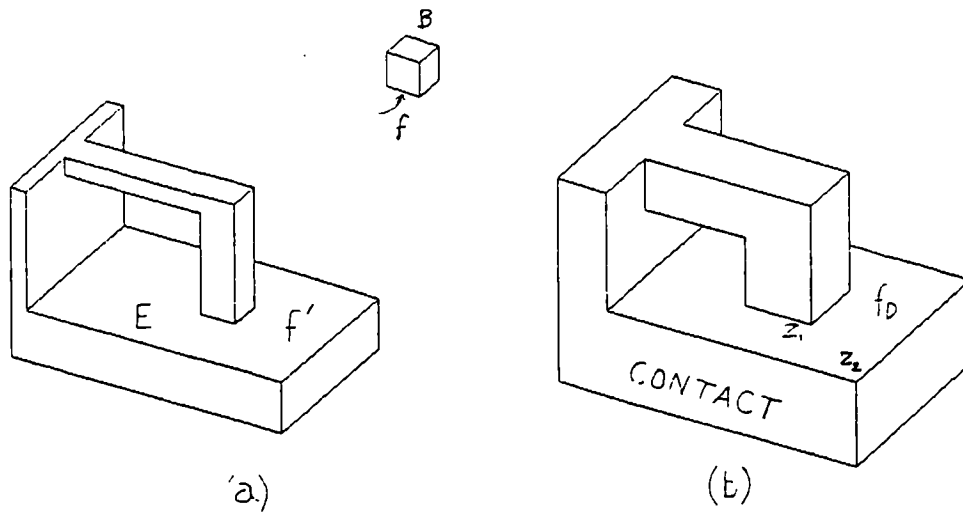
(a) shows the stationary object E and the moving object B; (b) shows the resulting contact space when only translations are allowed, assuming that the origin of B is its center point. Configurations Z_1 and Z_2 are vertices of the contact space in the boundary of the 2-dimensional face F_D. Although Z_1 and Z_2 are edge connected, they are not edge connected in the boundary of the face F_D.

Figure 3–23: Faces of the contact space with holes: Case 1
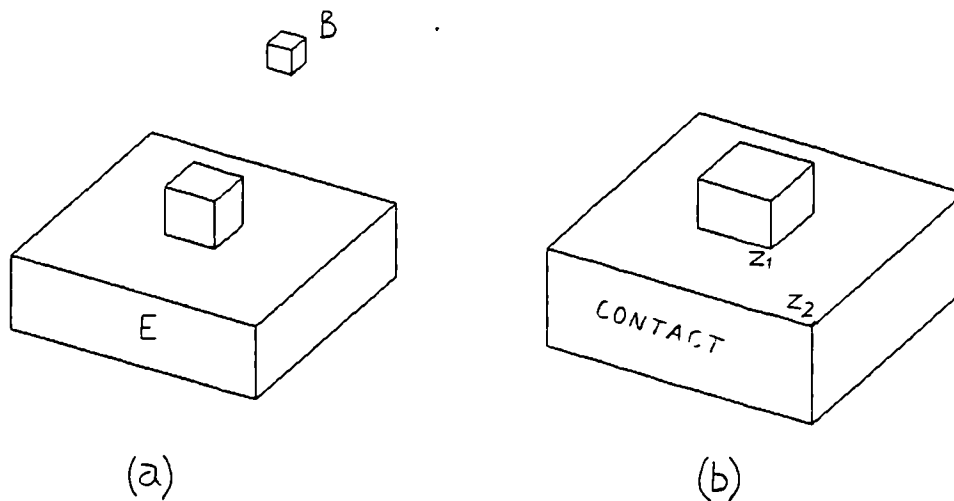
(a) shows the objects and (b) the resulting contact space, if only translations are permitted. Assume that the origin of B is its center point. Z_1 and Z_2 are vertices of the contact space which are not edge-connected.

Figure 3–24: Faces of the contact space with holes: Case 2

$(d \geq 2)$. In other words, the boundary of a d-cell $(d \geq 2)$ of the decomposition is not necessarily connected. This, again, is a violation of a property of cell complexes (see item 7, page 56). We can conclude, therefore, that the decomposition is not a cell complex.

The problem indicated by the above two examples arise because the cells of the decomposition have 'holes': although a cell $P_D(i)$ has been defined to be path-connected, its closure, $F_D(i)$, is not necessarily simply-connected. It can be seen, therefore, that the definition of a cell complex is violated (see item 5 of Definition 3.3.1, page 54).

Furthermore, from the second example (Figure 3–24) it can be concluded that in the case of 3-dimensional objects, even if only translations are allowed, the vertices of the contact space are not edge-connected. Therefore, the answer to question (Q6) is also no.

Problem 3: A vertex of a face which does not lie on an edge of the face

In the case shown in Figure 3–25 vertex v_2 of B is restricted to lie on f_1' and v_1 must coincide with v'. In this case, the object B has two rotational d.o.f.. The locus of the vertex v_3 can be seen to be the surface of a sphere. Configuration Z_0 lies thus on a 2-dimensional face F_D. Now, what happens if we constrain the vertex v_3 to be on a plane face f_3' of E? There are two cases. In Figure 3–25b, the distance from v_1 to f_3' is less than the length of the edge v_1v_3, that is to say, the plane of f_3' intersects the sphere. The locus of v_3 becomes then the circle where the plane intersects the surface of the sphere, and B has one rotational d.o.f. about the axis shown in the figure. Configuration Z_1 lies thus on a 1-dimensional face of the contact space, which belongs to the boundary of F_D. In Figure 3–25c, the plane of f_3' is tangent to the sphere and the locus of v_3 becomes the point where the plane is tangent to the sphere. Configuration Z_2 is thus a vertex which lies on the boundary of F_D. What is of interest in this case is the fact that if the object is at configuration Z_2 (vertex configuration) and the
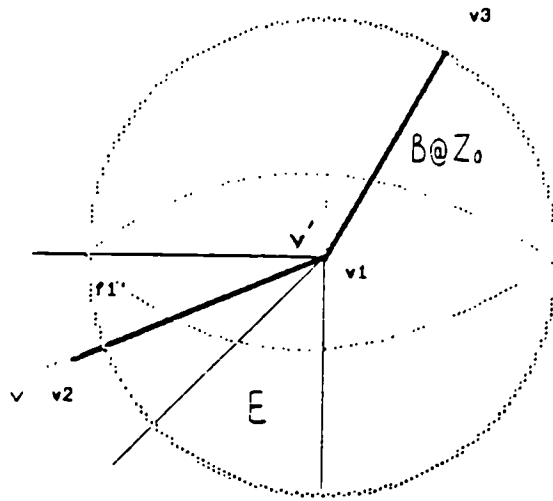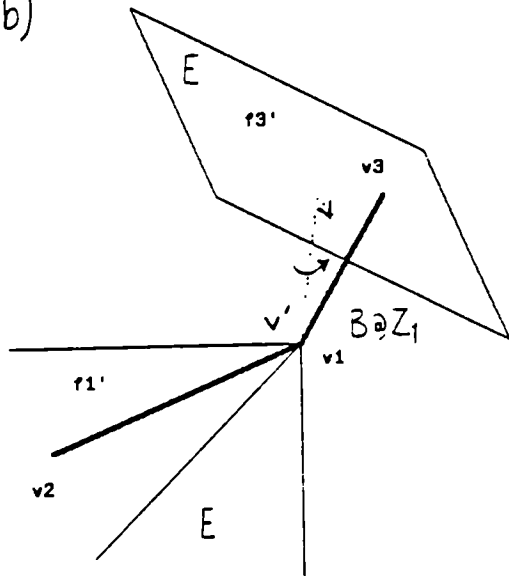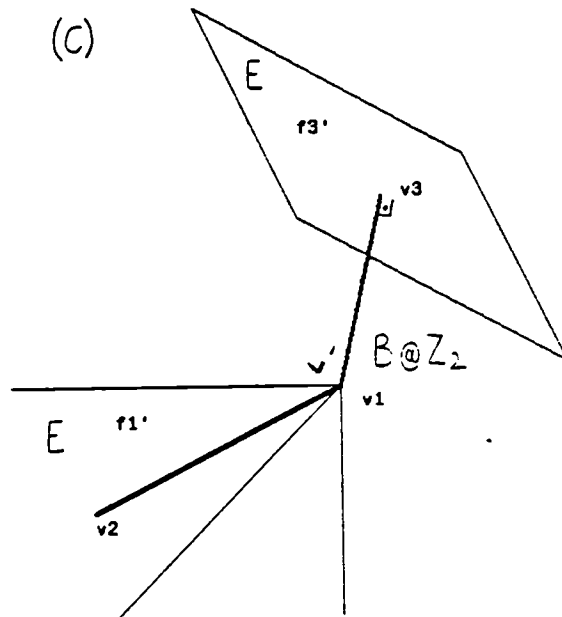
For illustration purposes, only the vertices v_1, v_2 and v_3 of B are shown. v_1 coincides with v' and v_2 is on f_1'. (a) B has two rotational degrees of freedom. The locus of v_3 is the surface of the sphere; Z_0 lies on a 2-dimensional face F_D. (b)v_3 is constrained to lie on a plane f_3' which intersects the sphere. Z_1 lies on a 1-dimensional face on the boundary of F_D; (c) v_3 is constrained to lie on a plane which is tangent to the sphere. Z_2 is a vertex on the boundary of F_D. Z_2 is not edge-connected in the face F_D but it is edge connected in the contact space.

Figure 3–25: A vertex on a face not lying on an edge on the face
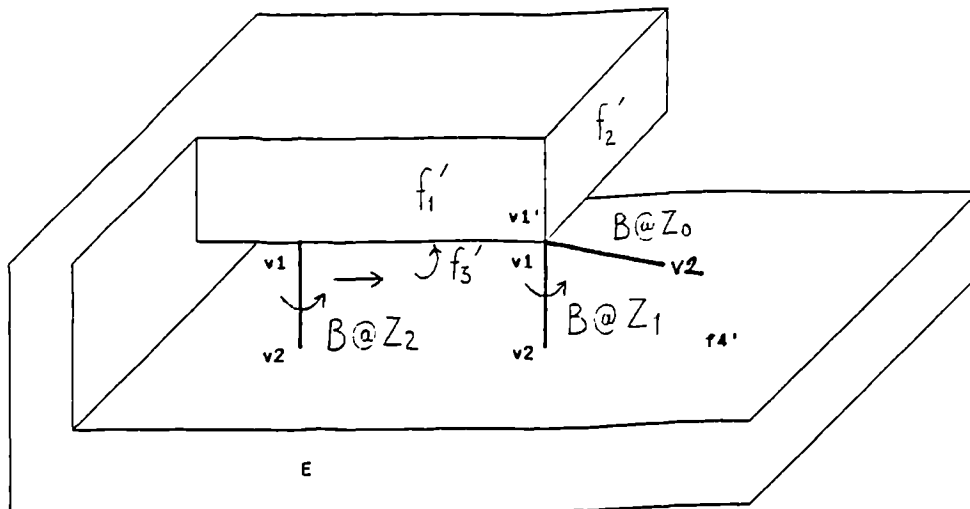
Only vertices v_1 and v_2 of object B are shown. Z_0 lies on a 3-dimensional face F_{D_1}, where D_1 specifies that v_1 coincides with v_1'. Z_1 lies on a 1-dimensional face F_{D_2} in the boundary of F_{D_1}. Z_2 lies on a 2-dimensional face F_{D_3}, where D_3 specifies that v_1 is on the intersection of f_1' and f_3' and v_2 is on f_4'. F_{D_2} is not connected to any edges in the boundary of F_{D_1} but it is connected to edges in the boundary of F_{D_3}.

Figure 3–26: An edge on a 3-dimensional face

constraint (v_3, f_3) is relaxed, then the object is at a configuration belonging to a 2-dimensional face (F_D) and not to an edge. That is to say, there are no edges in the boundary of the face F_D connecting vertex Z_2 with other vertices.

We can conclude, therefore, that the case shown in Figure 3–25c is one more manifestation of the fact that, for the proposed decomposition, the boundary of a cell is not necessarily connected. However, it has to be noticed that, similarly to the case shown in Figure 3–23, the vertex Z_2, although it is not edge-connected on the boundary of F_D, it is edge-connected in the contact space: if B is at configuration Z_2 and the constraint (v_1, f_1) is relaxed, then B could rotate about the line $v_1 v_2$. Z_2 lies thus on an edge of the contact space.

A similar but higher dimensional situation is shown in Figure 3–26. Let the initial situation be v_1 against v_1' (D_1). There are 3 rotational degrees of freedom and, therefore, the configuration Z_0 lies on a 3-dimensional face, F_{D_1}. If the constraint (v_2, f_4') is introduced, then there is only one rotational d.o.f. Configuration Z_1 lies thus on a 1-dimensional face. This is an edge on the boundary of the 3-dimensional face F_{D_1} which is not connected to other edges on the boundary of the face. Again, this example illustrates that the boundary of a face is not necessarily connected. As in the previous example, there are, however, 2-dimensional faces which have this edge on their boundary. For example, configuration Z_2 lies on such a 2-dimensional face. Therefore, although the edge is not connected to any edges in the boundary of F_{D_1}, it is connected to edges through 2-dimensional faces.

Let us summarise the implications of the last two examples (Figures 3–25 — 3–27). It has been indicated that there are cases where the addition of a single constraint (clause) eliminates two degrees of freedom and this results in vertices that are not connected by edges on a 2-dimensional face, edges that are not connected by 2-dimensional faces on a 3-dimensional face etc. This implies that the boundary of a face is not necessarily connected and, as a result, the decomposition is not a cell complex. The answers to questions (Q5) and (Q7) in page 73 are, therefore, no.

Problem 4: An edge which doesn't lie on the boundary of a 2-dimensional face

Consider the case shown in Figure 3–27. Configuration Z_1 lies on a 1-dimensional edge on the boundary of a 3-dimensional face F_{D_1}, where D_1 specifies that v_1 coincides with v_1'. If any of the contacts (v_1 or v_2) is broken, there would be two extra degrees of freedom introduced. Therefore, the edge does not lie on the boundary of any 2-dimensional face and is not connected to any edges through 2-dimensional faces.
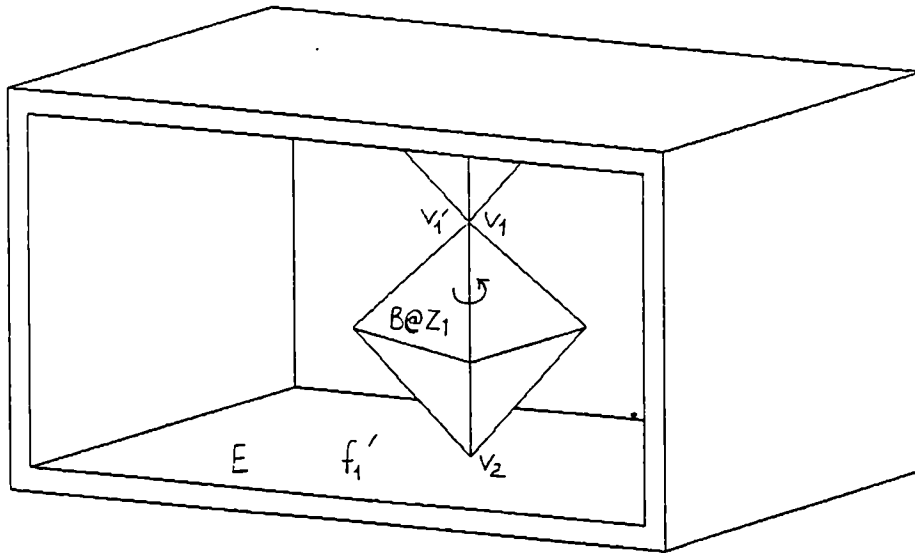
D_1 specifies that v_1 and v_1' coincide. D_2 specifies that, in addition, v_2 lies on f_1'. F_{D_2} is an edge in the boundary of the 3-dimensional F_{D_1}. F_{D_2} does not lie on any 2-dimensional faces.

Figure 3–27: An edge not lying on a 2-dimensional face: Case 1
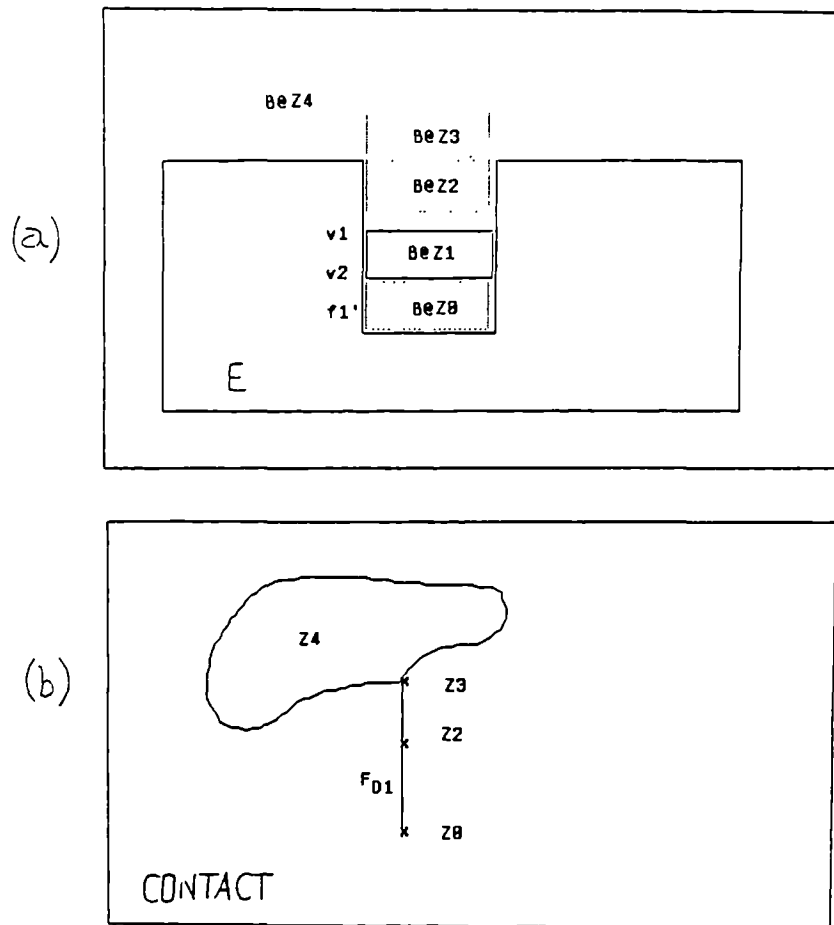
(a) The dimensions of B are such that it fits perfectly in the hole of E. Z_1 satisfies the descriptor $D_1 = (v_1, f_1') \wedge (v_2, f_1')$. In (b), the 'topology' of the resulting contact space is shown. F_{D_1} is an edge which does not lie on the boundary of any 2-dimensional face.

Figure 3–28: An edge not lying on a 2-dimensional face: Case 2

Consider the case shown in Figure 3–28 (2-D with rotations). Object B is at configuration Z_1 in F_{D_1} where $D_1 = (v_1, f_1') \wedge (v_2, f_1')$ (Figure 3–28a). There is one translational d.o.f. and F_{D_1} is en edge of the contact space. If B fits perfectly in the hole then none of the contacts can be broken and no extra d.o.f. gained. The edge F_{D_1} does not lie, thus, in the boundary of a 2-dimensional face of the Contact Space (Figure 3–28b). In order to understand why this happens, let us consider the descriptors $D_2 = (v_1, f_1')$ and $D_3 = (v_2, f_1')$. Because of the tight fit, the set of *legal* configurations of S_{D_2} is equal to F_{D_1} and $E_{D_2} = \emptyset$. Therefore, $F_{D_2} = \emptyset$. Similarly, $F_{D_3} = \emptyset$. Although F_{D_1} does not lie in the intersection of F_{D_2} and F_{D_3}, since these sets are empty, it does lie in the intersection of H_{D_2} and H_{D_3}.

From the last two examples (Figures 3–27 and 3–28) it can be concluded that there might be edges which do not lie on the boundary of a 2-dimensional face. The answer to question (Q4) is, therefore, also no: not all d-dimensional faces lie on the boundary of some $(d+1)$-dimensional face. In the first example, this situation has arisen because any contact broken would result in at least two extra d.o.f. In the second example, the situation has arisen because no contact could be broken without resulting in physical interference.

From the second example, it can also be concluded that there might be d-dimensional faces which don't lie in the intersection of higher dimensional faces, but all faces lie in the intersection of some higher dimensional manifolds defined by some set H_D. 'Dangling' edges, like the edge of Figure 3–28 are not allowed in polytopes, but they don't violate any of the properties of cell complexes.

3.4.5 Assessment of the proposed decomposition

From the discussion in the previous section, we can conclude that the cells (P_D's) of the decomposition do not form a cell complex and the vertices of the space are not necessarily edge-connected. The problems encountered and the reasons why the decomposition is not a cell complex are summarised below:

- There are cases when the boundary of a face is empty. We argued that this happens in the case of 1-dimensional faces when there is one rotational d.o.f. and in the case of 2-dimensional faces when there are two rotational d.o.f.

- Although the cells are path-connected, they are not simply-connected, and therefore, they are not homeomorphic to open d-dimensional balls. As a result, there can be vertices that are not edge connected, although the contact space is path-connected. This happens even in the case when only translations are permitted.

- There can be a vertex on the boundary of a face, which does not lie on any edges in the boundary of the same face. We argued that in this case, although the vertices will not be edge-connected on the boundary of the face, the vertices will be edge-connected in the contact space.

- There can be an edge on the boundary of a 3-dimensional face which does not lie on any 2-dimensional face in the boundary of the 3-dimensional face. In this case, the edge will not be connected to any edges on the boundary of the 3-dimensional face. Moreover, it is possible that the edge will not be connected to any edges at all through 2-dimensional faces.

- Finally, it is possible that an edge does not lie on the intersection of higher-dimensional faces. It was argued that it will, however, lie on the intersection of some higher-dimensional manifolds of the contact space.

The following properties of the decomposition can be proved (proofs are similar to the proofs given in [Hopcroft and Wilfong 84b]):

1. A configuration which lies on a face satisfies the descriptor that defines the face.

2. A configuration which lies on the boundary of a face, lies on a face with dimension less than that of the face.

3. The boundary of a face is exactly those configurations in the face that lie in faces with dimensions less than the dimension of the face.

4. A configuration which lies on the intersection of two faces, lies on a face with dimension less than the minimum of the dimensions of the two faces.

We can now present the answers to the questions posed at the beginning of this section (page 73):

A1: Not all d-dimensional faces have non-empty boundaries $(d \geq 1)$.

A2: The boundary of an d-dimensional face consists of faces of dimension less than d $(d \geq 1)$.

A3: The intersection of two faces consists of faces of dimension less than the minimum of the dimension to the two.

A4: Not all d-dimensional faces lie on the boundary of some $(d+1)$-dimensional face $(d \leq 4)$.

A5: There isn't always a path on the boundary of a face between two configurations on its boundary.

A6: The vertices are not necessarily edge-connected.

A7: The decomposition is not a cell complex.

Having identified the shortcomings of the proposed decomposition, a slightly different decomposition will be described in the next section.
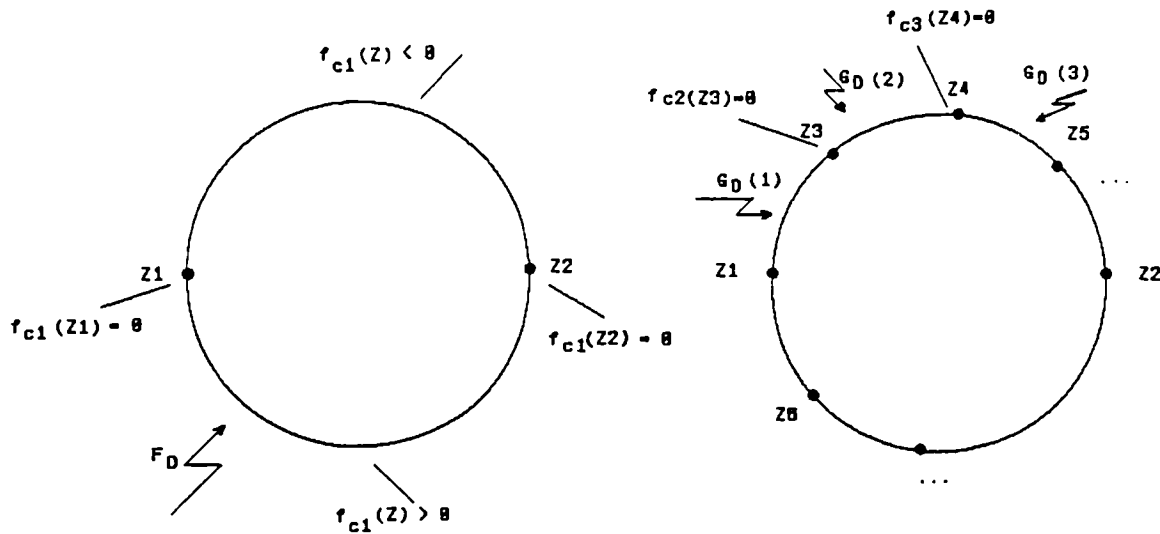
The edge of the contact space F_D (see Figure 3-21) is partitioned into faces $G_D(i)$ by configurations which satisfy some additional constraint. (a) Z_1 and Z_2 lie on the intersection of F_D with H_{D_1}, where $D_1 = c_1 = (v_3, f_1')$; (b) Z_i lies on the intersection of F_D with H_{D_j}, where D_j consists of clauses not in D.

Figure 3–29: Partitioning a face

3.4.6 An alternative decomposition of the Contact Space

Consider again the case shown in Figure 3-21 (page 75) where object B is at configuration Z and it is allowed to rotate about the axis $v_1 v_2$. It was shown that configuration Z lies on an edge F_D with an empty boundary. If we imagine B rotating, there will be two configurations Z_1 and Z_2 at which vertex v_3 will lie on the *plane* of face f_1', although not on the face itself. These are the configurations where F_D intersects H_{D_1}, where $D_1 = c_1 = (v_3, f_1')$. Therefore, $f_{c_1}(Z_1) = f_{c_1}(Z_2) = 0$, while at every other configuration Z on the edge F_D, $f_c(Z) < 0$ or $f_c(Z) > 0$ (Figure 3-29). If we consider all intersections of F_D with H_{D_i}, where D_i consists of clauses which are not in D, then face F_D will be partitioned into smaller edges *with boundaries*. Moreover, for every configuration Z' in the interior of each of the generated smaller edges, the sign of $f_{c_j}(Z')$ for each c_j not in D will remain constant.

The same method could be followed in order to partition a 2-dimensional

face with empty boundary into smaller faces with boundaries. We can conclude that by introducing new 0 to $(d-1)$-dimensional faces which correspond to intersections of a d-dimensional face with higher dimensional manifolds, the problem of faces with empty boundaries (Problem 1) can be overcome. The d-dimensional face F_D is partitioned by these intersections into a number of smaller faces $G_D(1)...G_D(n)$, and the sign of the functions $f_c(Z)$ remains invariant within each face $G_D(i)$.

Let us now focus our attention on Problem 2, namely the problem that there can be faces with 'holes'. As is shown in Figure 3-30, following the same method described above, the 2-dimensional face F_D can be partitioned into a set of faces $G_D(i)$. Such a partitioning results in vertices which are edge-connected. Again it can be noticed that the sign of the functions f_c's remains invariant within each of the new faces of the contact space.

The new decomposition will be defined by formalising the notion of sign invariance of the functions f_c's (see also [Yap ss]).

Definition 3.4.14 *A sign-assignment s is a function from the set of all clauses CLAUSES to the set $\{-1, 0, +1\}$*

$$s : CLAUSES \mapsto \{-1, 0, +1\}.$$

A configuration Z satisfies a sign assignment s, $sat(Z, s)$, if for each clause c in CLAUSES, $f_c(Z)$ is negative, zero or positive according to whether $s(c)$ is -1, 0, or +1. A cell of the contact space consists of configurations for which all the functions f_c remain sign invariant:

Definition 3.4.15 *A cell C_s of the contact space is a set of configurations in the contact space which satisfy a fixed sign-assignment, s.*

$$C_s = \{Z | Z \in CONTACT \land sat(Z, s)\}.$$

For each cell of the contact space there is a unique descriptor D, which is the conjunction of all the clauses c_i for which $s(c_i) = 0$. Therefore, each configuration in a cell satisfies the system of equations of the associated descriptor of the cell and thus $C_s \subset H_D$.

Definition 3.4.16 *A face $G_D(i)$ of the contact space is defined to be the closure of a cell C_s in H_D*

$$G_D(i) = cl(C_s),$$

$$D = \bigwedge_i c_i, \quad \forall c_i : s(c_i) = 0.$$

The **boundary** *of a face is defined to be*

$$cl(C_s) - C_s.$$

Since all configurations in the face are in the contact space,

$$G_D \subset H_D \cap CONTACT.$$

Let C_s be a cell such that $f_{c_i} = 0$, $i \in I_1$, $f_{c_k} < 0, k \in I_2$ and $f_{c_l} > 0, l \in I_3$, where I_1, I_2, I_3 are indices. It can be seen that the closure of the cell C_s, i.e. the face, consists of those configurations for which $f_{c_i} = 0$, $f_{c_k} \leq 0$ and $f_{c_l} \geq 0$. The boundary of a face consists of those configurations of the face for which some of the inequalities become equalities and, therefore, belong to cells of lower dimensions.

Let us now examine through an example how the faces of this decomposition relate to the faces of the previous decomposition. Consider again the situation shown in Figure 3–30a, where object B is allowed to translate but not to rotate. Figure 3–30b shows the contact space. According to the decomposition of Section 3.4.4, F_D is a 2-dimensional face, where $D = c_1 = (v_1, f_1')$. Let $c_i = (v_1, f_i')$. For any configuration Z in F_D:
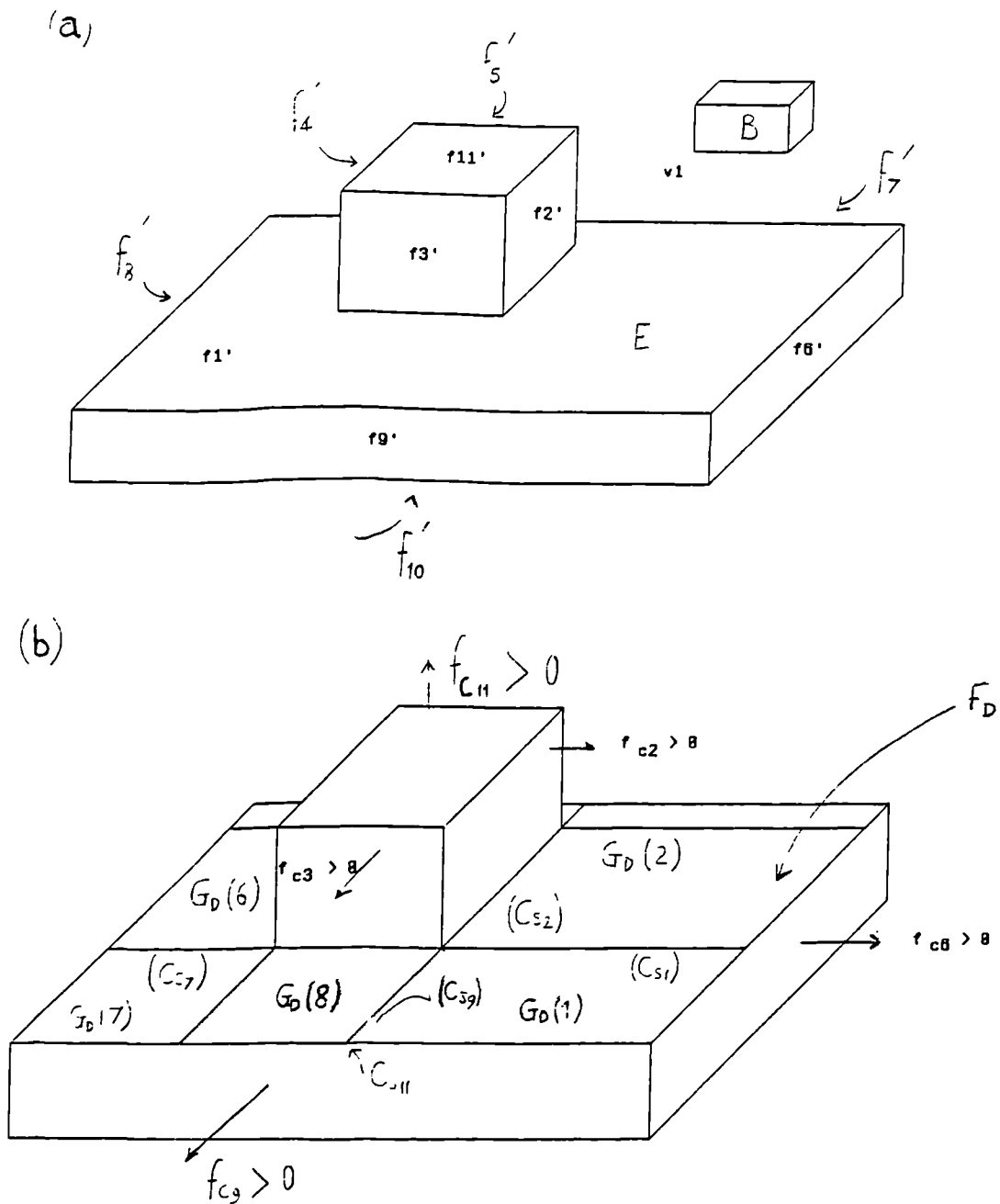
(a) Object B is only allowed to translate; the origin of B is on v_1; (b) The descriptor D specifies that the bottom face of the block is on f_1' of E. The 2-dimensional face F_D of the contact space is partitioned into 2-dimensional faces $G_D(i)$.

Figure 3–30: A sign-invariant decomposition of the space

	$f_{c_1}(z)^*$	$f_{c_2}(z)$	$f_{c_3}(z)$	$f_{c_4}(z)$	$f_{c_5}(z)$	$f_{c_6}(z)$	...	$f_{c_{11}}(z)$
C_{s_1}	0	+	+	−	−	−	...	−
C_{s_2}	0	+	−	−	−	−	...	−
C_{s_3}	0	+	−	−	+	−	...	−
...	...	...	...	...	...	...	...	...
C_{s_8}	0	−	+	−	−	−	...	−
C_{s_9}	0	0	+	−	−	−	...	−
$C_{s_{10}}$	0	+	0	−	−	−	...	−
$C_{s_{11}}$	0	0	+	−	−	0	...	−
...	...	...	...	...	...	...	...	...

$^*c_i = (v, f_i')$

$D_1 = c_1 \wedge c_2$, $D_2 = c_1 \wedge c_3$, $D_3 = c_1 \wedge c_2 \wedge c_3$

Table 3–1: A sign-invariant decomposition of the space

- $f_{c_1}(Z) = 0$, since Z satisfies the descriptor $D = c_1$.

- $f_{c_i}(Z) \leq 0$, for $6 \leq i \leq 11$, since B is always located at the same side of the faces f_i, for $6 \leq i \leq 11$.

- the sign of functions $f_{c_2}, \ldots, f_{c_6}$ varies.

According to the new decomposition, the face F_D is partitioned into 2-cells $C_{s_1}, \ldots, C_{s_8}$ such that configurations within each cell satisfy a unique sign-assignment, as is shown in Table 3–1. The descriptor of these 2-cells is $D = c_1 = (v_1, f_1')$. The closure of each of these 2-cells C_{s_i} is a 2-dimensional face $G_D(i)$, for $1 \leq i \leq 8$. Partitioning the face F_D gives rise not only to cells of the same dimension but also to new cells of lower dimension. For example, as can be seen from Figure 3–30 and Table 3–1, C_{s_9} is a new 1-cell, with descriptor $D_1 = c_1 \wedge c_2$, where $c_2 = (v_1, f_2')$. The closure of C_{s_9} is a 1-dimensional face $G_{D_1}(1)$ which lies in the intersection of the 2-dimensional faces $G_D(1)$ and $G_D(8)$.

We can conclude that, in the proposed new decomposition, the major problems encountered in the previous decomposition are overcome at the expense of introducing more cells. [Yap ss] has proved that this decomposition is a cell complex if the objects are two-dimensional, and they allowed only to translate. The proof is based on the fact that the functions f_c are linear and the cells are homeomorphic to k-dimensional polyhedra. It has to be noted, though, that for the three-dimensional case with rotations, the situation described in Problem 3 is not changed: it is possible for there to be a vertex on the boundary of a 2-dimensional face which does not lie on an edge in the face. Therefore, this decomposition is not a cell complex. However, as has been shown for the case of Figure 3-25 (page 79), even though the vertex is not connected to any edges on the boundary of the 2-dimensional face, it is connected to edges of the contact space. I have not been able to find an example where the vertices of the space resulting in this manner are not edge-connected. I believe, therefore, but have not been able to prove, that the vertices of the space, as defined in Definitions 3.4.15 and 3.4.16 are edge-connected.

3.5 An Algorithm for Constructing the 1-skeleton of the Contact Space

3.5.1 Outline of the Algorithm

In this section an algorithm for the construction of the 1-skeleton of the contact space will be presented, based on the decomposition of the contact space described in Section 3.4.6.

The 1-skeleton of the contact space is the collection of 0-dimensional faces (vertices) and 1-dimensional faces (edges) of the space. A face of the contact space is a subset of the solution space, H_D, of a system of equations $f_{c_i}(Z) = 0$. If the dimension of the solution space of D is zero, then H_D is a 0-dimensional surface in configuration space. If the dimension of the solution space is one, then H_D is a 1-dimensional surface in configuration space. The vertices of the contact space are 0-dimensional surfaces which are legal configurations and for which the objects are in contact. The edges of the contact space are legal, path-connected components of 1-dimensional surfaces.

The construction of the 1-skeleton of the contact space is based on the fact that the surfaces of the contact space can be found by considering only the solutions of systems of equations in configurations, while ignoring the aspects of physical occupancy of objects and body interference. Subsequently, the faces can be found by identifying legal components of the surfaces.

The algorithm consists of two main stages. In the first stage only the surfaces of the contact space are considered, by solving the systems of equations associated with the descriptors of the space. Section 3.5.2 examines how the solution sets of the descriptors of the contact space are related and it is shown that they form a lattice. Section 3.5.3 describes an algorithm for finding the surfaces of the contact space. In the second stage, the vertices of the space are found by considering which 0-dimensional surfaces are legal configurations. Then the edges of

the space are found by partitioning the 1-dimensional surfaces of the space into path-connected, legal components. This process is described in Section 3.5.4.

3.5.2 The Surfaces of the Contact Space

It was noted in Section 3.2.6 that if the moving object is in one of the basic types of the contact with the environment, then its configuration is constrained to lie on a 5-dimensional connected manifold in configuration space. That is to say, the solution set, H_D, of the corresponding 5-dimensional descriptor, D, is a 5-dimensional connected manifold or surface[4], whose equation is given by $f_c(Z) = 0$, where $D = c$. In this section it will be shown that the solutions sets H_D of all possible consistent descriptors constitute a lattice.

A partially ordered set M is called a lattice if any pair of elements of the set has a *least upper bound* and a *greatest lower bound* [Birkhoff and MacLane 65]. The least upper bound of two elements a, b is called their *join* and is usually denoted by $a \sqcup b$. The greatest lower bound of the two elements is called their *meet*, denoted by $a \sqcap b$. If every subset of the ordered set has a meet and a join, the lattice is said to be *complete*. A complete lattice always has a greatest element, which is called the *unit element* and a least element, which is called the *zero element*. In addition, every finite lattice is complete. For the operations meet and join of a lattice, as can be proved from the above definitions, the commutative, associative and absorption laws hold.

Consider the set of all possible consistent descriptors

$$DESCRIPTORS = \{D_1, D_2, \ldots, D_n\},$$

so that $H_{D_i} \neq \emptyset$, and the set

$$L_h = \{C, H_{D_1}, \ldots H_{D_n}, \emptyset\},$$

---

[4]The term n-dimensional surface will be employed to denote a n-dimensional connected manifold.

where H_{D_i} is the solution set of a descriptor D, and C is the configuration space. The set L_h is partially ordered, where the ordering is defined by set inclusion, i.e.

$$H_{D_i} \leq H_{D_j} \text{ iff } H_{D_i} \subseteq H_{D_j}.$$

The element C is the greatest (unit) element of L_h and the empty set is the least (zero) element of L_h. It will be shown that the set L_h constitutes a lattice.

Theorem 3.4 *The set* $(L_h, \leq)$ *where*

$$L_h = \{C, H_{D_1}, \ldots, H_{D_n}, \emptyset\},$$

$$H_{D_i} \leq H_{D_j} \text{ iff } H_{D_i} \subseteq H_{D_j}$$

is a lattice with the operations

$$H_{D_i} \sqcap H_{D_j} = H_{D_i} \cap H_{D_j}$$

$$H_{D_i} \sqcup H_{D_j} = \begin{cases} H_{D_i} \cup H_{D_j} & \text{if } \exists c_k : c_k \text{ in } D_i \text{ and } c_k \text{ in } D_j \\ C & \text{otherwise} \end{cases}.$$

Proof

Let $a = H_{D_i} \sqcap H_{D_j}$. It is easy to see that $H_{D_i} \cap H_{D_j} = H_{D_i \wedge D_j}$, that it is to say, a is the solution set to the system of equations corresponding to the descriptor which is the conjunction of the clauses of the two descriptors. If the descriptor $D_i \wedge D_j$ is inconsistent, then $a = \emptyset$, and a is the greatest lower bound of H_{D_i} and H_{D_j}. If, on the other hand, the descriptor $D_i \wedge D_j$ is consistent, then a is an element of L_h, since L_h includes the solution sets of all consistent descriptors. Moreover, a is the greatest lower bound of H_{D_i} and H_{D_j}, since the ordering of the set L_h is defined by set-theoretic inclusion.

Let $a = H_{D_i} \sqcup H_{D_j}$. If D_i and D_j have some clauses in common, then a is the solution set of the system of equations f_{c_k}, where c_k is a clause both in D_i and D_j. Since a subsystem of a satisfiable system of equations is always satisfiable,

it follows that a is a member of L_h. Also, it is the least upper bound, because the order is defined by set-theoretic inclusion. If, on the other hand, D_i and D_j do not have any clauses in common, then a is the whole configuration space. $\square$

With each set H_D we can associate a *unique* set of clauses n_D. The set n_D consists of all the clauses c_i for which the corresponding function $f_{c_i}(Z)$ is zero-valued for every configuration Z in H_D. In other words, the set n_D is the union of the clauses of all descriptors D_i with solution set equal to H_D. Recall that two descriptors have been defined to be equivalent (Definition 3.4.8, page 65) if they have the same set of solutions. The set n_D can thus be defined by

$$n_D = \bigcup_{D_i \in \tilde{D}} clauses(D_i). \tag{3.2}$$

where $\tilde{D}$ denotes the equivalence class in $DESCRIPTORS$ generated by D and

$$clauses(D) = \{c_i | c_i \ in \ D\}.$$

I will assume that there isn't a configuration for which the associated functions of *all* clauses are zero-valued, that is, I will assume that the system of *all* equations $f_{c_i}(Z) = 0$ is inconsistent. The justification for this assumption is that if this wasn't the case, then the number of vertices of the contact space would be equal to one: the only vertex would be the configuration Z for which $f_{c_i}(Z) = 0$ for all c_i. This situation is possible but it is definitely an exceptional one. Having made this assumption, all sets of inconsistent clauses can be mapped to the set $CLAUSES$.

Consider now the set

$$L_c = \{\emptyset, n_{D_1}, \ldots, n_{D_n}, CLAUSES\}.$$

The set L_c is partially ordered,

$$n_{D_i} \leq n_{D_j} \text{ iff } n_{D_i} \supseteq n_{D_j}. \tag{3.3}$$

The element $CLAUSES$ is the least (zero) element of L_c and the empty set is the greatest (unit) element of L_c. It can be seen that $(L_c, \leq)$ is a lattice isomorphic to the lattice $(L_h, \leq)$. The meet and join operations of L_c are defined as follows:

$$n_{D_i} \sqcap n_{D_j} = \begin{cases} n_{D_i \wedge D_j} & \text{if } D_i \wedge D_j \text{ is a consistent descriptor} \\ CLAUSES & \text{otherwise} \end{cases}$$

$$n_{D_i} \sqcup n_{D_j} = n_{D_i} \cap n_{D_j}.$$

Figure 3–31 shows a 2-dimensional example. In Figure 3–31a, the moving and the stationary objects are shown. In Figure 3–31b, a part of the lattice is shown. For each element H_{D_m} of the lattice L_h the corresponding element n_{D_m} of the lattice L_c is also shown. From the figure it can be seen, for example, that:

$$H_{D_1} \sqcap H_{D_3} = H_{D_8} = H_{D_1 \wedge D_2 \wedge D_3},$$

$$H_{D_{10}} \sqcap H_{D_i} = \emptyset$$

$$H_{D_8} \sqcup H_{D_9} = H_{D_5}$$

$$H_{D_5} \sqcup H_{D_7} = C$$

From the above discussion it can be concluded that all the solution sets can be found by considering the solution sets of all 5-dimensional descriptors and their intersections. In the next section, an algorithm is sketched which performs these operations. It has to be noted, that the intersection of two 5-dimensional manifolds is not necessarily a manifold. [5] In addition, even if the intersection is a manifold, it is not necessarily a connected manifold. Since we are interested in path-connected components of the contact space, we are interested in identifying those subsets of a solution set H_D which are connected manifolds. As an example, consider the set H_{D_5} in Figure 3–31. It can be seen, that the configurations in H_{D_5} lie on three lines: $x = 0, y = 0$; $y = 0, \theta = 0$; $y = 0, \theta = 180$. In cases

---

[5]The class of manifolds in not closed under set operators. They are closed under the operation called *connected sum*.

Figure 3–31: Lattice of the solution sets H_{D_i}
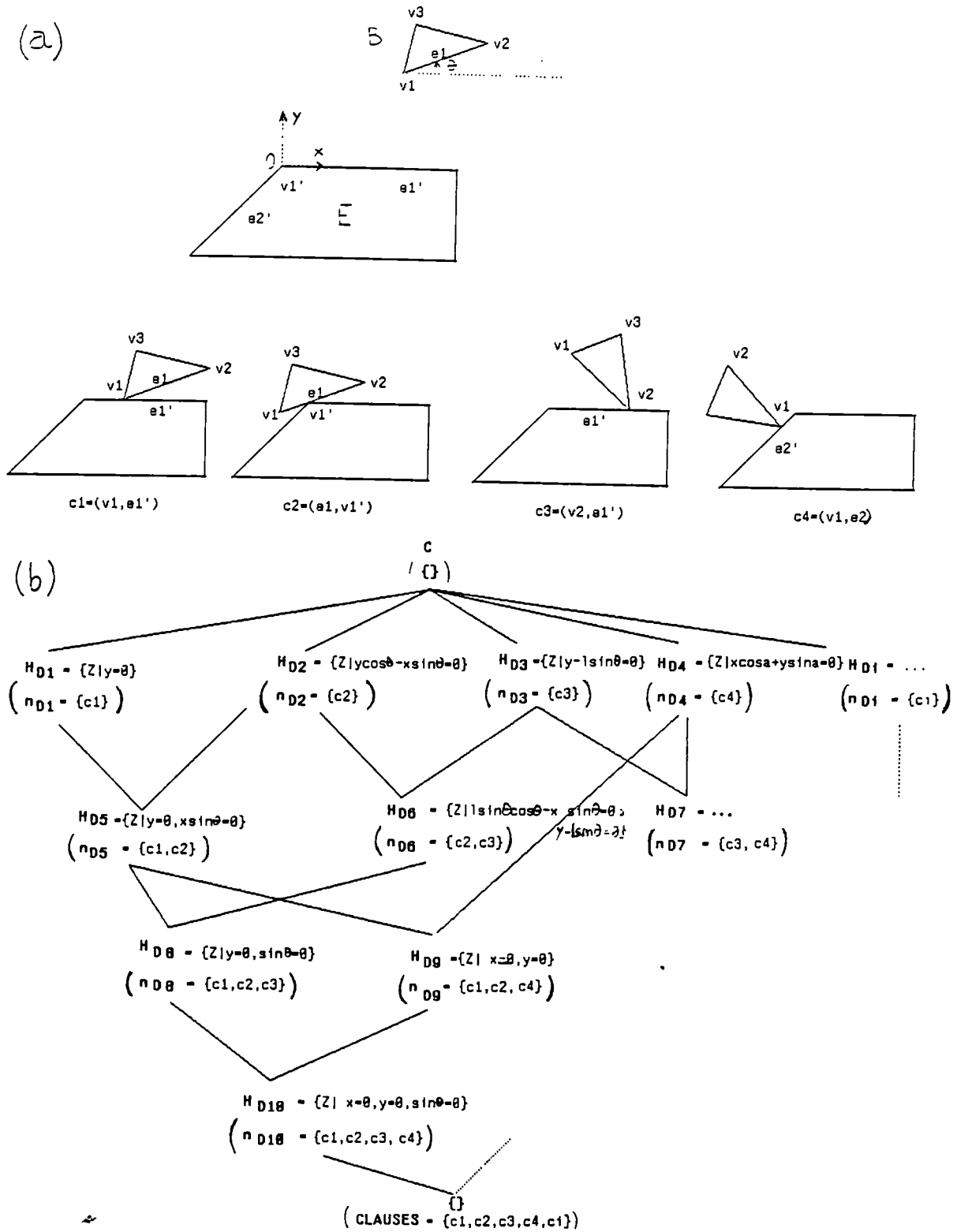
In (a) the two objects are shown. The configuration of B is specified by $Z = (x, y, \theta)$, where x and y are the coordinates of vertex v_1 with respect to the axes Oxy. In (b), part of the lattice of the solution sets is shown. l, a are constants. With each solution set H_{D_m} there is an associated set of clauses n_{D_m}. c_i is some clause such that the solution set $H_{D_1 \wedge D_2 \wedge D_3 \wedge D_4 \wedge D_i}$ is the empty set, e.g. $c_i = (v_3, e_1')$.

Figure 3–32: Graph of the surfaces of the contact space

See also Figure 3–31. The meet of the elements n_{D_1} and n_{D_2} consists of three elements. The node H_{D_3} of Figure 3–31 is represented by these three elements, while the node H_{D_4} is represented by the two elements marked n_{D_6}. Note that the labelling by means of clauses is not unique. The graph in this figure is not a lattice.

like this, we wish to represent separately these subsets of the solution set, so that 1-dimensional manifolds can be parameterised and partitioned into edges. If such a representation is employed, however, then the resulting structure is not a lattice. This can be seen from the fact that two elements could meet in more than one element (see Figure 3–32). In addition, the labelling of the elements by means of the sets n_D is no longer unique.

In Chapter 4, where the solution of the system of equations is considered, it will become clear when and why situations similar to the one depicted in Figure 3–32 occur. Moreover, it will be shown how the subsets of a set H_D are chosen.

3.5.3 An algorithm for finding the surfaces of the contact space

Starting from the set of all possible clauses ($CLAUSES$), the algorithm has to find all possible consistent combinations of clauses and to establish which combinations are equivalent. The algorithm is summarised below:

1. Find all clauses, $CLAUSES$. All 5-dimensional descriptors are thus found and inserted in the graph.

2. Repeat for $d = 5, 4, \ldots, 1$:

   For each pair of an d-dimensional descriptor D_i with a 5-dimensional descriptor D_j, set $n_D = clauses(D_i) \cup clauses(D_j)$ and :

   (a) examine if the descriptor $D = D_i \wedge D_j$ is consistent. If it is not consistent then n_D is discarded.

   (b) If it is consistent, examine if D is equivalent to the descriptor of some element n_k of the graph. In the case that such an element is found, then n_k becomes the union of the clauses of the equivalent descriptors.

   (c) If D is consistent and it is not equivalent to any descriptors already in the graph, then:

      i. if the solution set H_D is a connected manifold, then n_D is a new element in the graph.

      ii. otherwise find the connected subsets of H_D, and for each one of them create a new element in the graph.

From the above we can conclude the following operations are required by the algorithm:

1. Deciding whether a descriptor is consistent, i.e. whether a system of equations is consistent.

2. Deciding whether two descriptors are equivalent.

3. Finding the dimension of a descriptor

4. Finding the connected subsets of a solution set.

These operations are performed by examining the system of equations of the descriptors. In Chapter 4 a spatial reasoning system is presented, which is able to perform the above operations.

The algorithm, described in this section is further refined and explained in Chapter 5, after the spatial reasoning system has been presented.

3.5.4   Finding the edges and vertices of the space

The faces of the contact space are path-connected components of the surfaces of the contact space, such that every configuration in a face is a configuration for which the two objects are in contact but do not overlap. According to the formal definition of a face (Definition 3.4.16), for every configuration in the interior of a face the sign of *each* function f_c remains invariant, unless the function is zero-valued. That is, if Z_1, Z_2 are two arbitrary configurations in the interior of a face, then either $f_{c_i}(Z_1) = f_{c_i}(Z_2) = 0$ or $f_{c_i}(Z_1) < 0$ and $f_{c_i}(Z_2) < 0$ or $f_{c_i}(Z_1) > 0$ and $f_{c_i}(Z_2) > 0$. Furthermore, it has been claimed (Section 3.4.6) that the boundary of a face consists of those configurations for which some of these inequalities become equalities. That is, every configuration in the boundary of a face lies in the intersection of the surface of the face with some 5-dimensional surface. Therefore, a surface of the contact space is partitioned into faces by the intersections of the surface with 5-dimensional surfaces.

In this section I will examine the problem of finding the vertices and edges of the contact space. As it will become clear, the method can not, however, be generalised to higher dimensional cases. The vertices of the space can be found by examining whether the configuration of each 0-dimensional surface is in the contact space, that is by examining whether it is a legal configuration for which the objects are in contact. In order to find the edges of the contact space which lie on some 1-dimensional surface, first we have to find all the configurations

which are the boundaries of the edges. Then, these configurations are used to partition the 1-dimensional surface into edges.

The boundaries of the edges on some 1-dimensional surface are simply the vertices which lie on the surface. Having constructed the lattice of the contact space, we can easily establish whether or not a vertex lies on 1-dimensional surface by examining if the element of the lattice corresponding to the vertex is a lower neighbour of the element corresponding to the 1-dimensional surface.

It will now be shown that if Z_1 and Z_2 are vertices which lie on a 1-dimensional surface such that there is not another vertex 'between' these two and if some arbitrary configuration 'between' the two vertices is a configuration in the contact space, then the open interval on the surface defined by Z_1 and Z_2 is an edge of the space. First it will shown that if all configurations on the interval are in the contact space then the interval is an edge.

Definition 3.5.1 *Let $P(t)$ be the parametric equation of 1-dimensional surface of the space and $Z_1 = P(t_1)$, $Z_2 = P(t_2)$ be vertices such that $t_1 \leq t_2$ and $Z = P(t_i)$ is not a vertex forall $t_i : t_1 < t_i < t_2$. The set S is defined to be*

$$S = \{Z | Z = P(t_i), \ \forall t_i : \ t_1 < t_i < t_2\}.$$

Lemma 3.5.1 *If $S \subset CONTACT$ then S is an edge.*

Proof

Let n_D be the element in the lattice corresponding to the surface. Let us assume that there is a configuration Z in S for which $f_{c_i}(Z) = 0$ and c_i is not a member of n_D. Then $n_D \cup \{c_i\}$ is a lower element of n_D and Z is a 0-dimensional surface which lies on the surface and since $Z \in CONTACT$, Z is a vertex. This is contrary to the assumption that S does not contain any vertices. Thus

$$\forall Z \in S, \ f_{c_i}(Z) = 0 \text{ iff } c_i \in n_D.$$

Let us now assume that there are two configurations Z', Z'' such that $f_{c_i}(Z') < 0$ and $f_{c_i}(Z'') > 0$ for some clause c_i not in n_D. Since the functions f_c are continuous, then there must be a configuration Z in S for which $f_{c_i}(Z) = 0$. As it has been proved this is not possible. Therefore, all functions f_{c_i} remain sign-invariant in S and S is an edge. □

It can also be seen that if there is an illegal configuration and a legal configuration in S, then there must be a legal configuration in S for which some function becomes zero-valued. Intuitively, this happens because during a transition from a legal to an illegal configuration some contact has to be made first. Similarly for the case of a transition from a configuration in the contact space to a configuration in the free space. Since we have assumed that there are no vertices in S, if there is a configuration Z in S which is both legal and one for which the objects are in contact, i.e. Z is in $CONTACT$, then all configurations in S must be in $CONTACT$:

Claim: If $\exists Z \in CONTACT \cap S$ then $S \subset CONTACT$.

Theorem 3.5 *If $\exists Z \in CONTACT \cap S$, then S in an edge.*

Proof

It follows directly from Lemma 3.5.1 and the above claim. □

In section 3.4.6 it was has been argued that the boundary of an edge consists of exactly two vertices. An algorithm for finding the edges which lie on a 1-dimensional surface can now be outlined:

1. Find the vertices which lie on the surface.

2. Find the value of the parameter of the surface for each vertex.

3. Order the values of the parameters.

4. Partition the surface into intervals, according to the values of the parameters.

5. For each interval, examine if an arbitrary configuration is a configuration in the contact space. If this is the case, then the interval defines an edge on the surface.

In order to find the vertices and edges of the space, we need to be able to decide whether some configuration is in CONTACT. I the implemented algorithm, as it will be described in more detail in Chapter 5, a solid modeller has been employed for this purpose.

3.6 Summary

Two methods have been proposed for decomposing the contact space into faces of various dimensions, using interactions among the features of the moving object and the environment. All such interactions can be expressed as conjunctions of three basic types of interactions: (a) vertex-face; (b) face-vertex; (c)edge-edge. Each basic interaction was called a clause and conjunctions of clauses were called descriptors. With each clause there is an associated function. Configurations for which this function is zero-valued lie on some d-dimensional manifold. Intersecting these manifolds, we partitioned the contact space into faces of various dimensions. 0-dimensional faces were called vertices and 1-dimensional faces were called edges. The boundary of each face consists of faces of lower dimensions.

The first decomposition is an extension to 3-dimensions and rotational motion of the decomposition described in [Hopcroft and Wilfong 84b]. According to that method, a transition from one face to a face on its boundary or vice-versa would imply that at least one contact had been established or broken. This method gave rise to 0-dimensional faces which were not connected by 1-dimensional faces. In the second decomposition, we introduce the idea of 'imaginary faces', in order to partition a face into smaller faces. In this case, the transition from a face to a face on its boundary does not necessarily imply that a contact has been established or broken, but that the value of the function corresponding to some contact becomes zero.

Neither of the proposed decompositions is a cell complex. However, through a series of examples, it was argued that the second decomposition results in vertices that are edge-connected, with the possible exception of some pathological cases.

Finally, an algorithm has been outlined for finding the vertices and edges of the contact space according to the second decomposition. The algorithm consists of two stages. In the first stage the lattice of the surfaces of the space is constructed and in the second stage the vertices and edges are found. In the first

stage a spatial reasoning system is used in order to find the intersections of the surfaces of the space. In the second stage, a solid modeller is used in order to decide whether a configuration is legal. The algorithm and its implementation is described in more detail in Chapter 5, after the spatial reasoning system is presented.

Chapter 4

A Spatial Reasoning System

4.1   Introduction

The spatial reasoning system presented in this chapter is used in this research for the construction of the 1-skeleton of the contact space, as described in the previous chapter. The requirements placed upon the system by the algorithm presented in Section 3.5 are twofold:

- Given a pair of constraints on the relative location of two objects, to check if these constraints can be satisfied simultaneously, and, if this is the case, to replace them by an equivalent one.

- To parameterise all constraints with one degree of freedom in the relative location of the objects, so that the range of the parameter can be partitioned into intervals, corresponding to sets of locations for which the objects do not interfere.

The spatial reasoning system is based on RAPT [Corner, Ambler, and Popplestone 83]. RAPT is a robot programming language, in which an assembly task is described in terms of spatial relationships holding among features of objects. The RAPT system is centered around an inference engine which infers the locations of objects from the specified spatial relationships. There are two versions of the RAPT system: the algebraic system and the Cycle Finder. In the algebraic system [Popplestone, Ambler, and Bellos 80] spatial relationships are transformed into location equations and these equations are then solved. In the

Cycle Finder [Popplestone and Ambler 83], the assembly description is regarded as a graph where nodes represent 'body instances', that is, objects at particular locations, and the arcs represent relationships among 'body instances'. When the relative location of two body instances becomes known, then they are merged into one conglomerate object. Since the objective of the system is to find the relative locations of all body instances with respect to some reference object, the objective of the Cycle Finder is to merge all the nodes to one node, representing the reference object.

This process involves two basic operations. The first one requires the substitution of two relationships holding between the features of two objects by an equivalent more constrained relationship. A 2-cycle of relationships in the graph is, thus, substituted by a single arc. If the inferred relationship is a relationship which does not allow any relative motion between the objects, that is, if the relationship can be expressed as some constant relative location, then the two nodes are merged. Given a pair of relationships between two body instances with a third one, the second operation involves the inference of a relationship holding between these two body instances. The two operations are performed by means of tables. In the first table, which will be referred to as the Substitution Table, there is an entry for various combinations of two types of relationships. For each such pair, there are rules, based on the geometry of the situation, for the inference of the type of the equivalent relationship and the location of the features between which the new relationship holds. The second table is similarly defined.

The spatial reasoning system described in this chapter is an extension of the Substitution Table of the RAPT inference engine. A formal method has been developed for defining all types of spatial relationships among polyhedral objects and for defining the rules which determine when a relationship which is equivalent to a pair of relationships between two objects may be found.

The use of a table is preferable to the use of an algebraic system which would solve the location equations because it increases significantly the speed of the operation. Although the system is basically a geometric system, in the sense

that the rules are described in terms of geometric relations between features, the substitution table has been constructed partially by the algebraic solution of location equations.

The substitution table clearly fulfills the first requirement for the reasoning system in this research, as specified in the beginning of this section. The second requirement, the parameterisation of one degree of freedom relationships will be treated separately.

Locations of objects and spatial relationships among their features are represented in terms of transformations. In Section 4.2 an overview of transformations and their representations in this research is presented. Section 4.3 introduces the concept of spatial relationships. Section 4.4 presents an algebraic method for finding a relationship equivalent to a pair of relationships, while Section 4.5 presents a geometric method. The two methods are compared in Section 4.6. The parameterisation of one degree of freedom relationships is described in Section 4.7. Finally, the constructed Substitution Table and its implementation is described in Section 4.8.

The substitution table and the main geometric functions it invokes, can be found in Appendix A.

4.2 Homogeneous Transformations

4.2.1 Definitions

Homogeneous transformations can be used to describe the position and orientation of coordinate systems in space. The position and orientation of an object can be described in terms of transformations by embedding a coordinate system in it.

In general, a homogeneous transformation H in $\mathcal{E}^3$ is a mapping from $\mathcal{E}^3$ onto itself and is represented by a 4×4 matrix. A homogeneous transformation can describe rotation, translation, stretching and perspective transformations. In motion planning for rigid objects, we are interested in rigid transformations, i.e. transformations which preserve distances and signed angles. Specifically, we are interested in rotation and translation transformations. In this section, I will summarise some of the basic definitions and present the notation used in this thesis. A thorough account of transformations in relation to robotics can be found in [Paul 81]. The notation used here follows the conventions described in [Ambler and Popplestone 75].

A point vector

$$\mathbf{v} = a\mathbf{i} + b\mathbf{j} + c\mathbf{k}$$

is represented in homogeneous coordinates as a row vector

$$(x, y, z, w),$$

where $a = x/w$, $b = y/w$ and $c = z/w$. The shorthand notation (x, y, z) will be used to denote a row vector $(x, y, z, 1)$.

Given a vector **v**, its transformation **u** by H is represented by the matrix product[1]

$$\mathbf{u} = \mathbf{v}H.$$

A transformation corresponding to a translation by a vector $\mathbf{d} = a\mathbf{i} + b\mathbf{j} + c\mathbf{k}$ is represented by the matrix

$$trans(\mathbf{d}) = trans(a, b, c) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ a & b & c & 1 \end{pmatrix} \tag{4.1}$$

A transformation corresponding to a rotation about a vector **u** by an angle θ is denoted as $Rot(\mathbf{u}, \theta)$. In this thesis, a rotation by θ about the x-axis is denoted by $twix(\theta)$ and a rotation by $\pi/2$ about the z-axis by XTOY [Popplestone, Ambler, and Bellos 80],

$$Rot(\mathbf{x}, \theta) = twix(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.2}$$

$$Rot(\mathbf{z}, \pi/2) = XTOY = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.3}$$

The elements of a transformation matrix can be interpreted as four vectors describing a coordinate system. Therefore, a coordinate system $O_1 x_1 y_1 z_1$ is represented by the matrix

---

[1]Note that throughout this thesis postfix notation for matrix multiplication is used, as a result of this definition.

$$\begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ o_1 \end{pmatrix},$$

where x_1, y_1 and z_1 are the direction vectors of the x, y and z axes respectively and o_1 is a point vector representing the origin. In particular,

$$W = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} i \\ j \\ k \\ 0 \end{pmatrix}$$

represents the reference coordinate system, the "world" axes.

Given two coordinate systems with the same origin, the orthonormal matrix

$$\hat{H} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.4}$$

represents a rotation from the first coordinate system to the second, where r_{ij} is the cosine of the angle between the ith axis of the rotated coordinate system and the jth axis of the original system. An orthonormal matrix will be denoted by the symbol being accented (e.g. $\hat{E}$).

We will refer to a homogeneous transformation which is the product of a translation and a rotation as a *location*. From Equations 4.1 and 4.4 it follows that the general form of a location is

$$H = \hat{H} trans(\mathbf{d}) = \begin{pmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ a & b & c & 1 \end{pmatrix}. \tag{4.5}$$

If W_1 is a coordinate system and p a location, then $W_1 p$ is the coordinate system W_1 transformed by p.

The transformations corresponding to a rotation about the x axis by θ and a rotation about the z axis by $\pi/2$ are given by Equations 4.2 and 4.3 respectively. In the next section it will be shown that a rotation about any vector can be represented in terms of these two types of rotations. Then, in Section 4.2.3 it will be shown that a rotation between two coordinate systems can also be represented in terms of *twix* and *XTOY*.

4.2.2 Rotation about a vector

The transformation representing a rotation $Rot(\mathbf{u}, \theta)$ about an arbitrary vector $\mathbf{u} = (k_x, k_y, k_z)$ located at the origin is given by the matrix [Paul 81] [2],

$$
\begin{pmatrix}
k_x k_x \text{vers}\theta + \cos\theta & k_x k_y \text{vers}\theta + k_z \sin\theta & k_x k_z \text{vers}\theta - k_y \sin\theta & 0 \\
k_y k_x \text{vers}\theta - k_z \sin\theta & k_y k_y \text{vers}\theta + \cos\theta & k_y k_z \text{vers}\theta + k_x \sin\theta & 0 \\
k_z k_x \text{vers}\theta + k_y \sin\theta & k_z k_x \text{vers}\theta - k_x \sin\theta & k_z k_z \text{vers}\theta + \cos\theta & 0 \\
0 & 0 & 0 & 1
\end{pmatrix},
$$

where $\text{vers}\theta = (1 - \sin\theta)$. In this section it will be shown that the only two kinds of rotations needed for the representation of rotations are those transformations of the form $twix(\alpha)$ and $XTOY$ (Equations 4.2 and 4.3).

Theorem 4.1 *The group of all rotations in 3-space can be generated by the set* $\{twix(\alpha)\}_{\alpha \in \Re} \cup \{XTOY\}$.

Proof [Arai 81]

Let $\mathbf{u}$ be a unit vector located at the origin, α the oriented angle between the i axis and $\mathbf{u}$, and β the oriented angle between the j axis and the projection of $\mathbf{u}$ on the $i = 0$ plane (see Figure 4–1). Then $\mathbf{u} = iQ_u$, where Q_u is the transformation

$$
Q_u = XTOY\,twix(\alpha)\,XTOY^{-1}twix(-(\pi/2 - \beta)). \tag{4.6}
$$

Substituting for $twix$ and $XTOY$ from Equations 4.2 and 4.3 into Equation 4.6 and then multiplying out,

$$
Q_u = \begin{pmatrix}
\cos\alpha & \sin\alpha\cos\beta & \sin\alpha\sin\beta & 0 \\
0 & \sin\beta & -\cos\beta & 0 \\
-\sin\alpha & \cos\alpha\cos\beta & \cos\alpha\sin\beta & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}.
$$

----

[2]The matrix in [Paul 81] has been transposed.
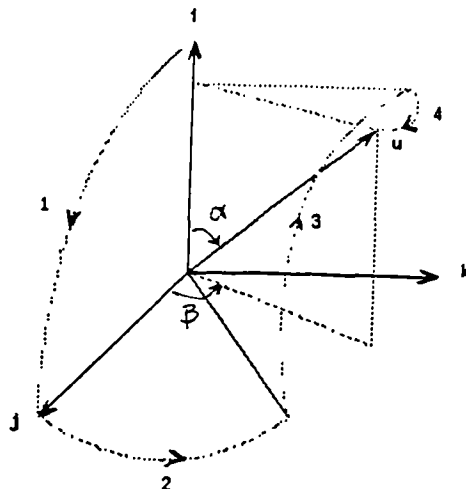
Figure 4–1: Rotation about a vector u

A rotation by θ about **u** can be expressed as

$$Rot(\mathbf{u}, \theta) = Q_u^{-1} twix(\theta) Q_u. \tag{4.7}$$

Using Equations 4.6 and 4.7, we obtain

$$Rot(\mathbf{u}, \theta) = twix(\pi/2 - \beta) XTOY twix(-\alpha) XTOY^{-1}$$
$$twix(\theta) XTOY twix(\alpha) XTOY^{-1} twix(-\pi/2 + \beta). \tag{4.8}$$

Since

$$XTOY^{-1} = twix(\pi) XTOY twix(\pi), \tag{4.9}$$

from Equation 4.8 it follows that a rotation about any vector can be represented in terms of the transformations twix and XTOY. □

4.2.3 General Representation of a Rotation

Theorem 4.2 *Any transformation, which is the product of a rotation and a translation, between two coordinate systems can be represented by*

$$m = twix(\psi)XTOYtwix(\phi)XTOYtwix(\theta)trans(d). \tag{4.10}$$

Proof [Arai 81]

Consider two coordinate systems $O_1x_1y_1z_1$ and $O_2x_2y_2z_2$. Let m be a transformation from the coordinate system $O_1x_1y_1z_1$ to the coordinate system $O_2x_2y_2z_2$. I will first consider the rotational part of m, $\hat{m}$.

Let us first examine the case when the x axes of the coordinate systems are not parallel. Let

$$\begin{aligned}
\mathbf{x_1} \times \mathbf{x_2} &= \mathbf{y_3} \neq 0, \\
\mathbf{y_1} \cdot \mathbf{y_3} &= \cos\alpha, \\
\mathbf{x_1} \cdot \mathbf{x_2} &= \cos\beta, \\
\mathbf{y_3} \cdot \mathbf{y_2} &= \cos\gamma
\end{aligned} \tag{4.11}$$

where α, β, γ are oriented angles as shown in Figure 4–2. It will be proved that

$$\hat{m} = twix(\gamma - \pi)XTOYtwix(\beta - \pi)XTOYtwix(\alpha).$$

Consider the coordinate systems $O_3x_3y_3z_3$ and $O_4x_4y_4z_4$, where $\mathbf{x_3} = \mathbf{x_1}$, $\mathbf{x_4} = \mathbf{x_2}$, $\mathbf{y_4} = \mathbf{y_3}$ and $O_3 = O_4$ is the foot of the perpendicular from O_1 to the intersection of the $x = 0$ planes of $O_1x_1y_1z_1$ and $O_2x_2y_2z_2$. The transformation m is the composite of the transformations H_1, H_2 and H_3

$$O_1x_1y_1z_1 \xrightarrow{H_1} O_3x_3y_3z_3 \xrightarrow{H_2} O_4x_4y_4z_4 \xrightarrow{H_3} O_2x_2y_2z_2$$

and, thus,

$$\hat{m} = \hat{H}_1\hat{H}_2\hat{H}_3. \tag{4.12}$$
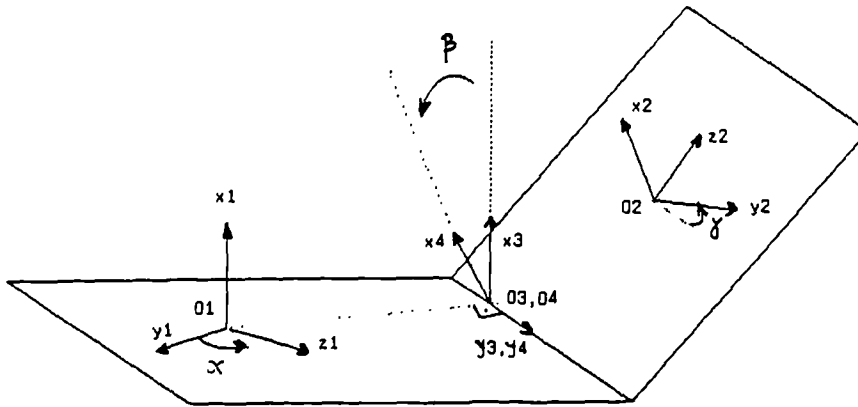
Figure 4–2: Transformation from coordinate system $O_1 x_1 y_1 z_1$ to $O_2 x_2 y_2 z_2$

It can be seen that

$$
\begin{aligned}
\hat{H}_1 &= twix(\alpha), \\
\hat{H}_2 &= Rot(\mathbf{y}_3, \beta), \\
\hat{H}_3 &= Rot(\mathbf{x}_4, \gamma)
\end{aligned}
\tag{4.13}
$$

and, thus,

$$
\hat{m} = twix(\alpha) Rot(\mathbf{y}_3, \beta) Rot(\mathbf{x}_4, \gamma).
\tag{4.14}
$$

Since

$$
\mathbf{y}_3 = \mathbf{y}_1 twix(\alpha) = \mathbf{x}_1 XTOY twix(\alpha)
$$

we can substitute $Q_u = XTOY twix(\alpha)$ in Equation 4.7 and express $Rot(\mathbf{y}_3, \beta)$ as

$$
Rot(\mathbf{y}_3, \beta) = (XTOY twix(\alpha))^{-1} twix(\beta)(XTOY twix(\alpha)).
\tag{4.15}
$$

Similarly we obtain

$$Rot(\mathbf{x}_4, \gamma) = (Rot(\mathbf{y}_3, \beta))^{-1} twix(\gamma)(Rot(\mathbf{y}_3, \beta)). \tag{4.16}$$

From Equations 4.14, 4.15 and 4.16 follows that

$$
\begin{aligned}
\hat{m} &= twix(\alpha) Rot(\mathbf{y}_3, \beta) Rot(\mathbf{x}_4, \gamma) \\
&= twix(\alpha + \gamma) Rot(\mathbf{y}_3, \beta) \\
&= twix(\gamma) XTOY^{-1} twix(\beta) XTOY twix(\alpha).
\end{aligned}
\tag{4.17}
$$

Using Equation 4.9, Equation 4.17 can be rewritten as

$$\hat{m} = twix(\gamma - \pi) XTOY twix(\beta - \pi) XTOY twix(\alpha). \tag{4.18}$$

Finally, substituting $\theta = \alpha$, $\phi = \beta - \pi$ and $\psi = \gamma - \pi$ we obtain

$$\hat{m} = twix(\psi) XTOY twix(\phi) XTOY twix(\theta). \tag{4.19}$$

The case when the x axes of the coordinate systems are parallel has to be considered separately since $\mathbf{x}_1 \times \mathbf{x}_2 = 0$. In this case we choose $\mathbf{y}_3 = \mathbf{y}_1$ and O_3 the same as O_1 (Figure 4-3). Then, $\mathbf{y}_1 \cdot \mathbf{y}_3 = 1$ and $\alpha = 0$. If $\mathbf{x}_2 = \mathbf{x}_1$ then $\mathbf{x}_2 \cdot \mathbf{x}_1 = 1$ and $\beta = 0$, while if $\mathbf{x}_2 = -\mathbf{x}_1$ then $\mathbf{x}_2 \cdot \mathbf{x}_1 = -1$ and $\beta = \pi$. Therefore, the rotational part of the transformation between two coordinate systems with parallel x axes can be represented by

$$\hat{m} = twix(\gamma) \quad if \ \mathbf{x}_2 \cdot \mathbf{x}_1 = 1, \tag{4.20}$$

$$\hat{m} = twix(\gamma - \pi) XTOY XTOY \quad if \ \mathbf{x}_2 \cdot \mathbf{x}_1 = -1. \tag{4.21}$$

It can be seen that Equation 4.20 can be obtained from Equation 4.19 by substituting $\theta = \alpha = 0$, $\phi = \beta - \pi = 0 - \pi = -\pi$ and $\psi = \gamma - \pi$. Then

$$
\begin{aligned}
\hat{m} &= twix(\gamma - \pi) XTOY twix(\pi) XTOY \\
&= twix(\gamma),
\end{aligned}
$$

since
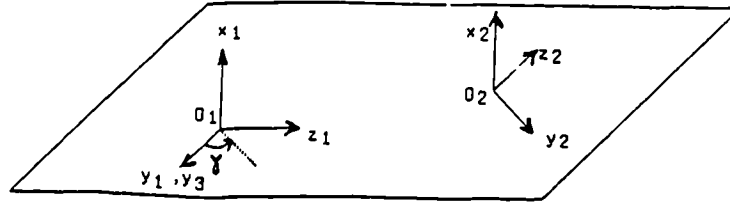
Figure 4–3: Transformation in case of parallel x axes

$$XTOYtwix(\pi)XTOY = twix(\pi).$$

Similarly, Equation 4.21 can be obtained from Equation 4.19 if we substitute $\theta = \alpha = 0$, $\phi = \beta - \pi = 0$ and $\psi = \gamma - \pi$. Therefore, from Equations 4.19 — 4.21 it follows that any transformation, which is the product of a rotation and a translation, between two coordinate systems can be represented by

$$m = twix(\psi)XTOYtwix(\phi)XTOYtwix(\theta)trans(\mathbf{d}), \qquad (4.22)$$

where d is a vector from the origin of $O_1 x_1 y_1 z_1$ to the origin of $O_2 x_2 y_2 z_2$. $\square$

In the case of perpendicular x axes, $\beta = \pi/2$, the rotational part of the transformation between the coordinate systems can be represented by

$$\hat{m} = twix(\psi)XTOYtwix(\phi) \quad if \ \mathbf{x_2} \cdot \mathbf{x_1} = 0. \qquad (4.23)$$

4.3 Spatial Relationships

4.3.1 Objects, Features and Spatial Relationships

The geometric reasoning system presented in this thesis is based on the notion of spatial relationships holding between features of objects. In this section, the definition of spatial relationships will be presented, as described in [Ambler and Popplestone 75].

Definition 4.3.1 *The* location of an object, *p*, *is defined to be the location that will transform the reference coordinate system, W, to a particular coordinate system embedded in the object. Therefore, the axes of an object are represented by Wp.*

Each object is composed of a number of geometric features. In order to be able to talk about spatial relationships between features of objects, we need to define the locations of the features of each object.

Definition 4.3.2 *The* location *f* of a feature *F* *of an object is defined to be the location that will transform the object coordinate system into the coordinate system of the feature.*

Polyhedral objects have three types of features: plane faces, edges and vertices. Coordinate axes are embedded in features according to the following rules: let F be a feature of object B and let the location p of B be equal to the identity element of locations, i.e. $p = I$. Then

1. F is a plane face: the axes represented by Wf have their origin lying in the plane and the x-axis of Wf points along the outward normal of the plane.

2. F is an edge: the axes represented by Wf have their origin somewhere on the edge and the x-axis of Wf points along the edge.
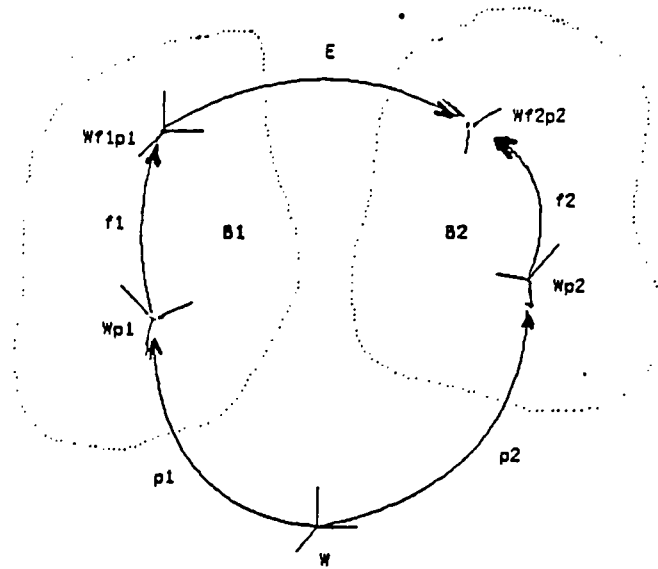
E is a spatial relationship holding between features F_1 and F_2 of objects B_1 and B_2 respectively. W is the reference coordinate system. p_1 and p_2 are the locations of objects B_1 and B_2. f_1 and f_2 are the locations of the features.

Figure 4–4: Spatial relationships between features of objects

3. F is a vertex: the origin of Wf coincides with the vertex.

When an object is at location p then the axes of the feature F of the object are represented by Wfp.

Definition 4.3.3 *A spatial relationship, E between the features F_1 of object B_1 and F_2 of object B_2 is a transformation expression from the coordinate system of feature F_1 to the coordinate system of F_2 [Popplestone, Ambler, and Bellos 80].*

Definitions 4.3.1 — 4.3.3 are illustrated in Figure 4–4. From the definitions it follows that

$$WEf_1p_1 = Wf_2p_2 \tag{4.24}$$

Since the matrices are non-singular, Equation 4.24 can be rewritten as

$$p_2p_1^{-1} = f_2^{-1}Ef_1 \tag{4.25}$$
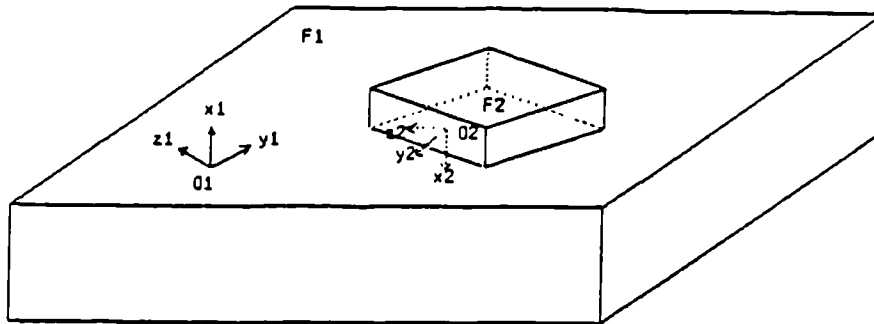
Figure 4-5: Face F_1 against face F_2

The relative location of two objects can thus be expressed in terms of a spatial relationship holding between the features of the objects.

As an example, consider two objects B_1 and B_2 with plane faces F_1 and F_2 respectively and let F_1 be "against" F_2, where the relationship "against" implies that the two faces are coplanar with opposed normals. The situation is depicted in Figure 4-5.

Let f_1 be the location of F_1, f_2 the location of F_2, p_1 the location of B_1 and p_2 the location of B_2. Then the coordinate system $O_1 x_1 y_1 z_1$ of F_1 is represented by $W f_1 p_1$ and the coordinate system $O_2 x_2 y_2 z_2$ of F_2 by $W f_2 p_2$. The "against" relationship is a transformation

$$E = \hat{E}trans(\mathbf{d})$$

between these two coordinate systems. It specifies that $\mathbf{x}_1 \cdot \mathbf{x}_2 = -1$ and that the origin O_2 lies on the $y_1 z_1$ plane. The rotational part of E is derived from Equation 4.21,

$$\hat{E} = twix(\psi)XTOY\,XTOY = XTOY\,XTOY\,twix(\theta) \tag{4.26}$$

where $\theta = -\psi$. Since O_2 lies on the $y_1 z_1$ plane, the vector **d** from O_1 to O_2 can be expressed as

$$\mathbf{d} = (0, y, z) \tag{4.27}$$

Therefore, from Equations 4.26 and 4.27,

$$E = XTOY\,XTOY\,twix(\theta)trans(0, y, z) \tag{4.28}$$

Since B_2 is free to rotate about its x axis, variable ψ of Equation 4.26 is a free variable and it corresponds to the rotational degree of freedom of B_2 allowed by the "against" relationship. Similarly, the variables y and z correspond to the translational degrees of freedom of B_2 relative to B_1.

It has to be noted that the given definition of a spatial relationship completely ignores the physical extent of features. In the given example, the two faces would still be "against" each other even if there was no physical contact.

4.3.2 Spatial relationships and Surfaces of the Configuration Space

Spatial relationships between features of objects can be seen as constraints on the relative locations of the objects. The set of relative locations which satisfy a spatial relationship lies on some surface of the configuration space. The dimension of the surface is equal to the degrees of freedom of relative motion of the objects implied by the relationship.

More specifically, let E be a spatial relationship between features with locations f_1 and f_2. Recall that Equation 4.25 gives the relative location of the two objects,

$$p_2 p_1^{-1} = f_2^{-1} E f_1.$$

The above equation can be regarded as the parametric representation of a surface of the configuration space consisting of all relative locations of the two objects which satisfy the spatial relationship E.

As an example, I will consider the case in which the configuration space is 3-dimensional, with two translational degrees of freedom and one rotational, and I will derive the form of the equation of the surface for the case of vertex to edge contact. Let the allowed motions be translation along the y and z axes and rotation about the x axis.

Consider the case depicted in Figure 4–6, where a vertex F_2 of moving object B_2 is "against" an edge F_1 of object B_1. Let f_1 be the location of F_1 and f_2 the location of F_2 as shown in the figure. Then we can find ρ, λ so that the "against vertex edge" relationship is described by the transformation[3]

$$E = twix(\rho)trans(0, \lambda, 0),$$

---

[3]This relationship will be referred to as the ROTYLIN relationship. Note also that, in this 2-dimensional case, the convention for the axes of the edge have not been followed.
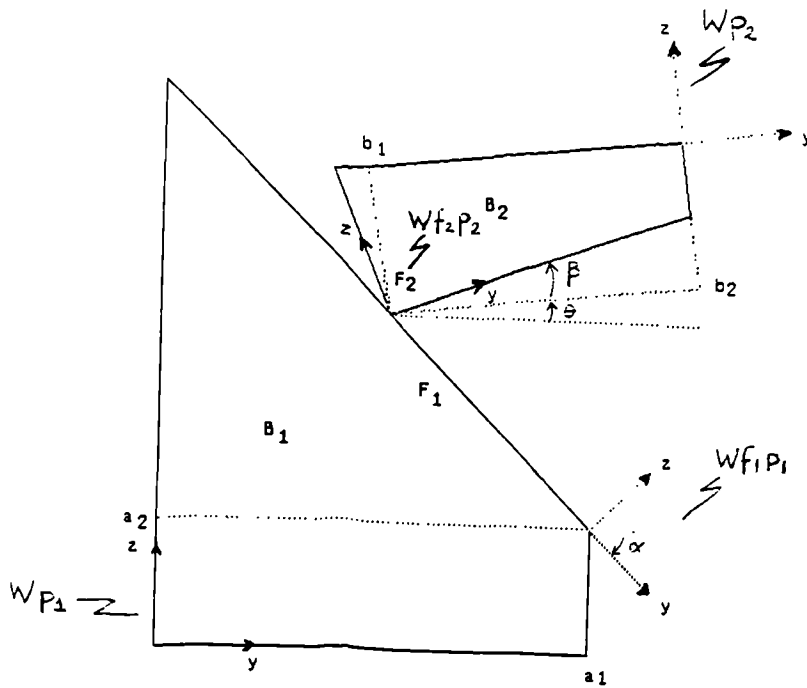
The vertex F_2 (location $f_2 = twix(\beta)trans(0, b_1, b_2)$) of B_2 (location p_2) is on the edge F_1 (location $f_1 = twix(\alpha)trans(0, a_1, a_2)$) of B_1 (location p_1).

Figure 4–6: A vertex against an edge in 2 dimensions

where ρ, λ are free variables.

The equation

$$p_2 p_1^{-1} = f_2^{-1} twix(\rho) trans(0, \lambda, 0) f_1 \tag{4.29}$$

is the parametric representation of the two dimensional surface defined by the set of locations which satisfy the relationship.

The explicit equation of the surface can be found by eliminating ρ and λ from the above equation:

Let

$$
\begin{aligned}
p_2 p_1^{-1} &= twix(\theta) trans(0, y, z) \\
f_1 &= twix(\alpha) trans(0, a_1, a_2) \\
f_2 &= twix(\beta) trans(0, b_1, b_2),
\end{aligned}
$$

where $\alpha, \beta, a_1, a_2, b_1$ and b_2 are constants (see Figure 4–6). Then Equation 4.29 becomes

$$twix(\theta)trans(0, y, z) =$$
$$twix(-\beta)trans(0, -b_1, -b_2)twix(\rho)trans(0, \lambda, 0)$$
$$twix(\alpha)trans(0, a_1, a_2). \tag{4.30}$$

Using the following property of transformations,

$$Rot(\mathbf{a}, \theta)trans(\mathbf{d}_1)Rot(\mathbf{b}, \theta)trans(\mathbf{d}_2) =$$
$$Rot(\mathbf{a}, \theta)Rot(\mathbf{b}, \theta)trans(\mathbf{d}_1 Rot(\mathbf{b}, \theta) + \mathbf{d}_2),$$

Equation 4.30 is rewritten as

$$twix(\theta)trans(0, y, z) =$$
$$twix(-\beta + \rho + \alpha)$$
$$trans((0, -b_1, -b_2)twix(\rho + \alpha) + (0, \lambda, 0)twix(\alpha) + (0, a_1, a_2)).$$

Carrying out the matrix multiplications in the second part of the above equation we obtain

$$twix(\theta)trans(0, y, z) =$$
$$twix(-\beta + \rho + \alpha)$$
$$trans(0, -b_1 \cos(\rho + \alpha) + b_2 \sin(\rho + \alpha) + \lambda \cos \alpha + a_1,$$
$$-b_1 \sin(\rho + \alpha) - b2 \cos(\rho + \alpha) + \lambda \sin \alpha + a_2).$$

Separating the rotational and translational parts of the above equation we get

$$\theta = -\beta + \rho + \alpha \tag{4.31}$$
$$y = -b_1 \cos(\rho + \alpha) + b_2 \sin(\rho + \alpha) + \lambda \cos \alpha + a_1 \tag{4.32}$$
$$z = -b_1 \sin(\rho + \alpha) - b_2 \cos(\rho + \alpha) + \lambda \sin \alpha + a_2. \tag{4.33}$$

From Equation 4.31, $\rho + \alpha = \theta + \beta$. Finally, eliminating λ from Equations 4.32 and 4.33 and substituting for $\rho + \alpha = \theta + \beta$, we get

$$y \sin \alpha - z \cos \alpha =$$
$$b_1 \sin(\theta - \beta - \alpha) + b_2 \cos(\theta - \beta - \alpha) + a_1 \sin \alpha - a_2 \cos \alpha$$

Therefore, for any relative location $p = p(\theta, y, z)$ which satisfies the spatial relationship E:

$$f(y, z, \theta) =$$
$$z \cos \alpha - y \sin \alpha + b_1 \sin(\theta - \beta - \alpha)$$
$$+b_2 \cos(\theta - \beta - \alpha) + a_1 \sin \alpha - a_2 \cos \alpha = 0 \qquad (4.34)$$

Equation 4.34 is the equation of the surface specified by Equation 4.29. [4]

---

[4]Substituting for $y = x$, $z = y$, $\alpha = \phi_j - 90$, $\beta = 0$, $a_1 = \parallel b_j \parallel \cos(\gamma_j)$, $a_2 = \parallel b_j \parallel \sin(\gamma_j)$, $b_1 = \parallel a_j \parallel \cos(\eta_i)$ and $b_2 = \parallel a_j \parallel \sin(\eta_i)$ into Equation 4.34 we get

$$f(x, y, \theta) = x \cos(\phi_j) + y \sin(\phi_j) - \parallel a_i \parallel \cos(\theta + \eta_i - \phi_j) - \parallel b_i \parallel \cos(\phi_j - \gamma_j),$$

which is the formula given in [Brooks and Lozano-Perez 83] for a Type B constraint.

4.3.3 Taxonomy of Spatial Relationships

Five degree of freedom relationships

In the work presented in this thesis, we are interested in spatial relationships among polyhedral objects which imply contact. If two objects are in contact, then the maximum number of translational degrees of freedom in the relative location of the objects is two, while the maximum number of rotational degrees of freedom is three. There are two types of 5 d.o.f. relationships among polyhedral objects which imply contact:

1. A plane face of an object is against a vertex of another object. Such a relationship will be denoted by AGPV [5], which stands for "against plane vertex". The inverse of this relationship is AGVP, "against vertex plane".

2. An edge of an object intersects an edge of another object Such a relationship will be denoted by AGEE, "against edge edge".

Let us consider first the AGPV relationship. Let F_1 be a plane with a coordinate system $O_1 x_1 y_1 z_1$ and F_2 a vertex with coordinate system $O_2 x_2 y_2 z_2$ and let the vertex be on the plane. The situation is shown in Figure 4–7. Let α, β, γ, O_3 be as defined in Section 4.2.3. In addition, let $a = |\vec{O_2 O_3}|$ and $b = |\vec{O_1 O_3}|$. From Equation 4.22, the rotational part of the AGPV relationship can be expressed as

$$\hat{E} = twix(\gamma - \pi)XTOYtwix(\beta - \pi)XTOYtwix(\alpha), \qquad (4.35)$$

where α, β and γ are free variables. The vector d from O_1 to O_2 can be expressed as

$$d = (0, -a, b)twix(\alpha) = (0, y, z), \qquad (4.36)$$

---

[5]The conventions for naming relationships are based on RAPT
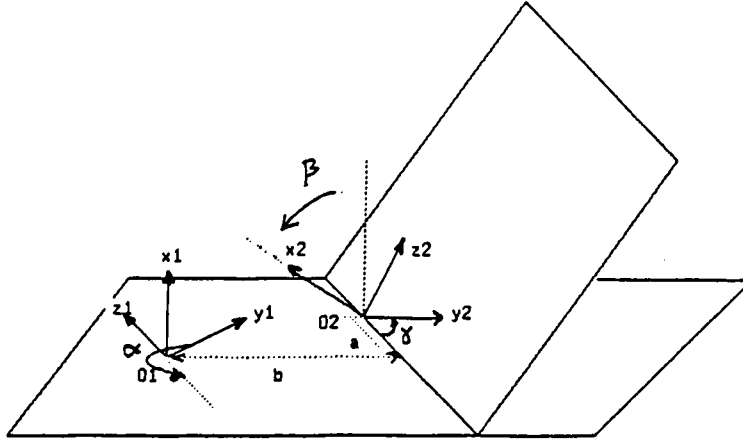
The vertex with coordinate system $O_2x_2y_2z_2$ lies on the plane face with coordinate system $O_1x_1y_1z_1$. $AGPV$ is the transformation from $O_1x_1y_1z_1$ to $O_2x_2y_2z_2$ and $AGVP$ is its inverse:

$AGPV = twix(\gamma - \pi)XTOYtwix(\beta - \pi)XTOYtwix(\alpha)trans((0, -a, b)twix(\alpha))$

$AGVP = twix(\pi - \alpha)XTOYtwix(\pi - \beta)XTOYtwix(-\gamma)$
$\qquad trans((a, 0, b)twix(\pi - \beta)XTOYtwix(-\gamma)).$

Figure 4-7: A plane against a vertex

where y and z are free variables. From Equations 4.35 — 4.36,

$$AGPV = twix(\gamma - \pi)XTOYtwix(\beta - \pi)XTOYtwix(\alpha)$$
$$trans(0, y, z), \qquad (4.37)$$

where $\alpha, \beta, \gamma, y, z$ are free variables.

It can be seen that

$$AGVP = AGPV^{-1} = twix(\pi - \alpha)XTOYtwix(\pi - \beta)XTOYtwix(-\gamma)$$
$$trans((a, 0, b)twix(\pi - \beta)XTOYtwix(-\gamma)). \qquad (4.38)$$

The case of two edges intersecting (AGEE) is depicted in Figure 4-8. In this case, let O_4 be the foot of the perpendicular from O_2 to the YZ plane of the the first edge and $a = |\overrightarrow{O_2O_4}|$ and $b = |\overrightarrow{O_1O_4}|$. Then,

$$AGEE = twix(\gamma - \pi)XTOYtwix(\beta - \pi)XTOYtwix(\alpha)$$
$$trans((a, 0, b)twix(\alpha)), \qquad (4.39)$$
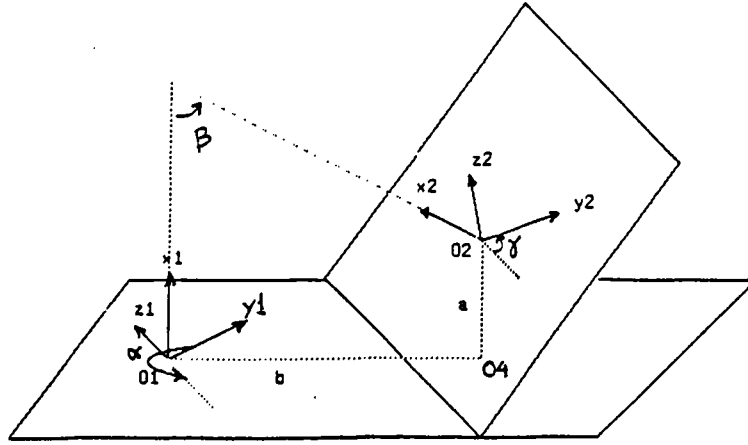
The two edges with coordinate systems $O_1x_1y_1z_1$ and $O_2x_2y_2z_2$ intersect. $AGEE$ is the transformation from $O_1x_1y_1z_1$ to $O_2x_2y_2z_2$:

$$AGEE = twix(\gamma - \pi)XTOYtwix(\beta - \pi)XTOYtwix(\alpha)trans((a, 0, b)twix(\alpha)).$$

Figure 4–8: An edge against an edge

where $\alpha, \beta, \gamma, a, b$ are free variables[6].

Types of relationships

All relationships with degrees of freedom less than five can be described as conjunctions of AGPV's, AGVP's and AGEE's. In order to find the set of all possible types of relationships with four degrees of freedom, we have to consider the combinations : [AGPV, AGPV], [AGPV, AGVP], [AGPV, AGEE], [AGEE, AGEE] and to find equivalent relationships for all possible cases. The method that has been followed will be presented in detail in the next section. Here, we just note that the type of equivalent relationship depends on the relative position of the features of objects. For example, consider the case [AGPV, AGPV]. If the same plane face is involved in the two relationships, then the edge connecting

---

[6]Note that in Equation 4.39 if $\beta = 0$ then the edges are parallel and do not intersect. This case will be inlcluded in the AGEE relationship.

the two vertices must lie on that plane too and, thus , the equivalent relationship is "against plane to edge", denoted AGPE. On the other hand, if the same vertex is involved in the two relationships, then the vertex is constrained to lie on the intersection of the two planes and, thus, the equivalent relationship is "against edge to vertex", denoted AGEV.

The set of all types of spatial relationships can be derived by considering the different cases arising from conjunctions of a 5 d.o.f. relationship with an i d.o.f. relationship, where $i = 5, 4, \ldots, 1$. The method will be presented in the rest of this chapter.

Degrees of freedom of relationships

Consider a spatial relationship E holding between features with locations f_1 and f_2,

$$p_2 = f_2^{-1} E f_1 p_1.$$

According to Theorem 4.2, any spatial relationship E, since it is defined to be a transformation expression between two coordinate systems, can be represented by

$$E = twix(\psi) XTOY \, twix(\phi) XTOY \, twix(\theta) trans(x, y, z), \tag{4.40}$$

where ψ, ϕ, θ, x, y, and z are expressions in the free variables of the equation. In the above equation ψ, ϕ and θ specify the relative orientation of the two objects and x, y and z the relative translation.

In the case of an *AGPV* relationship, for example, we have derived that

$$E = AGPV = twix(\psi) XTOY \, twix(\phi) XTOY \, twix(\theta) trans(0, y, z).$$

Let the vertex belong to the moving object B_2. In this case ψ, ϕ, θ, y, and z are free variables. The free variables θ, ϕ and ψ correspond to the three rotational degrees of freedom implied by the relationship and the free variables y and z

correspond to the two translational degrees of freedom of B_2: B_2 is permitted to rotate about the x, y and z axes and to translate along the y and z axes of $W f_1 p_1$.

In the case of an *AGVP* relationship, Equation 4.38 can be rewritten as

$$E = AGVP = twix(\psi) XTOY trans(x, 0, z) twix(\phi) XTOY twix(\theta).$$

Let the plane face belong to the moving object B_2. The *AGVP* relationship can be interpreted as: B_2 is permitted to rotate about the three axes of $W f_1 p_1$ and to translate along the x and z axes of $W twix(\phi) XTOY twix(\theta) f_1 p_1$. The above equation can also be rewritten as

$$E = AGVP = trans(0, y', z') twix(\psi) XTOY twix(\phi) XTOY twix(\theta),$$

where ψ, ϕ, θ, y' and z' are free. Expressed in this form, the relationship can be interpreted as a relationship according to which B_2 is allowed to rotate about the axes of $W f_1 p_1$ and to translate along the y and z axes of $W f_2 p_2$.

In general, if a spatial relationship is expressed in the form of Equation 4.40,

$$E = twix(\psi) XTOY twix(\phi) XTOY twix(\theta) trans(x, y, z),$$

then the free variables in the expression would correspond to the degrees of freedom of motion allowed by the relationship. Whether ψ, ϕ, θ, x, y or z are free variables or expressions depends on the locations of the coordinate axes of the features between which the relationship E holds. The number of degrees of freedom of a relationship is equal to number of free variables in the above expression and, thus, it is equal to the dimension of the surface in configuration space represented by the equation $p_2 p_1 = f_2^{-1} E f_1$. In the next sections a method is presented for solving location equations in order to derive a relationship equivalent to a pair of relationships. The coordinate axes of the features between which the equivalent relationship holds are chosen in such a way, if possible, that ψ, ϕ, θ, x, y or z are either constants or free variables. If this is the case, then the d.o.f of the relationship are independent.
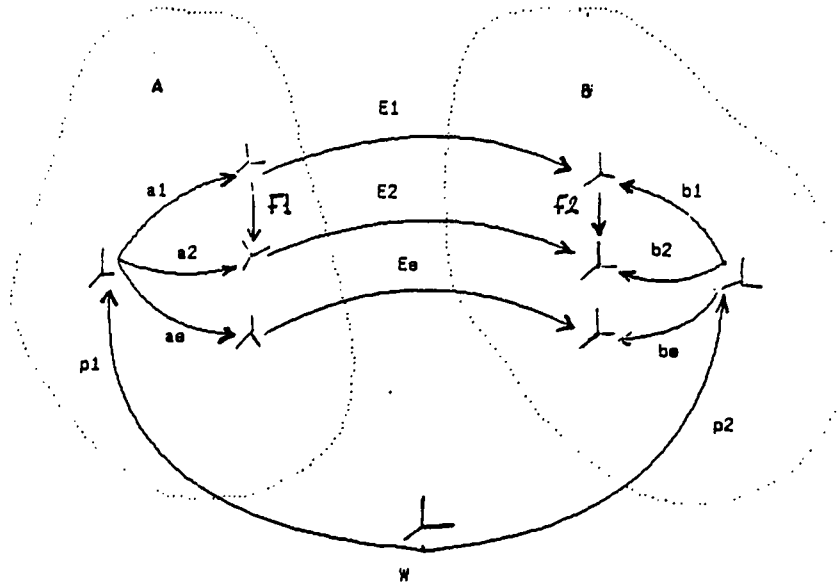
E_i is a spatial relationship between feature with location a_i of object A and feature with location b_i of object B. The relationship E_c is equivalent to the conjunction of E_1 and E_2.

Figure 4–9: A 2-cycle of relationships

4.4 Algebraic Method for Replacing a 2-Cycle of Relationships

4.4.1 General form of the equations

Let E_1 be a relationship between features F_{A1} of object A and F_{B1} of object B and E_2 a relationship between features F_{A2} of A and F_{B2} of B. Let the locations of A, B, F_{A1}, F_{A2}, F_{B1} and F_{B2} be p_1, p_2, a_1, a_2, b_1 and b_2 respectively. The situation is depicted in Figure 4–9.

From Equation 4.25,

$$p_2 = b_1^{-1} E_1 a_1 p_1, \tag{4.41}$$

$$p_2 = b_2^{-1} E_2 a_2 p_1. \tag{4.42}$$

Combining Equation 4.41 and Equation 4.42, we obtain

$$b_1 b_2^{-1} E_2 a_2 a_1^{-1} E_1^{-1} = I. \tag{4.43}$$

Since the features are fixed on the bodies, the relationship between two features on the same body is a fixed transformation. Let

$$F_1 = a_2 a_1^{-1} = \hat{F}_1 trans(\mathbf{u}_1), \tag{4.44}$$

$$F_2 = b_2 b_1^{-1} = \hat{F}_2 trans(\mathbf{u}_2), \tag{4.45}$$

$$E_1 = \hat{E}_1 trans(\mathbf{d}_1), \tag{4.46}$$

$$E_2 = \hat{E}_2 trans(\mathbf{d}2). \tag{4.47}$$

Substituting Equations 4.44 – 4.47 in Equation 4.43, we get

$$(\hat{F}_2 trans(\mathbf{u}_2))^{-1}(\hat{E}_2 trans(\mathbf{d}_2))(\hat{F}_1 trans(\mathbf{u}_1))(\hat{E}_1 trans(\mathbf{d}_1))^{-1} = I \tag{4.48}$$

Using the following property of transformations,

$$Rot(\mathbf{a}, \theta) trans(\mathbf{d}_1) Rot(\mathbf{b}, \theta) trans(\mathbf{d}_2) =$$
$$Rot(\mathbf{a}, \theta) Rot(\mathbf{b}, \theta) trans(\mathbf{d}_1 Rot(\mathbf{b}, \theta) + \mathbf{d}_2),$$

Equation 4.48 can be rewritten as

$$\hat{E}_2 \hat{F}_1 trans(\mathbf{d}_2 \hat{F}_1 + \mathbf{u}_1) = \hat{F}_2 \hat{E}_1 trans(\mathbf{u}_2 \hat{E}_1 + \mathbf{d}_1)$$

Finally, separating the rotational and the translational parts of the above equation we get the following two equations:

$$Rotation \quad : \quad \hat{F}_2^{-1} \hat{E}_2 \hat{F}_1 \hat{E}_1^{-1} = I \tag{4.49}$$

$$Translation \quad : \quad \mathbf{d}_2 \hat{F}_1 - \mathbf{d}_1 = \mathbf{u}_2 \hat{E}_1 - \mathbf{u}_1 \tag{4.50}$$

The relationships E_1 and E_2 in general include variables, each variable corresponding to a degree of freedom of motion implied by the relationship. Solving Equations 4.49 and 4.50 we obtain the values for some of the free variables. If we then substitute these values back into one of the original Equations 4.41 or 4.42, we obtain a new equation of the form

$$p_2 = b_e^{-1} E_e a_e p_1, \tag{4.51}$$

where E_e is a relationship equivalent to the conjunction of E_1 and E_2 and a_e and b_e are the locations of the new features. In the next pages we will discuss the solution of Equations 4.49 and 4.50 and methods for finding the equivalent relationship E_e, and the new features, a_e and b_e.

4.4.2 Degrees of Freedom

Let n_{Ri} and n_{Li} denote the number of the rotational and the translational d.o.f. respectively of a relationship E_i. The following observations can be made concerning the relation between the degrees of freedom of the initial relationships and the equivalent relationship:

1. $n_{Ri} \leq 3$, $n_{Li} \leq 3$ and $n_{Ri} + n_{Li} \leq 6$

2. $n_{R3} \leq min(n_{R1}, n_{R2})$ and $n_{L3} \leq min(n_{L1}, n_{L2})$

where E_3 is a relationship equivalent to the conjunction of E_1 and E_2. In the rest of this section it will be assumed that we have chosen E_1 and E_2 so that

$$n_{R1} \leq n_{R2}.$$

1. Rotational Equation

The general form of the rotational equation is given by Equation 4.49. This equation can always be reduced to the form [Ambler and Popplestone 75]

$$twix(l_1)a_1 twix(l_2)...a_{n-1}twix(l_n) = c, \qquad (4.52)$$

where a_i and c are all constant matrices which cannot be expressed in the form $twix(\alpha)$ or $XTOYXTOYtwix(\alpha)$, and l_i are linear expressions. The solution of this equation for $n \leq 3$ is discussed in [Ambler and Popplestone 75]. The results are summarised below:

1. $n = 1$

$$twix(\theta) = a.$$

Soluble iff $a_{11} = 1$. Choose θ so that

$$e^{i\theta} = a_{22} + ia_{23}. \qquad (4.53)$$

2. $n = 2$

$$twix(\theta)\,btwix(\phi) = a \tag{4.54}$$

Soluble iff $a_{11} = b_{11}$. Choose θ so that

$$e^{i\theta} = (a_{12} + ia_{13})/(b_{12} + ib_{13}). \tag{4.55}$$

3. $n = 3$

$$twix(\psi)\,ctwix(\phi)\,btwix(\theta) = a.$$

Soluble iff $(a_{11} - b_{11}c_{11})^2 \leq (1 - b_{11}^2)(1 - c_{11}^2)$. Choose ϕ so that

$$(c_{12}b_{21} + c_{13}b_{31})\cos\phi + (c_{12}b_{31} - c_{13}b_{21})\sin\phi = c_{11} - a_{11}b_{11} \tag{4.56}$$

which has two solutions. Choose θ, ψ using Equation 4.54 for each value of ϕ.

When $n > 3$ then Equation 4.52 is underdetermined, and therefore the solutions are expressed in some of the variables. However, we can make inferences about the dependency between the degrees of freedom of the initial and the equivalent relationships.

The number of rotational degrees of freedom of the equivalent relationship depends on the constant matrices F_1 and F_2. Of particular interest is the case when F_1, F_2 can be expressed in the form $twix(\theta)$ or $XTOYXTOYtwix(\theta)$. This condition occurs if the x axes of the features of body A are parallel, or the x axes of the features of body B are parallel.

As an example consider the following case:

$$\hat{E}_1 = twix(\theta), \tag{4.57}$$

$$\hat{E}_2 = twix(\gamma)XTOYtwix(\beta)XTOYtwix(\alpha). \tag{4.58}$$

Then Equation 4.49 becomes

$$\hat{F}_2^{-1}twix(\gamma)XTOYtwix(\beta)XTOYtwix(\alpha)\hat{F}_1twix(-\theta) = I. \tag{4.59}$$

Let us consider the case when the x axes of A are parallel. Then $[F_1]_{11} = \pm 1$ and $\hat{F}_1 = twix(\eta)$ or $\hat{F}_1 = twix(\eta)XTOYXTOY$, where η is a free variable. Equation 4.59 becomes

$$\hat{F}_2^{-1} twix(\gamma)XTOY\,twix(\beta)XTOY\,twix(\xi) = I,$$

where $\xi = \alpha + \eta \pm \theta$. Using Equations 4.56 and 4.54 we obtain values for β, γ and ξ and, thus,

$$\theta = \alpha + constant.$$

Therefore, we can express θ as a linear expression in α, but we cannot find a value of θ using the rotational equation. The same applies for the case of $[F_2]_{11} = \pm 1$.

In general, the remaining rotational variables will appear in the translational equation. Using Equations 4.53 – 4.56 and the above observations Table 4–1 has been constructed which summarises the relation between the rotational degrees of freedom in the solution and the initial relationships. In this table only the rotational equation is considered. As it will be shown, the value of rotational variable(s) can sometimes be obtained through the solution of the translational equation. Therefore, Table 4–1 should be considered in conjunction with the translational equation.

2. Translational Equation

The translational equation 4.50 will be solved by transforming $\mathbf{d}_1$ and $\mathbf{d}_2$ into canonical form as defined below:

$$\mathbf{d}_1 = \delta_1 \hat{H}_1$$
$$\mathbf{d}_2 = \delta_2 \hat{H}_2,$$

where δ_i is of the form $(x_1, 0, 0)$ or $(x_1, y_1, 0)$. As an example, consider the case of the AGPV relationship (Equation 4.37):

$$\mathbf{d}_1 = (0, y, z).$$

d.o.f. in equations		d.o.f. in solutions		
n_{R1}	n_{R2}	n_{R3}		
		General Case	$[F_1]_{11}$ or $[F_2]_{11} = \pm 1$	$[F_1]_{11}$ and $[F_2]_{11} = \pm 1$
0	1	0	0	0
0	2	0	0	x
0	3	0*	0*	0
1	1	0	1_L	1
1	2	0*	1_L	x
1	3	1*	1_L*	1_L
2	2	1*	1_L*	2_L
2	3	2*	2* $(1_L + 1$*$)$	2_L*
3	3	3*	3* $(1_L$* $+ 2$*$)$	3* $(2_L$* $+ 1$*$)$

x: insoluble

* : 2 solutions (see item 3, 137)

n_L : n variables can be expressed as a linear form, e.g. 2*$(1_L + 1)$

means that there are two free variables in the solution, e.g. θ and

ϕ, and one of them can be expressed as a linear form, e.g. $\theta =$

$\alpha + constant$

Table 4-1: Rotational d.o.f. of equivalent relationship

The above equation can be rewritten as

$$d_1 = (y, z, 0)\, twix(\pi/2)\, XTOY.$$

Therefore, in this case

$$\delta_1 = (x_1, y_1, 0),$$

where $x_1 = y$ and $y_1 = z$ and

$$\hat{H}_1 = twix(\pi/2)\, XTOY.$$

In the case of the AGVP relationship (Equation 4.38),

$$d_1 = (a, 0, b)\, twix(\pi - \beta)\, XTOY\, twix(-\gamma).$$

By setting $x_1 = a$, $y_1 = b$, $\phi = 3\pi/2 - \beta$ and $\theta = -\gamma$, d_1 is transformed into canonical form:

$$
\begin{aligned}
d_1 &= (x_1, 0, y_1)\, twix(-\pi/2)\, twix(\phi)\, XTOY\, twix(\theta) \\
&= (x_1, y_1, 0)\, twix(\phi)\, XTOY\, twix(\theta),
\end{aligned}
$$

and

$$\delta_1 = (x_1, y_1, 0)$$

$$\hat{H}_1 = twix(\phi)\, XTOY\, twix(\theta).$$

⤝ It can be observed that $\hat{H}_i$ will sometimes include some of the rotational variables of the relationship, as in the case of AGVP.

Having transformed d_1 and d_2 into the above defined canonical form, we can rewrite Equation 4.50 as

$$\delta_2\hat{Q} - \delta_1 = (u_2\hat{E}_1 - u_1)\hat{H}_1^{-1}, \tag{4.60}$$

where $\hat{Q} = \hat{H}_2 \hat{F}_1 \hat{H}_1^{-1}$.

Equation 4.60 can be solved as a system of linear equations if both $\hat{Q}$ and $(u_2 \hat{E}_1 - u_1) \hat{H}_1^{-1}$ are constants. The following cases can be distinguished:

1. If $\hat{Q}$ is a constant matrix then

    (a) if $\hat{E}_1$ is also a constant matrix then Equation 4.60 can be solved as a system of linear equations in the translational variables.

    (b) If $[\hat{E}_1]_{11} = \pm 1$ and $u_2 = (l_{22}, 0, 0)$, then $(u_2 \hat{E}_1 - u_1)$ is constant and, therefore, Equation 4.60 can again be solved as a system of linear equations.

    (c) Otherwise Equation 4.60 will include the rotational variables of E_1 for which no values could be obtained by considering the rotational equation (4.49). In this case, the condition of solubility of Equation 4.60 can be used to obtain values for the remaining rotational variables of E_1.

2. If $\hat{Q}$ is not constant then Equation 4.60 cannot be solved as a system of linear equations. In this case the equivalent relationship will be found geometrically, as it will be explained in Section 4.5.

Table 4–2 summarises the relation between the translational degrees of freedom of the original and the equivalent relationships, in the case of linear equations.

d.o.f. in equations		d.o.f. in solution		
		General Case	Special Cases	
n_{L_1}	n_{L_2}	n_{L_3}	n_{L_3}	Condition
0	1			
0	2			
0	3	0	-	-
1	0			
2	0			
3	0			
1	1	0	1	$Q_{11} = \pm 1$
1	2	0	1	$Q_{31} = 0$
2	1	0	1	$Q_{13} = 0$
2	2	1	2	$Q_{33} = \pm 1$

Table 4–2: Translational d.o.f. of equivalent relationship in case of linear equations

4.4.3   Procedure for the solution of the equations

In Section 4.4.1 the following two equations were derived (4.49 and 4.50):

$$Rotation \quad : \quad \hat{F_2}^{-1}\hat{E_2}\hat{F_1}\hat{E_1}^{-1} = I$$

$$Translation \quad : \quad \mathbf{d_2}\hat{F_1} - \mathbf{d_1} = \mathbf{u_2}\hat{E_1} - \mathbf{u_1}.$$

In this section the method which has been followed for the solution of these equations will be outlined.

Let us consider the case that E_2 is the AGPV relationship (Equation 4.37):

$$E_2 = AGPV = twix(\gamma)XTOY\,twix(\beta)XTOY\,twix(\alpha)trans(0, y_2, z_2).$$

In general E_1 will be of the form

$$E_1 = \hat{E_1}trans(x_1, y_1, z_1).$$

Let

$$F_1 = \begin{pmatrix} f_{11} & f_{12} & f_{13} & 0 \\ f_{21} & f_{22} & f_{23} & 0 \\ f_{31} & f_{32} & f_{33} & 0 \\ l_{11} & l_{12} & l_{13} & 1 \end{pmatrix}, \tag{4.61}$$

$$\mathbf{u_2} = (l_{21}, l_{22}, l_{23}), \tag{4.62}$$

and

$$\mathbf{u_1} = (l_{11}, l_{12}, l_{13}). \tag{4.63}$$

Substituting Equations 4.61 – 4.63 into Equations 4.49 and 4.50 we obtain:

$$Rot : \hat{F_2}twix(\gamma)XTOY\,twix(\beta)XTOY\,twix(\alpha)\hat{F_1}\hat{E_1}^{-1} = I \tag{4.64}$$

$$Tran \quad : \quad (0, y_2, z_2) \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix} - (x_1, y_1, z_1) =$$

$$(l_{21}, l_{22}, l_{23}) \hat{E}_1 - (l_{11}, l_{12}, l_{13}). \tag{4.65}$$

Let

$$(A_1, A_2, A_3) = (l_{21}, l_{22}, l_{23}) \hat{E}_1 - (l_{11}, l_{12}, l_{13}).$$

In solving Equations 4.64 and 4.65, we try to find the values of the free variables of E_1 and to obtain the equivalent relationship by substituting these values into E_1.

1. First we consider the rotational equation:

   From Table 4–1 follows that in this case we can not use Equation 4.64 to obtain any values for the rotational variables of E_1. If $n_{R1} + n_{R2} \leq 3$ then the rotational equation could be used in order to determine the rotational variables of E_1.

2. Translation Equation:

   Only the case when $d_1 = (x_1, y_1, z_1)$ is independent of the rotational variables of E_1 will be considered. Let

   $$k = (A_1, A_2, A_3).$$

   The following cases can be considered according to the number of translational degrees of freedom n_{L1} of E_1:

   (a) $n_{L1} = 0$

   The condition for solubility of Equation 4.65 is

   $$f_{11} A_1 + f_{12} A_2 + f_{13} A_3 = 0. \tag{4.66}$$

It can be seen from Equation 4.65 that k is a constant if (i) $\hat{E}_1$ is constant; (ii) $[E_1]_{11} = \pm 1$ and $l_{22} = l_{23} = 0$ or (iii) $l_{21} = l_{22} = l_{23} = 0$. In the above cases, Equation 4.66 will determine the conditions under which E_1 and E_2 are consistent. Otherwise, k will include some of the rotational variables of E_1 and Equation 4.66 can be solved to obtain values for these variables.

(b) $n_{L1} = 1$

In this case $\mathbf{d}_1 = (x_1, 0, 0)$.

$D = 0$: We examine first the case that the determinant of Equation 4.65 is zero, which means that the translational variable x_1 is free. The condition for solution is again

$$f_{11}A_1 + f_{12}A_2 + f_{13}A_3 = 0$$

We follow the same procedure as in case (a) to decide when k is constant. If k is not a constant then the above equation can be used to find the values of some of the rotational variables.

$D \neq 0$: In this case, we can not determine the values of any of the rotational variables of E_1. If k is constant then the solution will not depend on the rotational variables. Otherwise the translational variable of E_1 will depend on the rotational variables of E_1.

(c) $n_{L2} = 2$

In this case $\mathbf{d}_1 = (x_1, y_1, 0)$. First we observe that we can not determine the values of both x_1 and y_1. Therefore, in all cases, the equivalent relationship will have a translational degree of freedom. Let us choose y_1 to be the free variable.

$D = 0$: We examine the case when the determinant of Equation 4.65 is zero, in which case x_1 will also be free. The condition for the solubility of the equation is given again by

$$f_{11}A_1 + f_{12}A_2 + f_{13}A_3 = 0$$

As in the previous cases this equation can be used to find values for the rotational variables if k is not constant.

$D \neq 0$: In this case a value for x_1 can be found. As in case (b), x_1 may or may not depend on the rotational variables according to whether **k** is constant or not.

Following the above procedure, we can determine the conditions under which we can obtain values for all, some or none of the rotational and translational variables of E_1 and we can find the values of these variables in terms of F_1 and F_2. These values are then substituted back into the original equation describing the relative location of the two objects in terms of E_1 (Equation 4.41)

$$p_2 = b_1^{-1} E_1 a_1 p_1.$$

The equivalent relationship and the location of the features between which the equivalent relationship holds are found by transforming the equation in a suitable way. For example, even if a solution for a translational variable of E_1 has been obtained in terms of the rotational variables of E_1, the equation can sometimes be transformed so that the translational variables of the equivalent relationship will be independent of the rotational variables.

4.4.4 Example: FITS -AGPV

The procedure described in Section 4.4.3 will be applied for the case of FITS-AGPV, where FITS denotes a relationship with one rotational degree of freedom about the x axis and one translational degree of freedom along the x axis. A list of spatial relationships and their algebraic form can be found in Table A–13 in Appendix A.

$$E_1 = FITS = twix(\theta)trans(x, 0, 0). \tag{4.67}$$

$$E_2 = AGPV = twix(\gamma)XTOY\,twix(\beta)XTOY\,twix(\alpha)trans(0, y, z). \tag{4.68}$$

Substituting for E_1 into Equation 4.41 we get

$$p_2 = b_1^{-1}twix(\theta)trans(x, 0, 0)a_1p_1. \tag{4.69}$$

Substituting for E_2 into Equation 4.42 we get

$$p_2 = b_2^{-1}twix(\gamma)XTOY\,twix(\beta)XTOY\,twix(\alpha)trans(0, y, z)a_2p_1. \tag{4.70}$$

The translation and rotation equations (Equations 4.49 and 4.50) become

$$Rot \; : \; \hat{F}_2^{-1}twix(\gamma)XTOY\,twix(\beta)XTOY$$
$$twix(\alpha)\hat{F}_1twix(-\theta) = I \tag{4.71}$$

$$Trans : (0, y, z)\hat{F}_1 - (x, 0, 0) = u_2twix(\theta) - u_1, \tag{4.72}$$

where

$$a_2a_1^{-1} = \hat{F}_1trans(u_1)$$

and

$$b_2b_1^{-1} = \hat{F}_2trans(u_2).$$

Equations 4.71 and 4.72 will be used in order to determine, if possible, the values of θ and x.

1. Rotational equation

   From Table 4–1, we deduce that we cannot use Equation 4.71 to obtain a value for θ. Therefore, for the moment, θ is free.

2. Translational equation

   Since

   $$(0,y,z)\hat{F_1} - (x,0,0) = (x,y,z)\begin{pmatrix} -1 & 0 & 0 \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix},$$

   Equation 4.72 can be rewritten as

   $$(x,y,z)\begin{pmatrix} -1 & 0 & 0 \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix} = \tag{4.73}$$

   $$(l_{21} - l_{11}, l_{22}\cos\theta - l_{23}\sin\theta - l_{12}, l_{22}\sin\theta + l_{23}\cos\theta - l_{13}).$$

   Let

   $$(A_1, A_2, A_3) =$$

   $$(l_{21} - l_{11}, l_{22}\cos\theta - l_{23}\sin\theta - l_{12}, l_{22}\sin\theta + l_{23}\cos\theta - l_{13}).$$

   Equation 4.73 can be solved as a system of linear equations in x, y and z. The determinant of the system is

   $$D = -\begin{vmatrix} f_{22} & f_{23} \\ f_{32} & f_{33} \end{vmatrix}.$$

   F_1 is an orthonormal matrix and therefore

   $$f_{11} = \begin{vmatrix} f_{22} & f_{23} \\ f_{32} & f_{33} \end{vmatrix}.$$

Therefore,

$$D = -f_{11}.$$

Using Cramer's rule,

$$x = D_1/D, \tag{4.74}$$

where

$$D_1 = f_{11}A_1 + f_{12}A_2 + f_{13}A_3. \tag{4.75}$$

We will examine first various special cases and then the general case.

(1) First we examine the case when the determinant of Equation 4.73 is zero, $D = 0$, which implies that $\boxed{f_{11} = 0.}$ In this case x is free. The solubility condition is $D_1 = 0$. From Equation 4.75,

$$Solubility\ Condition:\ f_{12}A_2 + f_{13}A_3 = 0. \tag{4.76}$$

Equation 4.76 will be used to find the value of θ. The following cases can be distinguished:

(1.1) $\boxed{l_{22} = l_{23} = 0}$

In this case

$$D_1 = -f_{12}l_{12} - f_{13}l_{13}.$$

Since D_1 is independent of θ, we can not use the solubility condition to determine the value of θ and, thus, θ remains a free variable. The solubility condition (Equation 4.76) becomes:

$$f_{12}l_{12} + f_{13}l_{13} = 0.$$

Therefore:

$$\left\{ \begin{array}{l} If \ f_{11} = 0 \ and \ l_{22} = l_{23} = 0 \ then : \\[1em] E_e = FITS \\[1em] a_e = a_1 \\[1em] b_e = b_1 \\[1em] Condition : \ f_{12}l_{12} + f_{13}l_{13} = 0 \end{array} \right.$$

(1.2) $\boxed{l_{22} \neq 0 \ and \ l_{23} \neq 0}$

In this case the solubility condition can be used to fix θ. Equation 4.76 can be rewritten as

$$(f_{12}l_{22} + f_{13}l_{23}) \cos \theta + (f_{13}l_{22} - f_{12}l_{23}) \sin \theta =$$
$$f_{12}l_{12} + f_{13}l_{13} \tag{4.77}$$

Equation 4.77 has either one or two solutions. The two cases are examined separately below:

(1.2.1) $\boxed{f_{12}l_{12} + f_{13}l_{13} = \sqrt{l_{22}^2 + l_{23}^2}}$

By setting

$$e^{i\alpha} = f_{12} + if_{13}, \ e^{i\beta} = (l_{22} + il_{23})/(\sqrt{l_{22}^2 + l_{23}^2}),$$

Equation 4.77 can be rewritten as

$$\cos(\alpha - \beta - \theta) = 1 \Rightarrow \theta = \alpha - \beta.$$

Substituting the value of θ into Equation 4.69,

$$\begin{array}{rcl} p_2 & = & b_1^{-1}twix(\alpha - \beta)trans(x,0,0)a_1p_1 \\[0.8em] & = & b_1^{-1}twix(-\beta)trans(x,0,0)twix(\alpha)a_1p_1 \\[0.8em] & = & b_e^{-1}E_e a_e p_1. \end{array}$$

Let *LIN* denote a spatial relationship with one translational degree of freedom (see Table A–13),

$$LIN = trans(x,0,0).$$

We can conclude that :

$$\begin{cases} \textit{If } f_{11} = 0 \textit{ and } f_{12}l_{12} + f_{13}l_{13} = \sqrt{l_{22}^2 + l_{23}^2} \textit{ then :} \\ E_e = LIN \\ a_e = twix(\alpha)a_1 \\ b_e = twix(\beta)b_1 \end{cases}$$

(1.2.2) $\boxed{f_{12}l_{12} + f_{13}l_{13} < \sqrt{l_{22}^2 + l_{23}^3}}$

Set

$$e^{i\alpha} = f_{12} + if_{13},$$

$$e^{i\beta} = (l_{22} + il_{23})/(\sqrt{l_{22}^2 + l_{23}^2})$$

and

$$e^{i\gamma} = (-f_{12}l_{12} + f_{13}l_{13})/\sqrt{l_{22}^2 + l_{23}^2} + im,$$

where m is a constant. Equation 4.77 can now be rewritten as

$$cos(\alpha - \beta - \theta) = -cos(\gamma) \Rightarrow \theta = \alpha - \beta \pm \gamma + \pi.$$

Substituting into Equation 4.69 we get

$$\begin{aligned} p_2 &= b_1^{-1}twix(\alpha - \beta \pm \gamma + \pi)trans(x,0,0)a_1p_1 \\ &= b_1^{-1}twix(\pi/2 \pm \gamma)twix(-\beta)trans(x,0,0)twix(\pi/2 + \alpha)a_1p_1. \end{aligned}$$

We observe that in this case there are two solutions. This fact is denoted by an asterisk next to the name of the relationship. We can conclude:

$$\begin{cases} \textit{If } f_{11} = 0 \textit{ then :} \\ E_e = LIN* \\ a_e = twix(\pi/2)twix(\alpha)a_1 \\ b_e = twix(-\pi/2 \pm \gamma)twix(\beta)b_1 \\ \textit{Condition : } f_{12}l_{12} + f_{13}l_{13} < \sqrt{l_{22}^2 + l_{23}^2} \end{cases}$$

(2) Now we examine the general case when $D \neq 0$

In this case we can not determine the value of θ. Therefore the equivalent relationship has at least one rotational degree of freedom.

(2.1) $\boxed{l_{22} = l_{23} = 0}$

Equation 4.75 becomes

$$D_1 = f_{11}(l_{21} - l_{11}) - f_{12}l_{12} - f_{13}l_{13}.$$

Substituting for D_1 into Equation 4.74 we get

$$x = \frac{D_1}{D} = \frac{f_{11}l_{11} + f_{12}l_{12} + f_{13}l_{13}}{f_{11}} - l_{21} = \mu - l_{21}.$$

Substituting in Equation 4.69,

$$\begin{aligned}
p_2 &= b_1^{-1}twix(\theta)trans(\mu - l_{21}, 0, 0)a_1p_1 \\
&= [trans(l_{21}, 0, 0)b_1]^{-1}twix(\theta)[trans(\mu, 0, 0)a_1]p_1.
\end{aligned}$$

Therefore:

$$\begin{cases}
If\ l_{22} = l_{23} = 0\ then: \\
E_e = ROT \\
a_e = trans(\mu, 0, 0)a_1 \\
b_e = trans(l_{21}, 0, 0)b_1
\end{cases}$$

(2.2) $f_{12} = f_{13} = 0 \Rightarrow \boxed{f_{11} = \pm 1}$

Using Equations 4.75 and 4.74 we get

$$x = l_{11} - l_{21},$$

and substituting in Equation 4.69 we get

$$p_2 = [trans(l_{21}, 0, 0)b_1]^{-1}twix(\theta)[trans(l_{11}, 0, 0)a_1]p_1.$$

Therefore:

$$\begin{cases} \textit{If } f_{11} = \pm 1 \textit{ then :} \\ E_e = ROT \\ a_e = trans(l_{11}, 0, 0)a_1 \\ b_e = trans(l_{21}, 0, 0)b_1 \end{cases}$$

(2.3) Finally in the general case:

$$\begin{aligned} x &= (f_{11}l_{11} + f_{12}l_{12} + f_{13}l_{13} - f_{11}l_{21})/f_{11} \\ &\quad -(f_{12}l_{22} + f_{13}l_{23})\cos\theta/f_{11} \\ &\quad -(f_{13}l_{22} - f_{12}l_{23})\sin\theta/f_{11} \\ &= a\cos\theta + b\sin\theta + c, \end{aligned}$$

where a, b, c are constants. Substituting in Equation 4.69 we get

$$\begin{aligned} p_2 &= [trans(l_{21}, 0, 0)b_1]^{-1}twix(\theta)trans(a\cos\theta + b\sin\theta, 0, 0) \\ &\quad [trans(c, 0, 0)a_1]p_1. \end{aligned}$$

Therefore:

$$\begin{cases} \textit{General :} \\ E_e = SCR \\ a_e = trans(c, 0, 0)a_1 \\ b_e = trans(l_{21}, 0, 0)b_1 \end{cases}$$

We observe that in this case x is a function of θ, that is, there is one rotational variable and one translational variable which depends on the rotational.

4.5 Geometric Method for Replacing a 2-cycle of Relationships

In Section 4.4 it was shown how a relationship equivalent to a 2-cycle of relationships can be found by solving the location equations. In this section, I will outline a different procedure. Through an example, I will show how the equivalent relationship can be inferred by examining the geometry of the situation, that is the geometric relation between the coordinate axes of the features. This is the technique that has been used in the RAPT cycle finder system [Popplestone, Ambler, and Bellos 80].

Let us consider again the case FITS-AGPV. The FITS relationship is a relationship between two edges, which are located in such a way that their x axes are aligned, as shown in Figure 4–10. Let E_1=FITS hold between two edges with coordinate systems a_1 and b_1 respectively and E_2=AGPV hold between a plane face with coordinate system a_2 and a vertex with coordinate system b_2[7]. Various constraints on the geometric relation between a_1, a_2, b_1 and b_2 will be considered. In particular, we are interested in the relations between the normal of the plane (x axis of a_2, denoted by x_{a2}), the origin of the vertex (origin of b_2, denoted by O_{b2}), and the direction of the edges (x axis of a_1, denoted by x_{a1} and x axis of a_2, denoted by x_{a2}). Different geometric relations between these, give rise to different equivalent relations. The equivalent relationship and the features between which the new relationship holds will be inferred by examining how the addition of the AGPV constraint affects the degrees of freedom of the FITS relationship.

Consider first the case that the x axes of a_1 and a_2 are perpendicular and the origin of b_2 is on the x axis of b_1. This situation is depicted in Figure 4–11. The

---

[7]Recall that the coordinate system of a feature F is represented by Wfp, where f is the location of the feature and p the location of the object to which it belongs
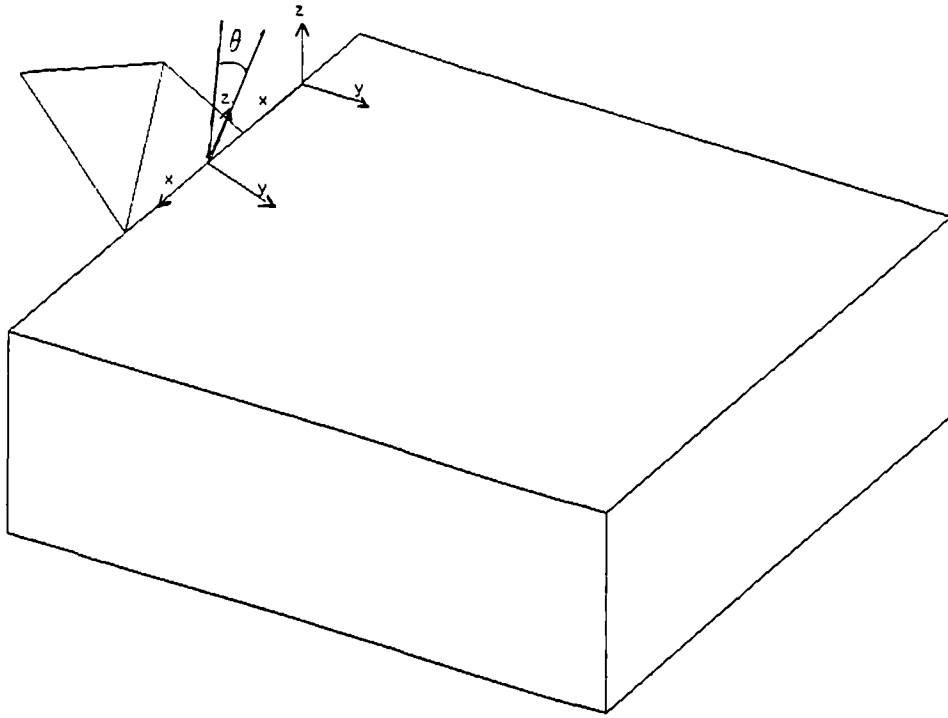
A relationship between two edges such that their x axes are aligned:

$FITS = twix(\theta)trans(x, 0, 0)$

Figure 4–10: FITS relationship
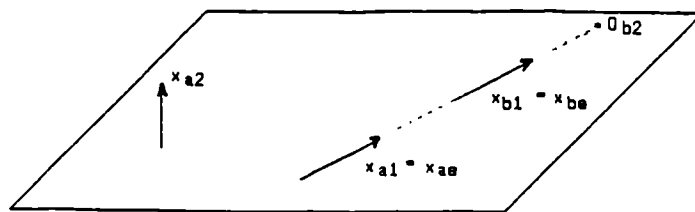
Only the coordinate axes of the features of the objects are shown. x axes of a_1 and a_2 are perpendicular (x-a-perp) and origin of b_2 is on the x axis of b_1 (ob2-on-xb1):

$E_e = FITS, \ a_e = a_1, \ b_e = b_1$

Figure 4–11: FITS-AGPV: x-a-perp, ob2-on-xb1
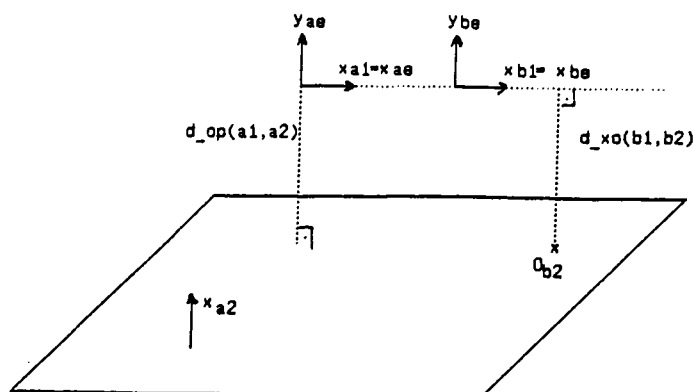
x axes of a_1 and a_2 are perpendicular (x-a-perp) and the distance from the origin of a_1 to yz-plane of a_2 is equal to the distance from origin of b_2 to x axis of b_1 (ed-op-xo):

$$\boxed{E_e = LIN,\ a_e = olx1x2(a_1, a_2),\ b_e = olx1xo(b_1, b_2)}$$

Figure 4–12: FITS-AGPV: x-a-perp, ed-op-xo

abbreviation "x-a-perp" is used to describe the first constraint, while "ob2-on-xb1" is used as an abbreviation of the second constraint. Tables A–2 — A–4 in Appendix A consist of a list of such abbreviations, together with their algebraic interpretation as it will be discussed in Section 4.6.

From Figure 4–11 it can be observed that the addition of the AGPV relationship does not put any new constraints in the relative location of the objects, since a translation along x_{b1} and a rotation about x_{b1} would still leave the vertex on the plane. The equivalent relationship is, thus, the FITS relationship, i.e. E_e=FITS, and the new features are $a_e = a_1$ and $b_e = b_1$.

The next case that will be examined, is the case where the the x axes of $a_1^{\ast}$ and a_2 are perpendicular (x-a-perp), while the distance from the origin of a_1 to the yz-plane of a_2 is equal to the distance from the origin of b_2 to the x axis of b_1. The second condition is denoted by "ed-op-xo". This situation is shown in Figure 4–12. In this case we observe that there is no restriction on the linear degree of freedom of motion, but a rotation about x_{b1} would result in the breaking of the contact between the vertex and the plane. The equivalent

relationship is, thus, a relationship with one translational degree of freedom and no rotational degrees of freedom. Such a relationship is denoted by LIN and holds between two edges.

The new coordinate system a_e can be constructed by making the following observations:

1. Its origin lies on the x axis of a_1.

2. Its x axis has the same direction as the x axis of a_1.

3. Its y axis has the same direction as the x axis of a_2.

Let us choose the origin of the new coordinate system to coincide with the origin of a_1 and let o1x1x2(p1,p2) be a function with the coordinate systems p1 and p2 as arguments which constructs a new coordinate system satisfying the above conditions. Tables A–5 — A–6 in Appendix A summarise the conventions for naming functions used to construct new features. Then,

$$a_e = o1x1x2(a_1, a_2).$$

Similarly, for the case of b_e we can observe that

1. Its origin lies on the x axis of b_1.

2. Its x axis has the same direction as the x axis of b_1.

3. Its y axis has the direction of the perpendicular from the origin of b_2 to the x axis of b_1.

Let o1x1xo(p1,p2) be a function with the coordinate systems p1 and p2 as arguments which constructs a new coordinate system satisfying the above conditions. Then,

$$b_e = o1x1xo(b_1, b_2).$$

The rest of the special cases and the general case are depicted in Figures 4–13 — 4–16.
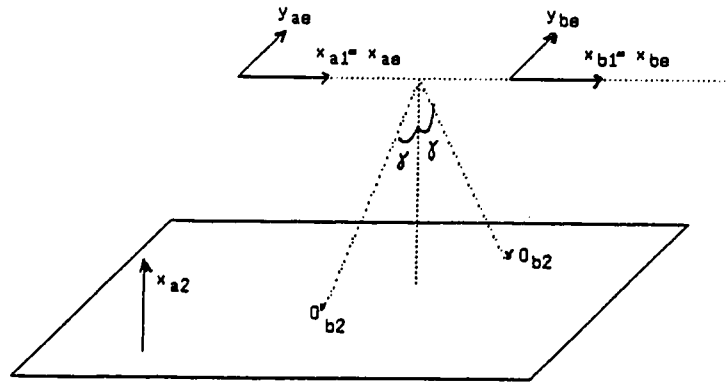
x axes of a_1 and a_2 are perpendicular (x-a-perp) and the distance from the origin of a_1 to yz-plane of a_2 is less than the distance from origin of b_2 to x axis of b_1 (ld-op-xo):

$$E_e = LIN*,\ a_e = o1x1xp(a_1, a_2),\ b_e = q1x1ps(a_1, a_2, b_1, b_2)$$

Figure 4–13: FITS-AGPV: x-a-perp
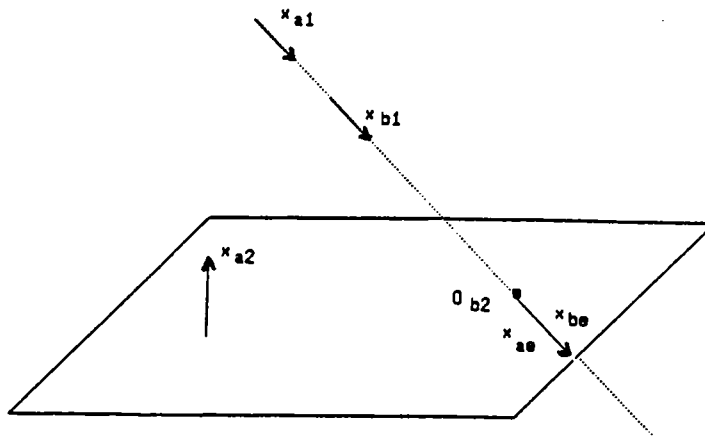
Origin of b_2 is on x axis of b_1 (ob2-on-x-b1):

$$E_e = ROT,\ a_e = opx1y1(a_1, a_2),\ b_e = o2x1y1(b_1, b_2)$$

Figure 4–14: FITS-AGPV: ob2-on-xb1
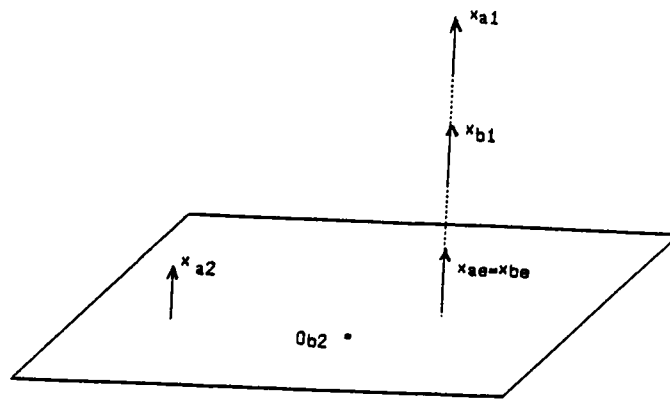
x axes of a_1 and a_2 are parallel (x-a-par):

$$E_e = ROT, \ a_e = opx1y1(a_1, a_2), \ b_e = oox1y1(b_1, b_2)$$

Figure 4–15: FITS-AGPV: x-a-par
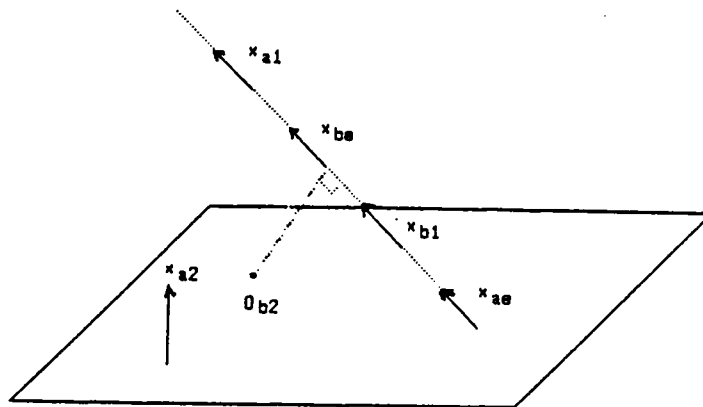
General: $\boxed{E_e = SCR, \ a_e = opx1y1(a_1, a_2), \ b_e = oox1y1(b_1, b_2)}$

Figure 4–16: FITS-AGPV: General

4.6 Combining the Algebraic and the Geometric Method

The advantage of the geometric method is obvious : it is an easier method. However, it relies completely on intuition and it is, therefore, error prone. In addition, it is not possible to decide if all the special cases have been considered. Using the algebraic method, on the other hand we can systematically define the equivalent relationship and the new coordinate systems and deduce the relation between obtained solutions and geometry. The solution of the algebraic equations can be, however, quite troublesome, especially in the case of more than four degrees of freedom.

The actual approach used in this research is a combination of both methods: geometric intuition is used to guide the solutions of the position equations. In addition, once the solutions are derived, they are transformed into geometric functions. This can be done by defining the algebraic equivalent of a geometric function.

There are two cases when we wish to find the algebraic form of a geometric function:

1. in the case of a geometric constraint among the axes of the features which gives rise to special cases in the combination of a certain pair of relationships;

2. in the case of the construction of the new features between which the equivalent relationship holds.

As an example of the first case, let us consider the condition that the distance from the origin of Wa_1 to the $x = 0$ plane of Wa_2, denoted by d-op(a_1, a_2), is equal to the distance from the origin of Wb_2 to the x axis of Wb_1, denoted by d-xo(b_1, b_2). From Figure 4-17 it can be seen that
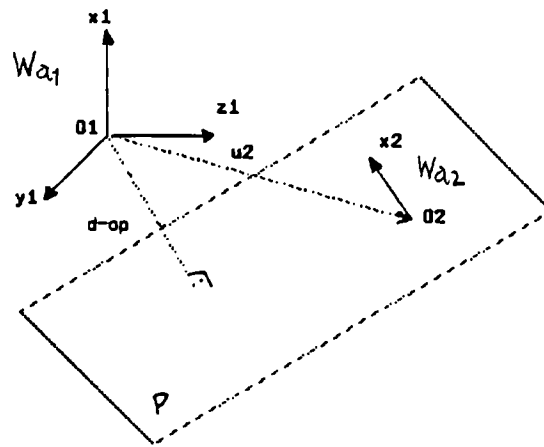
The direction of the normal to the plane P is given by the vector (f_{11}, f_{12}, f_{13}). The vector from O_1 to O_2 is $u_2 = (l_{11}, l_{12}, l_{13})$. Therefore, the distance from O_1 to the plane P is:

d-op$(a_1, a_2) = -(f_{11}l_{11} + f_{12}l_{12} + f_{13}l_{13})$

Figure 4–17: Distance from origin of Wa_1 to plane of Wa_2 (d-op)

$$\text{d-op}(a_1, a_2) = -(f_{11}l_{11} + f_{12}l_{12} + f_{13}l_{13}), \tag{4.78}$$

and from Figure 4–18 it can be seen that

$$\text{d-xo}(b_1, b_2) = \sqrt{l_{22}^2 + l_{23}^2}, \tag{4.79}$$

where

$$F_1 = a_2 a_1^{-1} = \begin{pmatrix} f_{11} & f_{12} & f_{13} & 0 \\ f_{21} & f_{22} & f_{23} & 0 \\ f_{31} & f_{32} & f_{33} & 0 \\ l_{11} & l_{12} & l_{13} & 1 \end{pmatrix},$$

$$F_2 = b_2 b_1^{-1} = \hat{F}_2 trans(l_{21}, l_{22}, l_{23}).$$

Tables A–1 — A–4 in Appendix A, list the geometric functions which have been used for defining the conditions under which special cases arise. In these tables, both the algebraic and the geometric interpretations are given.
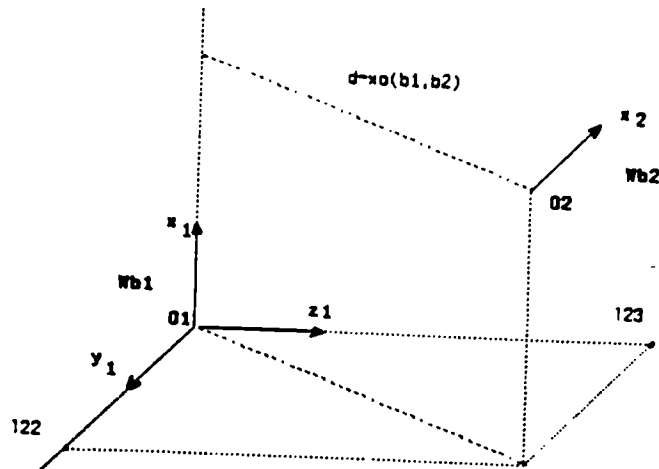
$$d\text{-xo}(b_1, b_2) = \sqrt{l_{22}^2 + l_{23}^2},$$

Figure 4–18: Distance from origin of Wb_2 to x axis of Wb_1 (d-xo)

As an example of a geometric function for constructing the new features, let us consider the case of the function $o1x1xo(p_1, p_2)$, where p_1 and p_2 are coordinate systems with axes $O_1x_1y_1z_1$ and $O_2x_2y_2z_2$ respectively, as shown in Figure 4–19. We wish to define a coordinate system $O_3x_3y_3z_3$ such that O_3 coincides with O_1, x_3 has the same direction as x_1 and y_3 has the direction of the perpendicular from O_2 to x_1. From Figure 4–19 it is clear that

$$p_3 = twix(\beta)p_1,$$

where

$$e^{i\beta} = (l_{22} + il_{23})/(\sqrt{l_{22}^2 + l_{23}^2}).$$

Therefore,

$$o1x1xo(p_1, p_2) = twix(\beta)p_1. \tag{4.80}$$

Similarly, it can be seen that

$$o1x1x2(p_1, p_2) = twix(\alpha)p_1, \tag{4.81}$$
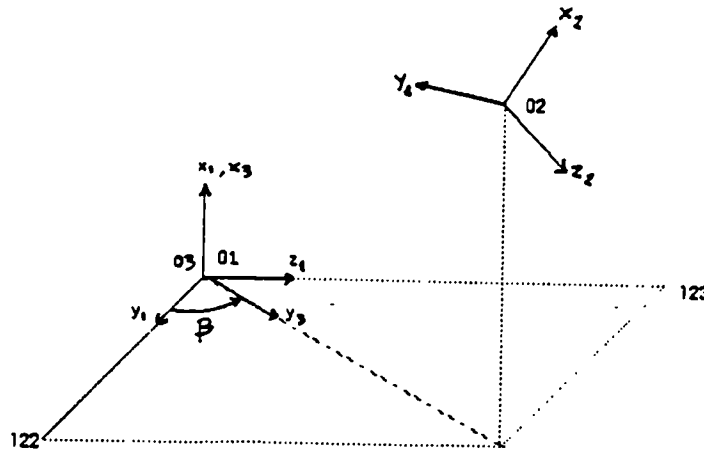
The axes of p_3 =olxlxo(p_1, p_2) are $O_3 x_3 y_3 z_3$ such that O_3 coincides with O_1, x_3 has the same direction with x_1 and y_3 has the direction of the perpendicular from O_2 to x_1.

Figure 4–19: Constructing a location: olxlxo(p_1, p_2)

where

$$e^{i\alpha} = f_{12} + i f_{13}.$$

We can now compare the results obtained by the algebraic and geometric methods in the case (see Figure 4–12)

FITS - AGPV, x-a-perp, ed-op-xo

From Equations 4.78 — 4.81 and from the results in page 151 we can see that the solutions are equivalent:

$$\begin{cases} \textit{If } f_{11} = 0 \text{ (x-a-perp) and } f_{12}l_{12} + f_{13}l_{13} = \sqrt{l_2^2 + l_3^2} \text{ (ed-op-xo) then:} \\ E_e = LIN \\ a_e = twix(\alpha)a_1 = \text{olxlx2}(a_1, a_2) \\ b_e = twix(\beta)b_1 = \text{olxlxo}(b_1, b_2) \end{cases}$$

4.7 Parameterising one degree of freedom relationships

One of the problems the described geometric system has to deal with, is the partitioning of the space of locations satisfying a relationship into legal [8] connected sets (cells), as described in Section 3.6. For the reasons explained there, we are only interested in one degree of freedom relationships. A one degree of freedom relationship is partitioned into intervals by a set of locations which satisfy some additional relationship. For a detailed description of the process refer to Section 3.6. Here we will only be concerned with the requirements which such operations put on the geometric reasoning system. The required capability is summarised below:

> Given a relationship holding between two objects, and a relative location of the objects which satisfies the relationship, we wish to find the value of the free variables of the relationship at that particular location.

Let p be a relative location of the objects A and B which satisfies a relationship E_1 holding between features with locations a_1 and b_1. Then from Equation 4.25

$$p = b_1^{-1} E_1 a_1.$$

The above equation can be rewritten as

$$a_2 = E_1 a_1. \tag{4.82}$$

where a_2 is the location b_1 transformed by p, i.e. $a_2 = b_1 p$. Let

---

[8]Legality implies no body interference

$$F_1 = a_2 a_1^{-1} = \begin{pmatrix} f_{11} & f_{12} & f_{13} & 0 \\ f_{21} & f_{22} & f_{23} & 0 \\ f_{31} & f_{32} & f_{33} & 0 \\ l_{11} & l_{12} & l_{13} & 1 \end{pmatrix}.$$

Equation 4.82 can now be rewritten as

$$F_1 = E_1. \tag{4.83}$$

The parameters of the relationship are found by solving Equation 4.83. The method will be illustrated by an example. Consider the case of a LIN relationship,

$$E_1 = LIN = trans(x, 0, 0).$$

Then,

$$F_1 = trans(x, 0, 0).$$

The conditions for the solution of the above equation are:

$$f_{11} = 1$$
$$f_{22} = 1$$
$$l_{12} = 0$$
$$l_{13} = 0$$

and the solution is

$$x = l_{11}.$$

The conditions for solutions specify the conditions under which a location satisfies a spatial relationship. From Tables A–3 — A–4 it can be seen that the geometric interpretation of the above conditions are:

x-a-eq ∧ y-a-eq ∧ oa2-on-xa1,

while the geometric interpretation of the solution is

$$x = \text{d-op}(a_2, a_1)$$

Table A–32 describes the conditions under which a location satisfies a one degree of freedom relationship and the value of the parameter of the relationship at that location.

4.8 Results and Implementation

In this chapter a spatial reasoning system has been presented which deals with the problem of substituting a pair of spatial relationships holding among features of polyhedral objects by an equivalent, more constrained relationship. A substitution table has been constructed which has entries for various combinations of types of relationships and rules according to which the equivalent relationship and the location of the features between which it holds can be established.

The process by which the table has been constructed is summarised below:

1. The types of all possible five degrees of freedom spatial relationships among the features of two polyhedral objects were first established. These are: against vertex to plane (AGPV), against plane vertex (AGVP) and against edge edge (AGEE)

2. All possible pairwise combinations among these types were considered. For each such pair, all possible types of relationships which can be equivalent to the conjunction of the pair were found, the conditions under which the conjunction gives rise to a particular type of relationship were determined, and rules for the construction of the coordinate systems of the features between which the new relationship holds were developed. All this information was entered in the substitution table.

3. For each new type of relationship, its conjunction with each type of a five degree of freedom relationship was examined, and, as before, an entry in the substitution table was made. This process was followed until all types of relationships were considered.

The method that has been used is based on the algebraic solution of the pair of location equations arising from the spatial relationships. It was shown in Section 4.4.1 that the two equations are of the form (Equations 4.41 and 4.42)

$$p_2 = b_1^{-1} E_1 a_1 p_1,$$

$$p_2 = b_2^{-1} E_2 a_2 p_1,$$

where E_i is a relationship between a feature of object A with location a_i and a feature of object B with location b_i and p_1, p_2 are the locations of objects a and B respectively. Solving these equations, we obtain values for some of the variables of E_1. Substituting these in Equation 4.41 we obtain a new equation of the form

$$p_2 = b_e^{-1} E_e a_e p_1,$$

where E_e is a relationship equivalent to the conjunction of E_1 and E_2 and a_e and b_e are the new features.

The solution of the equations has been guided by examining the geometric relation of the coordinate systems of the features, as described in Section 4.5. Finally, the conditions under which a specific type of equivalent relationship E_e arises and the location of the new features a_e and b_e, were transformed with suitable geometric functions.

The obtained results are presented in the Appendix A. As it can observed, the constructed substitution table (Tables A–14 — A–30), as it currently stands, is not complete, in the sense that there are no entries for some combinations of relationships. In particular, it does not deal extensively with relationships where there is some interdependence between the degrees of freedom. Although the situations in which such relationships arise have been established, their conjunction with another five degree of freedom relationship has not been pursued. An exception has been made in the case of such relationships of one degree of freedom An example is the 'ladder' relationship, denoted by LAD, which is shown in Figure 4–20. As it can be seen from the figure, although there is both a rotational and a translational degree of freedom, they are not independent. The algebraic form of the equation can be found in Table A–13 in Appendix A.

It also has to be observed that, in contrast with the substitution table of the RAPT system, only combinations with a five degree of freedom relationship have been considered. This is a result of the requirements placed upon the spatial reasoning system from the motion planning algorithm presented in this thesis.
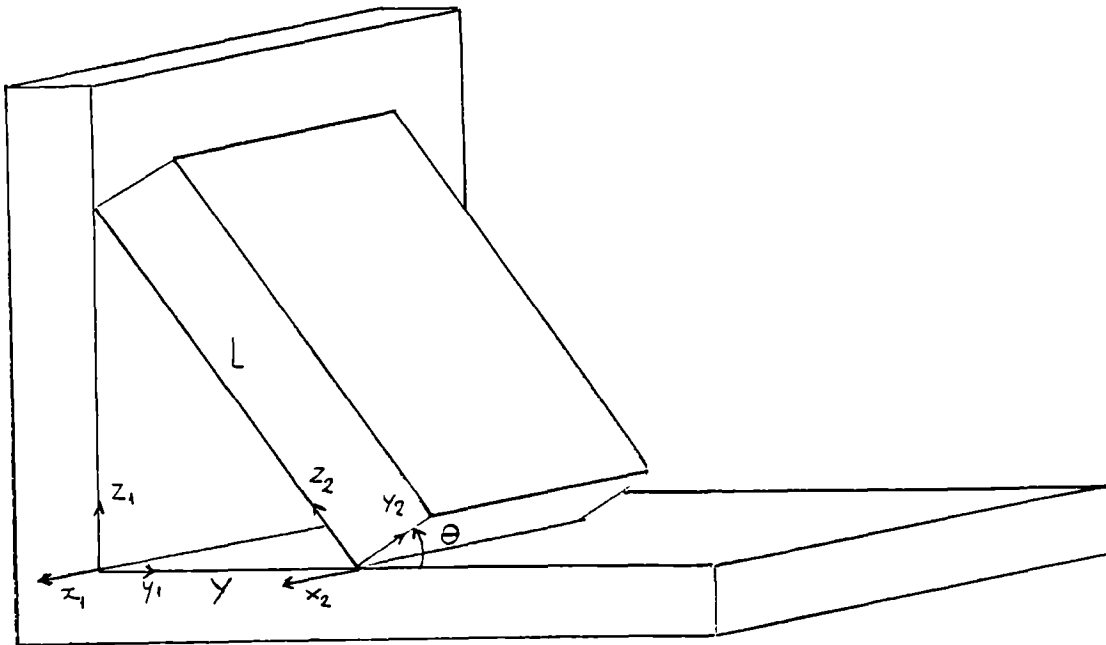
$$LAD = twix(\theta)trans(0, l\sin\theta, 0)$$

Figure 4-20: The 'ladder' relationship, LAD

The substitution table, as described in this chapter and in Appendix A, has been implemented in POP11 [Barrett, Ramsay, and Sloman 85].

Chapter 5

Planning a Motion in Contact

5.1 Introduction

In this chapter I come finally to the problem of developing a plan for a motion in contact. Working towards this objective there are three distinct and interrelated stages.

1. I start by bringing together the theoretical themes developed in Chapters 3 and 4. In Chapter 3 I presented a theory for decomposing the contact space into faces of various dimensions. Following from that I developed an algorithm for constructing the graph of 0- and 1-dimensional faces. In Chapter 4 I presented a spatial reasoning system which handles spatial relationships among features of objects. In this chapter I address the question of how the spatial reasoning system can be used to construct a model of the decomposed contact space. Specifically, I examine ways in which spatial relationships can be used as a means of representing the surfaces of the contact space and finding their intersection.

2. The next step is to discuss the implementation of the algorithm for building the graph of the space. It is in this context that the importance of the spatial reasoning system becomes apparent. The system is used in the first stage of the algorithm, building the graph of surfaces. Once this is accomplished, the algorithm moves on to the building of the graph of 0-dimensional faces (vertices) and 1-dimensional faces (edges) of the space by using a geometric solid modelling system.

3. At this stage the model of the space has been constructed. The decomposed contact space is represented as two graphs: a graph of vertices and edges and a graph of surfaces. Next I use these graphs to plan a motion in contact. A motion in contact is a path in the contact space. Having decomposed the space into path-connected faces, the object of planning a motion in contact becomes finding a sequence of faces that have to be traversed, given some initial and final contact state. If the initial and final states correspond to vertices or edges of the contact space, then a path can be found by searching the graph of vertices and edges. If on the other hand the initial and/or final states correspond to higher dimensional faces, the graph of surfaces is used in order to plan a motion from the initial state to some vertex or edge and from the final state to some vertex or edge.

At the close of the chapter I examine ways in which the information encompassed in these two graphs can be used for the purpose of transforming the path into a sequence of motions.

5.2 Constructing a Model of the Contact Space

In Chapter 3 a theory was developed for decomposing the contact space into faces of various dimensions, the properties of the decomposition were explored and an algorithm was presented for finding the graph of edges and vertices of the space. Each n-dimensional face of the space is a region of some n-dimensional manifold (surface). The algorithm is based on the fact that all these surfaces of the space can be found by considering the 5-dimensional surfaces and their intersections. The algorithm consists of two main steps:

1. Find the surfaces of the space, by considering all possible intersections of 5-dimensional surfaces.

2. Find the 1-skeleton of the space, that is the 0- and 1-dimensional faces of the space.

It was pointed out in Section 3.5 that a spatial reasoning system would be used for dealing with the problem of intersections of surfaces and for representing a model of the thus decomposed contact space.

Having presented the spatial reasoning system, we are now in a position to revise and extend the algorithm of Section 3.5. In Section 5.2.1 spatial relationships are put into the context of the concepts introduced in Chapter 3. Section 5.2.2 deals with the problem of finding the surfaces of the space using the spatial reasoning system. In Section 5.2.3 an algorithm is presented for constructing the 1-skeleton of the space, using a solid modelling system. Finally, the developed model of the space is overviewed in Section 5.2.4.

5.2.1 Clauses, Descriptors and Spatial Relationships

Clauses and 5 d.o.f. spatial relationships

In Section 3.4.1 the notion of a clause was introduced to describe the three basic types of contact among polyhedral objects (Definition 3.4.1). More specifically, a clause c was defined to be a pair (F_1, F_2), where F_1 and F_2 are features of two objects and either F_1 is a vertex and F_2 is a plane face, or F_1 is a plane face and F_2 is a vertex or, finally, both F_1 and F_2 are edges. In Section 4.3.2 the three basic types of contact were defined by means of three types of five degrees of freedom spatial relationships, the AGPV, AGVP and AGEE relationships.

The important difference in the two definitions is that if a location satisfies a clause then the features in the clause are in contact, while this is not necessarily so if the location satisfies a spatial relationship[1]. This distinction was introduced in Section 3.4.1 by associating with each clause c a real valued function f_c on location space, such that for every location p which satisfies the clause, $f_c(p) = 0$. Let, for example, $c = (F_1, F_2)$ be a clause, where F_1 is a vertex of the moving object and F_2 is a plane face of the environment. The set of locations p which satisfy the clause lie on a surface of the contact space. The equation $f_c(p) = 0$ is the equation of this surface.

In Section 4.3.2 it was shown that spatial relationships describe surfaces in location space, defined by the set of locations which satisfy the relationship. Let R be a spatial relationship "against plane vertex" (AGPV) between the vertex F_1 of the moving object and the plane face F_2 of the environment and let f_1 and f_2 be the locations of features F_1 and F_2 respectively. Then from the equation (Equation 4.37)

$$p = f_1^{-1} twix(\theta) XTOY twix(\phi) XTOY twix(\psi) trans(0, y, z) f_2,$$

---

[1]By location we mean the relative location of the objects, or the location of the moving object in the case where there is only one moving object.

where p is the location of the moving object, the equation of the surface defined by the set of locations satisfying the relationship can be derived.

The relation between the function f_c associated with a clause $c = (F_1, F_2)$ and a five degree of freedom spatial relationship R holding between the features F_1 and F_2 now becomes apparent: they can be both used to represent the space of locations which satisfy the clause, if the features F_1 and F_2 are considered to be of infinite extent (in the case when the features are edges or plane faces). We can, therefore, associate with each clause a spatial relationship, replacing the function f_c. Let $R_c = (R, f_1, f_2)$ be the 5 degree of freedom spatial relationship R holding between features F_1 and F_2 corresponding to a clause $c = (F_1, F_2)$, where f_1 and f_2 are the locations of F_1 and F_2 respectively.

Descriptors and spatial relationships

The notion of a descriptor D was introduced in order to express all types of contact among polyhedral objects. A descriptor was defined to be a predicate that is a conjunction of clauses. With every descriptor there is an associated system of equations of the form $f_{c_i} = 0$, where c_i is a clause of the descriptor. If a location p satisfies the descriptor then it satisfies the system of equations $f_{c_i}(p) = 0$ for c_i in D.

Since for every clause there is an associated 5 degrees of freedom spatial relationship, every location which satisfies the descriptor D, satisfies the relationships R_{c_i} for c_i in D. The set of locations H_D which satisfy the system of equations lies on the intersection of the five dimensional surfaces defined by $f_{c_i} = 0$. The equation of the intersection can be found from the equation

$$p = f_2^{-1} R f_1,$$

where R is a relationship equivalent to the conjunction $\bigwedge_i R_{c_i}$ and f_1 and f_2 are the locations of the features between which R holds. Thus

$$H_D = \{p \mid p = f_2^{-1} R f_1\}.$$

In this way spatial relationships can be used to represent the set of locations satisfying the system of equations of a descriptor. Therefore, we can associate with a descriptor D, instead of the system of equations of the form $f_{c_i} = 0$, a spatial relationship. Let $R_D = (R, f_1, f_2)$ be the spatial relationship associated with a descriptor D.

Two descriptors have been defined to be equivalent if the associated systems of equations have the same set of solutions. Therefore, two descriptors D_1, D_2 are equivalent if the associated relationships are equivalent,

$$D_1 \sim D_2 \text{ iff } R_{D_1} \sim R_{D_2}.$$

Seen in this context, the substitution table presented in Chapter 4 can be used in order to find the parametric equation of the intersection of a five dimensional surface with a n-dimensional surface ($n \leq 5$). Therefore, spatial relationships and the substitution table can be used to find the intersection of the surfaces of the contact space or, in other words, sets of locations H_D which satisfy the system of equations of a descriptor D.

5.2.2 The Surfaces of the Contact Space

This section deals with the first phase of the algorithm presented in Section 3.5. In the first phase, starting from the 5-dimensional surfaces, the n-dimensional surfaces are found by considering intersections of $(n + 1)$-dimensional surfaces with 5-dimensional surfaces. The process is summarised below:

1. Find all sets H_D where D is a 5-dimensional descriptor, i.e. $dof(D) = 5$;

2. Find all consistent descriptors, their dimension and the corresponding sets H_D by the following method:

   Repeat for $n = 5, 4, \ldots, 1$: For each $D = D_1 \wedge D_2$ where $dof(D_1) = n$ and $dof(D_2) = 5$ find the set H_D. If H_D is empty, then D is an inconsistent descriptor and can be discarded.

As was shown in Section 3.5.2 the solution sets H_D of all consistent descriptors constitute a lattice. It was also noted that we wish to find the connected subsets of the sets H_D, and to associate with each such subset a spatial relationship $R_D = (R, f_1, f_2)$. The objective of the above algorithm is to construct the graph of the surfaces of the contact space.

Representation of the graph of the contact space

The elements of the graph of the contact space are the surfaces of the space, An element of the graph will be represented as a node, $node_D$. A node consists of:

1. A descriptor D;

2. The spatial relationship of the descriptor $R_D = (R, f_1, f_2)$.

The clauses of the descriptor of $node_D$ will be denoted by $clauses(node_D)$ and the spatial relationship by $Rel(node_D)$. For every node the following information is thus available: (1) the 5-dimensional surfaces on whose intersection the surface represented by the node lies; (2) the parametric equation of the surface

(implicitly). The dimension of a node is equal to the dimension of the surface which it represents. This is also equal to the dimension of the descriptor and the degrees of freedom of the spatial relationship:

$$dim(node_D) = dim(D) = dof(R_D).$$

I will use the term *level* of the graph to refer to the set of nodes of the same dimension. There are six levels of the graph. Let L^d denote the dth level of the graph, that is the set of nodes of dimension d,

$$L^d = \{node_D | dim(node_D) = d\}.$$

A node $node_{D_1}$ is *descendant* of a node $node_{D_2}$ if the set of configurations which satisfy the spatial relationship of $node_{D_1}$ is a subset of the set of configurations which satisfy the relationship of $node_{D_2}$. For this to be the case, we require that (a) the set of clauses of D_1 is a superset of the set of clauses of D_2,

$$clauses(node_{D_1}) \supset clauses(node_{D_2});$$

(b) some configuration which satisfies the relationship $Rel(node_1)$ also satisfies the relationship $Rel(node_2)$. The first requirement follows from Equation 3.3 (page 96). As has been explained in Section 3.5.2, there are cases where two nodes have the same sets of clauses, but different spatial relationships. This situation arises when there are more than one solutions to a pair of location equations. For this reason, it is not sufficient to examine the clauses of a node. The second requirement is placed so as to guarantee that the $node_1$ is a descendant of $node_{D_2}$. As an example, consider the situation shown in Figure 3–32 (page 99). The node $n_{D_{10}}$ is not a descendant of n_{D_8}, although $clauses(n_{D_{10}}) \supset clauses(n_{D_8})$.

Algorithm for constructing the graph

Having established the relation between clauses, descriptors, surfaces and spatial relationships, the algorithm can now be reformulated in terms of spatial rela-

tionships. The algorithm for constructing the graph of spatial relationships is given below.

1. Find all the clauses of the contact space and the corresponding spatial relationships and construct thus the fifth level of the graph, L^5.

2. Repeat for $d = 5, 4, 3, 2$ and 1:

   For each pair of a d-dimensional node with a 5-dimensional node:

   (a) Using the substitution table examine if the spatial relationships of the two nodes can be satisfied simultaneously. If they can, find the equivalent relationship and the features between which it holds.

   (b) If an equivalent relationship has been found, create a new node whose descriptor is the conjunction of the descriptors of the initial nodes and whose spatial relationship is the spatial relationship equivalent to the the conjunction of the relationships of the initial nodes.

   (c) Obtain the degrees of freedom of the relationship of the new node from a table (see Table A-13) and insert the node at the appropriate level of the graph, i.e. if $dof(R) = 4$ then insert the new node in L^4.

This algorithm, which will be referred to as Algorithm GRA, is presented in more detail in Figure 5–1. The algorithm is explained in more detail below by examining separately some issues.

1. The clauses of the contact space

In the first step of the algorithm the set of all possible clauses of the space are found and a node is created for each clause. This means that all vertices, edges and faces of the moving object and the environment have to be found, and the appropriate pairs constructed. Transforming the clauses into spatial relationships requires that the locations of these features be known. The locations of the features can be found using a solid modelling system. The current system

begin

1. construct L^5;

    for $n \leftarrow 5$ **down to 1 do**

      for all n_1 **in** L^n **do**

       for all n_2 **in** L^5 **do**

        begin

        $D_3 \leftarrow clauses(n_1) \cup clauses(n_2)$;

        if $clauses(n_2) \not\subseteq clauses(n_1)$ **and**

         D_3 is not in $FOUND$ **and**

         $D_3 \not\supseteq$ of a member of $INCONSISTENT$ **then**

         use substitution table to find $Rel(n_1) \wedge Rel(n_2)$;

2.   **if** $inconsistent(Rel(n_1) \wedge Rel(n_2))$ **then**

         insert D_3 in $INCONSISTENT$

        else

3.   **if** $Rel(n_1) \wedge Rel(n_2)$ has been solved **then**

         begin

4. $R_3 \leftarrow Rel(n_1) \wedge Rel(n_2)$;

5. $d \leftarrow dof(R_3)$;

6.   **if** $R_3 \sim Rel(n_3)$ where n_3 is a node in $FOUND$ **then**

          $clauses(n_3) \leftarrow clauses(n_3) \cup D_3$

         else

          begin

          make new node: $D = D_3$ and $R_D = R_3$;

          insert node in L^d;

          insert D_3 in $FOUND$;

          end

         end

       end

end

Figure 5–1: An algorithm for constructing the graph of the contact space(GRA)

though uses the RAPT input system for defining the models of objects in terms of their features.

Problems may arise from the fact that the number of nodes that is generated can be very large so that the number of vertices of the contact space would be astronomical. One possible way of dealing with this problem would be to construct the model of the space incrementally. That is to say, only a subset of the clauses is initially considered and then clauses are added as required. This issue is discussed in Sections 5.4 and 6.3.2.

2. Keeping track of inconsistent combinations

The algorithm GRA keeps track of inconsistent combinations of clauses so that not all possible descriptors would have to be considered. If for example, $D_i = \bigwedge_i c_i$ is found to be inconsistent, then all descriptors which contain the clauses c_i are inconsistent and do not have to be examined. For this reason every time an inconsistent descriptor is found, the set of the clauses of the descriptor is entered into the set of inconsistent sets of clauses, *INCONSISTENT*.

3. Dealing with an incomplete substitution table

It is clear that a descriptor $c_1 \wedge c_2 \wedge c_3$ is equivalent to a descriptor $c_2 \wedge c_3 \wedge c_1$, that is to say, the order of the clauses is not important. Therefore, it would be sufficient to examine all combinations of clauses, and not all ordered combinations of clauses.

As it has been mentioned in Chapter 4, the substitution table of the spatial reasoning system is not complete. As a result, the order in which a certain combination of clauses is considered is of importance. In some cases the combination of a certain pair of relationships cannot be handled by the system. For example, the spatial reasoning system may be able to deal with the combination $c_1 \wedge (c_2 \wedge c_3)$ but not with the combination $(c_1 \wedge c_2) \wedge c_3$.

For this reason, algorithm GRA considers different ordered combinations until the substitution table can produce a solution. Specifically, each node $node_1$

is conjoined with *all* 5-dimensional nodes *node₂* which yield combinations for which no nodes in the model exist yet, that is $clauses(node_1) \cup clauses(node_2)$ is not in *FOUND*, where

$$FOUND = \bigcup_i \{clauses(node_i)\} \quad \forall node_i \in L.$$

There is no guarantee, however, that the substitution table will always find an equivalent relationship or determine that the combination is inconsistent. As a result, there might be cases when there are no nodes for some surfaces of the graph, especially for high dimensional surfaces.

4. Double solutions

In chapter 4 it has been shown that there are cases when there are two sets of solutions for a combination of two relationships. Geometrically this means that there are two different surfaces of intersection. In this case both solutions have to be retained. Therefore, there are nodes where the descriptors are the same but the relationships are not equal. In most cases, one node would consist of illegal locations. This is illustrated in Figure 5-2. In both cases shown in this figure, the bottom vertices ($bv1, bv2, bv3$) of the block B are 'against' the top face (*top*) of the obstacle and vertex $bv1$ of the block is 'against' the back face of the obstacle E. This results in a ROTYLIN relationship between the two objects. The difference between (a) and (b) is that in (a) the x axis of the bottom face of the block points downwards, that is the block rests on the obstacle, while in (b) the x axis points upwards. Case (b) shows an illegal location.

5. Degrees of freedom

The degrees of freedom of a relationship determine the level of the graph at which a new node should be inserted. Usually, the conjunction of a n d.o.f. relationship with a 5 d.o.f. relationship will give an $(n - 1)$ d.o.f. relationship. As can be seen from the substitution table, there are special cases where a $(n - 2)$ relationship can be produced. Therefore, although we would expect the
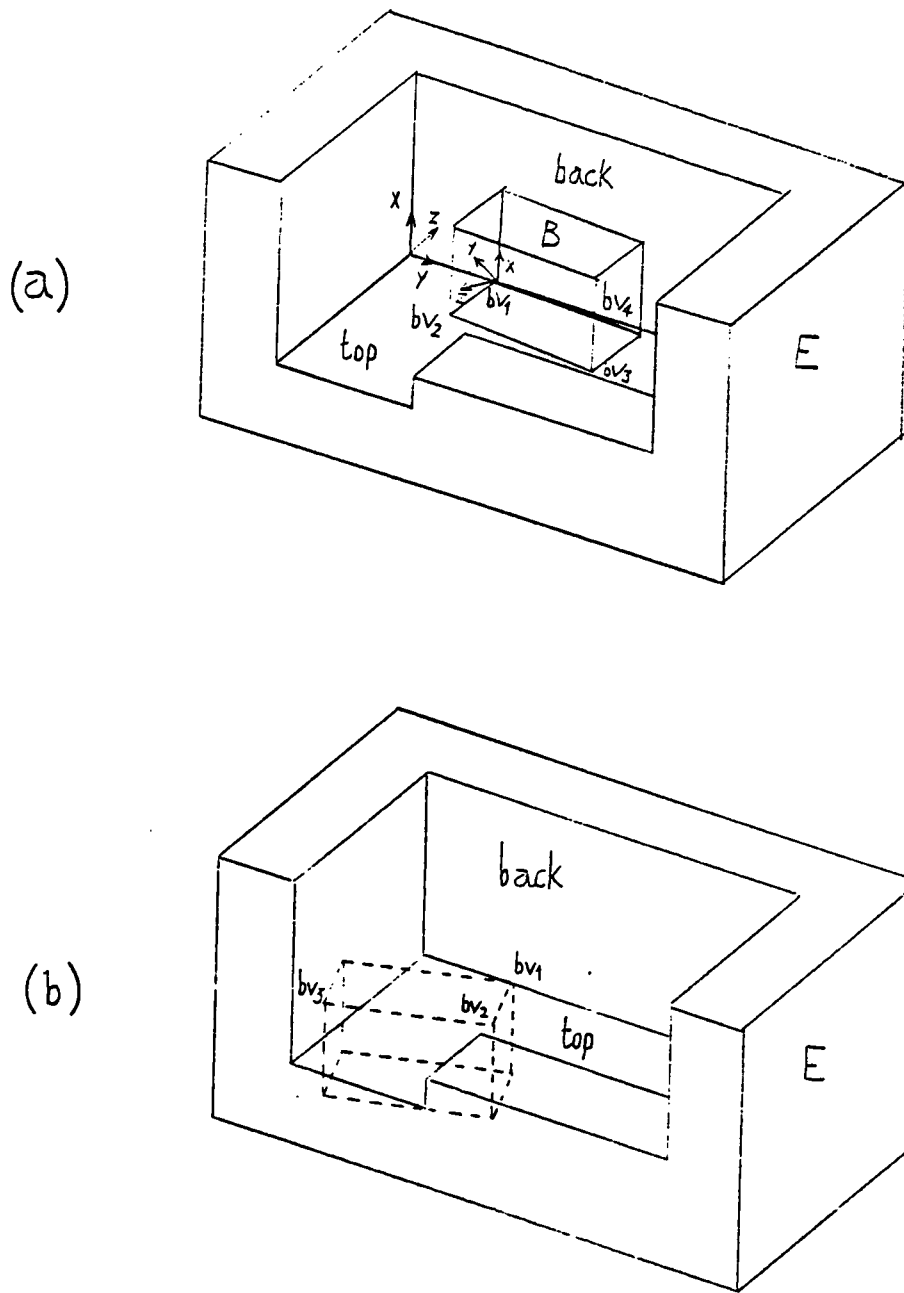
For both cases (a) and (b):

D=(bv1,top) ∧ (bv2,top) ∧ (bv3,top) ∧ (bv1,back)

R=ROTYLIN

In (a) object B is 'on' the top face of E, while in (b) object B is 'under' the top face of E. Case (b) shows an illegal location, a location for which the two objects interfere.

Figure 5–2: Double solutions

descriptor of a n-dimensional node to be a conjunction of $(6-n)$ clauses it could be the conjunction of less than $6 - n$ clauses.

As an example of this, refer back to Figure 3.25 (page 79), when the problem of whether the decomposition of the contact space is a cell complex was examined. In this case the conjunction of a ROT2 relationship (two rotational degrees of freedom) with an AGPV relationship, gives a FIX relationship (zero degrees of freedom) as can be seen from the substitution table.

Algorithm GRA obtains the number of the degrees of freedom of a relationship using a table which assigns a number to each type of relationship (Table A-13).

6. Equivalent Relationships

In order to have only one node representing each surface of the contact space, we have to deal with the problem of equivalent descriptors. Two descriptors have been defined to be equivalent (Definition 3.4.8) if the corresponding systems of equations have the same sets of solutions, i.e. the surfaces are identical. Therefore, two descriptors D_1 and D_2 are equivalent if the associated spatial relationships are equivalent, i.e. $R_{D_1} \sim R_{D_2}$.

Let $D_3 = D_1 \wedge D_2$, where $dim(D_2) = 5$, and D_3 is a consistent descriptor. As can be seen from the substitution table, in some special cases the new relationship is equal to one of the initial relationships, i.e. $R_{D_3} = R_{D_1}$, which means that relationships R_{D_1} and R_{D_3} are relationships of the same type and have the same feature locations. In this case, D_3 is equivalent to D_1, $D_1 \sim D_3$. Intuitively, this happens when the addition of a contact does not affect the degrees of freedom of motion. The geometric interpretation is that the surface H_{D_1} lies on the surface H_{D_2}. An example is shown in Figure 5-3.

The above case occurs when a constraint is superfluous. There is also the possibility that two descriptors of the same dimension have the same set of solutions. Intuitively, this happens when two different combinations of contacts
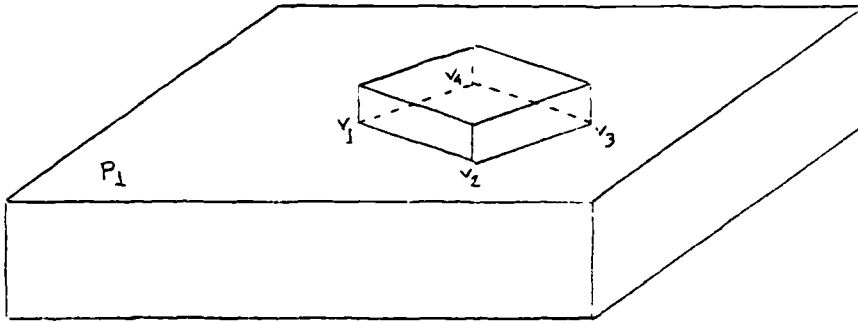
$D_1 = (v_1, p_1) \wedge (v_2, p_1) \wedge (v_3, p_1), \; R_{D_1} = (AGPP, f_1, f_2)$

$D_2 = (v_4, p_1), \; R_{D_2} = (AGPV, f_1, f_3)$

$D_3 = D_1 \wedge D_2 = (v_1, p_1) \wedge (v_2, p_1) \wedge (v_3, p_1) \wedge (v_4, p_1), \; R_{D_3} = R_{D_1}$

The relationships R_{D_1} and R_{D_3} are equal and, therefore, the descriptors D_1 and D_3 are equiv-

alent, $D_1 \sim D_3$

Figure 5–3: Equivalent Relationships – 1
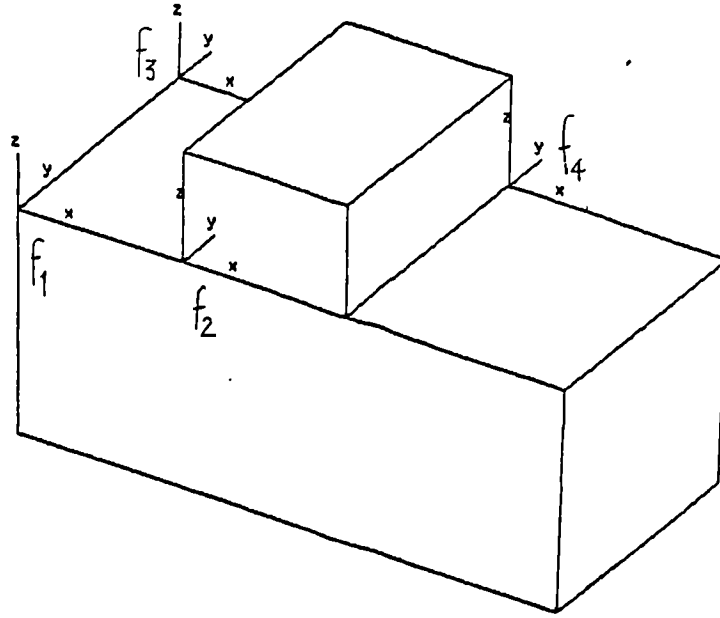
$$R_{D_1} = (LIN, f_1, f_2)$$

$$R_{D_2} = (LIN, f_3, f_4)$$

$R_{D_1} \sim R_{D_2}$ according to Table A-31. Therefore $D_1 \sim D_2$.

Figure 5-4: Equivalent Relationships - 2

give rise to the same situation. As an example, consider the case shown in Figure 5-4, where the two LIN relationships are equivalent and thus the associated descriptors are equivalent.

Table A.31 in Appendix A is used to determine if two relationships are equivalent. Let $R_{D_1} = (R, f_1, f_2)$ and $R_{D_2} = (R, f_3, f_4)$ be two relationships of the same type. Table A.31 gives the conditions on the locations of the features f_1, f_2, f_3, f_4 which have to be tested in order to determine if the two relationships are equivalent.

In both cases, algorithm GRA does not create a new node for a relationship which is equivalent to some relationship already in the graph. Instead, it updates the clauses of the node to be the union of the clauses of the two equivalent descriptors. Therefore, when the graph is finally built, the clauses of a node $node_D$ will be (cf. Equation 3.2, page 96),

$$clauses(node_D) = \bigcup_{D_i \in \tilde{D}} clauses(D_i).$$

Although not shown in Figure 5-1, the algorithm also keeps track of the various subsets of the clauses of a node which are equivalent. For example, in the case shown in Figure 5-3, the clauses of the node will be the set of all four clauses (v_i, p_1), $i = 1 \ldots 4$. It will also be known that a combination of any three of them gives a descriptor equivalent to the descriptor of the node. In this way, a node can be accessed by specifying a sufficient but not necessary complete set of clauses.

An example

Figure 5-5 shows two objects and the names of some of their features. A partial model of the contact space was first constructed: a set of ten 5 d.o.f. relationships was specified between the objects shown in the figure resulting to the following set of clauses:

$CLAUSES =$

$\{(bv1, bot), (bv2, bot), (bv3, bot), (bv4, bot), (bv1, left),$

$(bv1, back), (bv2, left), (bv4, back), (bv4, left), (bv4, right)\}$

The constructed graph of the contact space consists of a total of 216 nodes. The CPU time for constructing the graph was 187 secs.
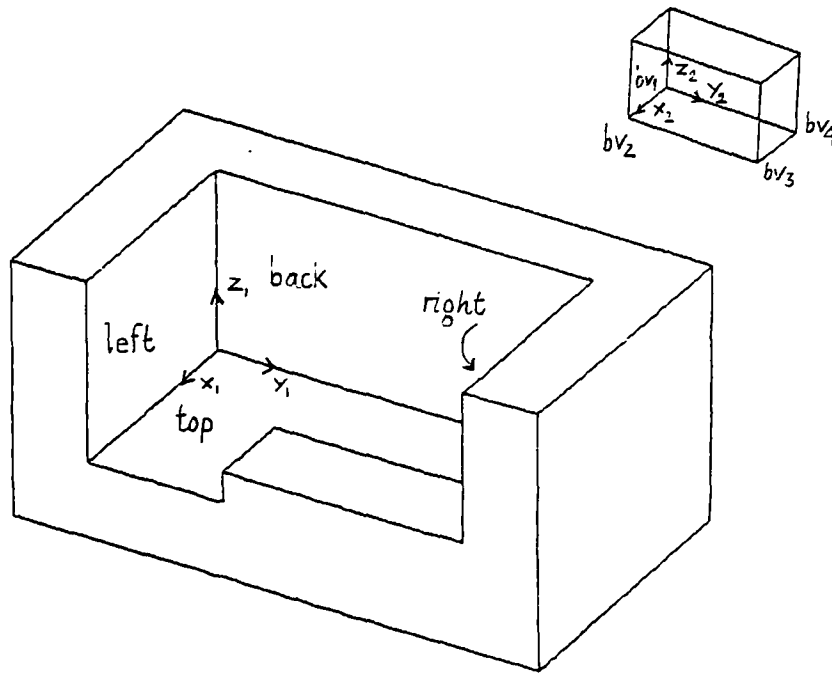
Figure 5–5: The models of two objects used in the example

5.2.3 Vertices and Edges of the Contact Space

In the previous section it was shown how the surfaces of the contact space can be found using the spatial reasoning system. The surfaces were represented as a graph of nodes, where for each node there is a descriptor and a spatial relationship. The problem which will be considered now is partitioning these surfaces into the faces of the contact space.

Let us recapitulate the definition of a face of the space (see Section 3.4.6). A n-dimensional face is subset of a n-dimensional surface of the the contact space such that:

1. all locations on the face are legal locations for which the objets are in contact.

2. the face is path-connected

3. all locations in the interior of the face satisfy a unique sign-assignment.

It has been shown in Section 3.5.4 that a n-dimensional surface is partitioned into n-dimensional faces which fulfill the above requirements by $(n-1)$-dimensional surfaces which lie on the intersection of the surface with some 5-dimensional surface. Here we will only be concerned with the problem of finding the 0- and 1-dimensional faces of the space, i.e. the vertices and the edges of the space. As will become clear, the technique cannot be generalised to higher dimensional faces.

Let us first consider the vertices of the contact space. A 0-dimensional face consists of a single location. Therefore requirements 2 and 3 are redundant and thus a 0-dimensional surface is a vertex of the space if the location is a legal location, i.e. it does not result in body interference. The location corresponding to a 0-dimensional node $node_D$ is given by

$$p = f_2^{-1} f_1,$$

```
procedure Findvertices(L⁶):
begin
    for all n in L⁶ do
        if legal(n) then
            make new vertex v from n
            insert new vertex v in Vertices
end
```

Figure 5-6: A procedure for finding the vertices of the contact space

where $R_D = (FIX, f_1, f_2)$ is the spatial relationship of the node. A 0-dimensional node $node_D$ is legal if the moving object at location p does not overlap the environment. A vertex v is thus created for each legal 0-dimensional node $node_D$. With each vertex v we associate a node $node_D$ and a set of edges on which the vertex lies. The second operation is performed after the edges of the space are determined.

In the implemented system, a solid geometric modeller has been employed for dealing with the problem of body interference [Cameron 84]. Given a solid model of the moving object and the environment and a location of the moving object, the modeller is able to decide whether the object overlaps the environment by considering whether or not the regularised intersection of the two is null or not.[2]

The procedure for finding the vertices of the space is shown in Figure 5-6.

Figure 5-7a shows a legal 0-dimensional surface, which is, therefore, a vertex of the contact space, while Figure 5-7b shows a 0-dimensional surface which is

---

[2]In order to check whether two objects are in contact, the modeller should check whether the intersection of the *interiors* of the objects is null, while the intersection of the objects is not null. This is outside the current capabilities of the employed modelling system.
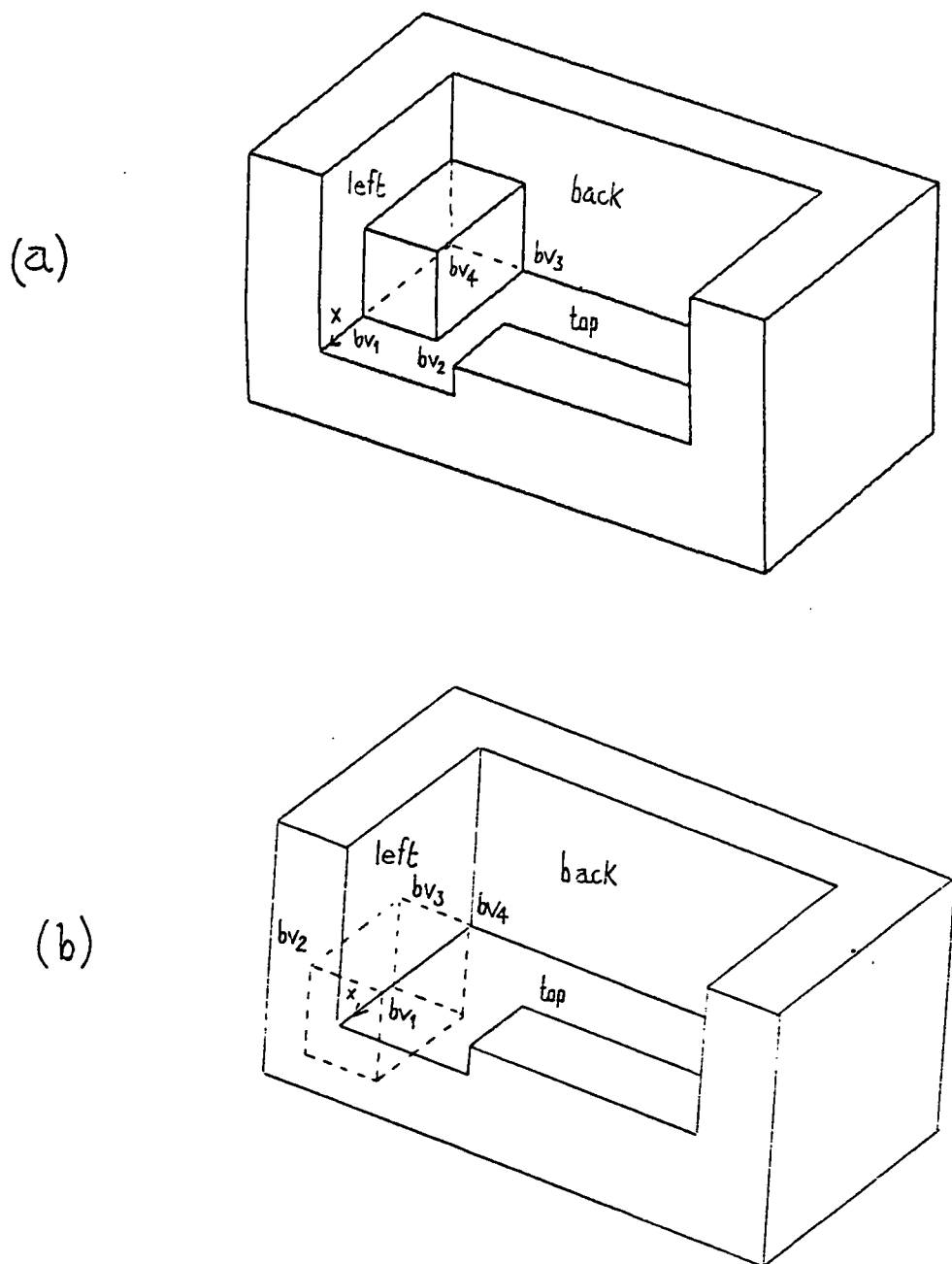
(a) and (b) have the same descriptor:

$$D = (bv1, top) \land (bv2, top) \land (bv3, top) \land (bv4, top) \land (bv1, left) \land (bv4, back) \land (bv4, left).$$

Case (a) shows a legal 0-dimensional surface, i.e. a vertex. In (b) object B is rotated about the x axis by π resulting to an illegal location.

Figure 5–7: A vertex and an illegal 0-dimensional surface of the contact space

illegal. In both cases the descriptor is the same: the bottom vertices of B are against the top face of E, vertex $bv4$ is against the left and back faces of E and vertex $bv1$ is against the left face of E. The two locations shown are the two solutions of the descriptor (cf. Figure 5–2).

Having thus found the vertices of the contact space we now proceed to find the edges of the space. A 1-dimensional surface is partitioned into intervals by the vertices which lie on the surface. The vertices which lie on the surface can be found by considering the descendants of the 1-dimensional node corresponding to the surface. Each such interval can be characterised either as legal or as illegal by examining the legality of some arbitrary point in the interval (Section 3.5.4, Theorem 3.5). Each legal open interval corresponds to an edge of the contact space.

The algorithm for finding the edges of the space is given in Figure 5–8. The steps of the algorithm are explained in more detail below:

Let $node_D$ be a 1-dimensional node of the graph and $R_D = (R, f_1, f_2)$ be the 1 d.o.f. relationship of the node, where $R \in \{LIN, ROT, LAD, SCR\}$ and D is the descriptor of the node. Let $p = p(x)$ be the parametric equation of the node, i.e.

$$p(x) = f_2^{-1} R f_1.$$

1. The descendants *desc* of a 1-dimensional node $node_D$ are the vertices which lie on the 1-dimensional surface of the node. Thus a vertex v corresponding to a node *node* is a descendant of the node $node_D$ if

   $$clauses(node) \supset clauses(node_D),$$

   and if the configuration of v satisfies the spatial relationship $Rel(node_D)$. Let $desc = \{v_1, v_2 \ldots v_n\}$.

2. For every vertex v_i in *desc*, the value of the parameter of the relationship defining the surface is calculated. This is done using Table A-32 of Appendix A. Let P be the set of values obtained thus.

```
procedure Findedges(L⁵,Vertices):
begin
      for all n in L⁵ do
1.        desc ← finddescendants(n, Vertices);
2.        P ← parameterise(n, desc);
3.        Pₒ ← sort(P);
4.        I ← partition(Pₒ);
          for all i in I do
                begin
                      m ← midpoint of i;
5.                    if legal(p(m)) then
                            begin
6.                                make new edge e from n;
7.                                find boundary vertices v₁, v₂ of e;
                                  mark vertices v₁, v₂ as incident on the edge e;
                                  insert edge e in Edges;
                            end
                end
end
```

Figure 5–8: A procedure for finding the edges of the contact space

3. The set P is ordered. Let P_o be the ordered sequence.

4. The sequence $P_o = \{x_1, x_2, \ldots x_i \ldots x_n\}$ is partitioned into intervals

$$i_1 = [x_1, x_2], \ldots, i_{i-1} = [x_{i-1}, x_i], i_i = [x_i, x_{i+1}], \ldots, i_{n-1} = [x_{n-1}, x_n].$$

If the parameter of the relationship R_D corresponds to a rotational degree of freedom then the interval $i_n = [x_n, x_1]$ has to be considered also. Let I be the set of intervals,

$$I = \{i_1, i_2, \ldots, i_n\}.$$

5. The midpoint $m = (x_i + x_{i+1})/2$ of each interval i_i is considered. If the location corresponding to m is legal, i.e. if $p(m)$ is a legal location, then all the locations in the interval $(p(x_i), p(x_{i+1}))$ are legal (Theorem 3.5). In this case we say that the interval i_i is legal.

6. For all legal intervals i_i, an edge e is generated. The equation of the edge is given by the equation $p(x)$ and the endpoints are given by $p(x_i)$ and $p(x_{i+1})$.

7. The bounding vertices of the edge are the vertices whose locations are $p(x_i)$ and $p(x_{i+1})$. These vertices are marked as incident on the edge e.

Figure 5–9 shows how the 1-dimensional surface corresponding to a LIN relationship is partitioned into edges.

The 1-skeleton of the subspace of the contact space for the case of the example of the previous section consists of four vertices and four edges. The CPU time for constructing the 1-skeleton is 78 secs.
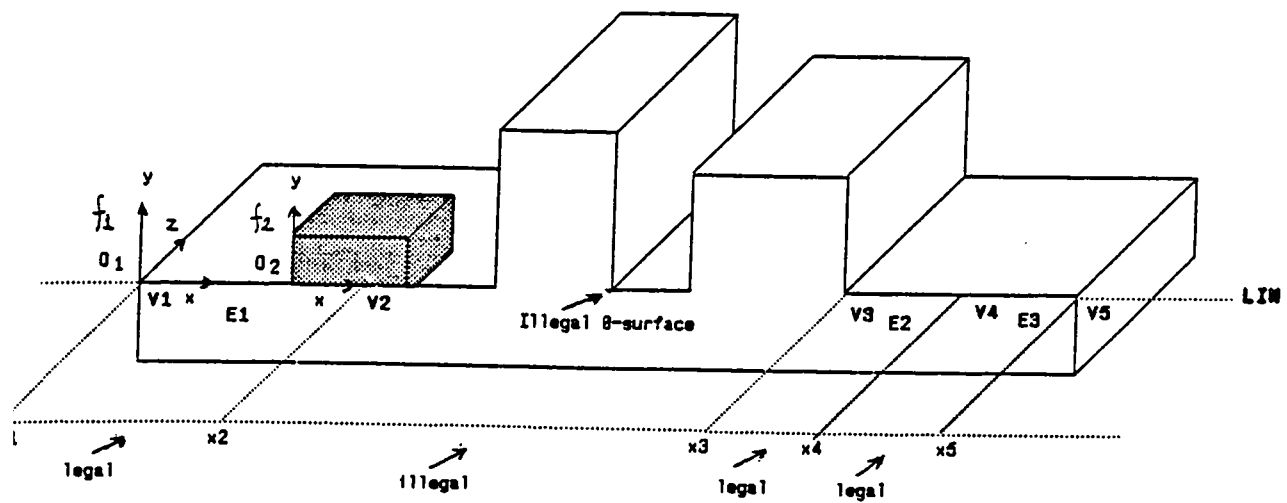
The relationship of the node is (LIN, f_1, f_2). The vertices V_i partition the 1-dimensional sur-face into edges E_1, E_2 and E_3. x_i is the value of the parameter of the relationship at some configuration. For example, $x_1 = 0$. The edges correspond to the legal intervals (x_i, x_{i+1}).

Figure 5-9: Partitioning a 1-dimensional surface into edges

5.2.4   Overview of the Constructed Model of the Contact Space

In the previous sections algorithms have been presented for generating (a) the surfaces the contact space and (b) the 1-skeleton of the contact space. Before proceeding to discuss how this model of the contact space can be used for planning a motion in contact let us briefly overview the constructed model.

The following geometric and topological information is known about each surface of the space.

1. The parametric equation of the surface by means of an associated spatial relationship between features of the moving object and the environment.

2. The dimension n of the surface which is equal to the degrees of freedom of the corresponding relationship.

3. The m-dimensional surfaces of the space which lie on the surface, where $m < n$ and the k-dimensional surfaces on which the surface lies, where $k > n$. This information can be found by examining the clauses of the descriptors of the surfaces, which describe the basic types of contact.

The following information is available for each vertex.

1. A location.

2. A descriptor and thus the generating 5 d.o.f. relationships.

3. The edges on whose boundary it lies.

The following information is available on each edge.

1. A spatial relationship giving the parametric equation of the edge.

2. A descriptor, and thus the generating 5 d.o.f. relationships.

3. The two bounding vertices.

This model of the space will be used for planning a motion in contact. For planning a motion we need to be able to identify the surface or face on which a location lies, that is, we need to be able to map locations to nodes of the graph of surfaces and the graph of the 1-skeleton.

The solution to this problem can be briefly sketched as follows: starting from vertices and working all the way up to 5-dimensional surfaces, check if the location satisfies the parametric equation of the surface by checking if a location satisfies the spatial relationship associated with the corresponding node in the graph of the surfaces.

5.3 Planning a Path in the Contact Space

In Chapter 3 it was argued that, in the proposed decomposition of the space, if the contact space is connected, then the vertices of the contact space will be edge-connected. The cases where some vertices might not be edge-connected were also indicated. If we assume that the vertices of the space are edge-connected then an algorithm for finding a path in the contact space can can be formulated as:

1. Find a path from the initial location to some vertex of the contact space

2. Find a path from the goal location to some vertex of the contact space

3. Find a path from the first vertex to the final vertex along the edges of the contact space.

In Section 5.3.1 the problem of finding a path along the edges of the space is considered. Section 5.3.2 deals with the problem of reducing the initial and final locations to vertices. Finally, in Section 5.3.3 the problem of transforming a path in contact space into a motion plan is considered.

5.3.1 Moving along the edges of the Contact Space

In Section 5.2.3 an algorithm was presented for finding the vertices and edges of the contact space and their adjacency relations. The problem that will be considered here is the following:

- Given two vertices v_1, v_2 of the contact space, find a path in the graph of the 1-skeleton of the contact space from v_1 to v_2.

The vertices of the graph $G(V, E)$ of the 1-skeleton of the contact space are the vertices of the contact space and the edges are the edges of the space. That is, two vertices of the graph $G(V, E)$ are adjacent, if the corresponding vertices of the contact space are the boundary vertices of an edge.

A number of graph searching algorithms can be used. In choosing an algorithm two points have to be considered: assigning a cost function to the edges of the graph and using the information we have about how close two vertices are.

A cost function can be assigned to the edges of the graph according to:

1. The type of motion required to traverse an edge. For example, a linear motion (an edge with a LIN relationship) should have a smaller cost than a motion where both a translation and a rotation are involved (e.g. an edge with a LAD relationship).

2. The length of the edge of the contact space, if it is an edge corresponding to a LIN relationship.

The above information can be extracted from the model of the contact space. In addition, from the model a measure of 'closeness' of vertices can be derived by examining how many contacts the two vertices share. In moving along the edges of the space, a shortest path corresponds to a motion where as few contacts as possible are broken or established. Therefore, the proximity of vertices can be established by examining the clauses, that is, the five degree of freedom relationships associated with the vertices. More specifically, let v_1, v_2, v_3 be vertices of the graph and C_1, C_2, C_3 be the sets of clauses of the vertices. Then v_1 is closer to v_2 than to v_3 if the cardinality of $C_1 \cap C_2$ is greater than the cardinality of $C_1 \cap C_3$. If the vertices are further apart, then the intersection of the sets of clauses would be empty, and thus no measure of proximity could be inferred.

The implemented algorithm is a depth-first algorithm, with no cost function assigned to the edges, which makes use of the above observation about the proximity of vertices: starting from a vertex v, the next vertex to be tried is an adjacent vertex which has as many common clauses with the goal vertex as possible.

The initial and final vertices can be specified either by a location or by a set of clauses, i.e. the five d.o.f. relationships that have to be satisfied. The path consists of a sequence of nodes which are vertices and edges of the contact space.
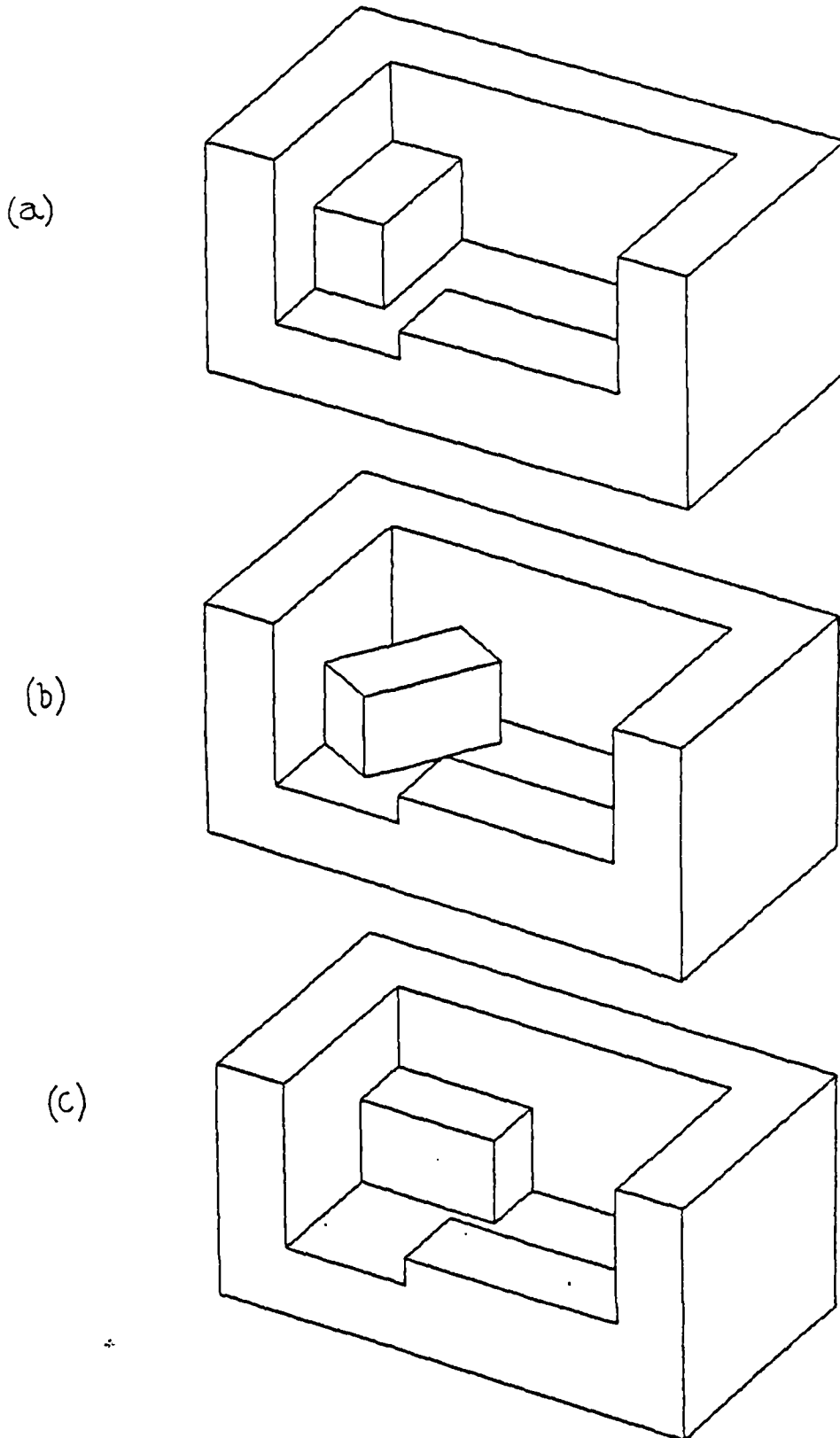
(a) the starting vertex, v_1; (b) a location on the edge e_1; e_1 corresponds to a LAD relationship;

(c) the vertex v_2.

Figure 5-10: A path along edges - 1

(a)

(b)
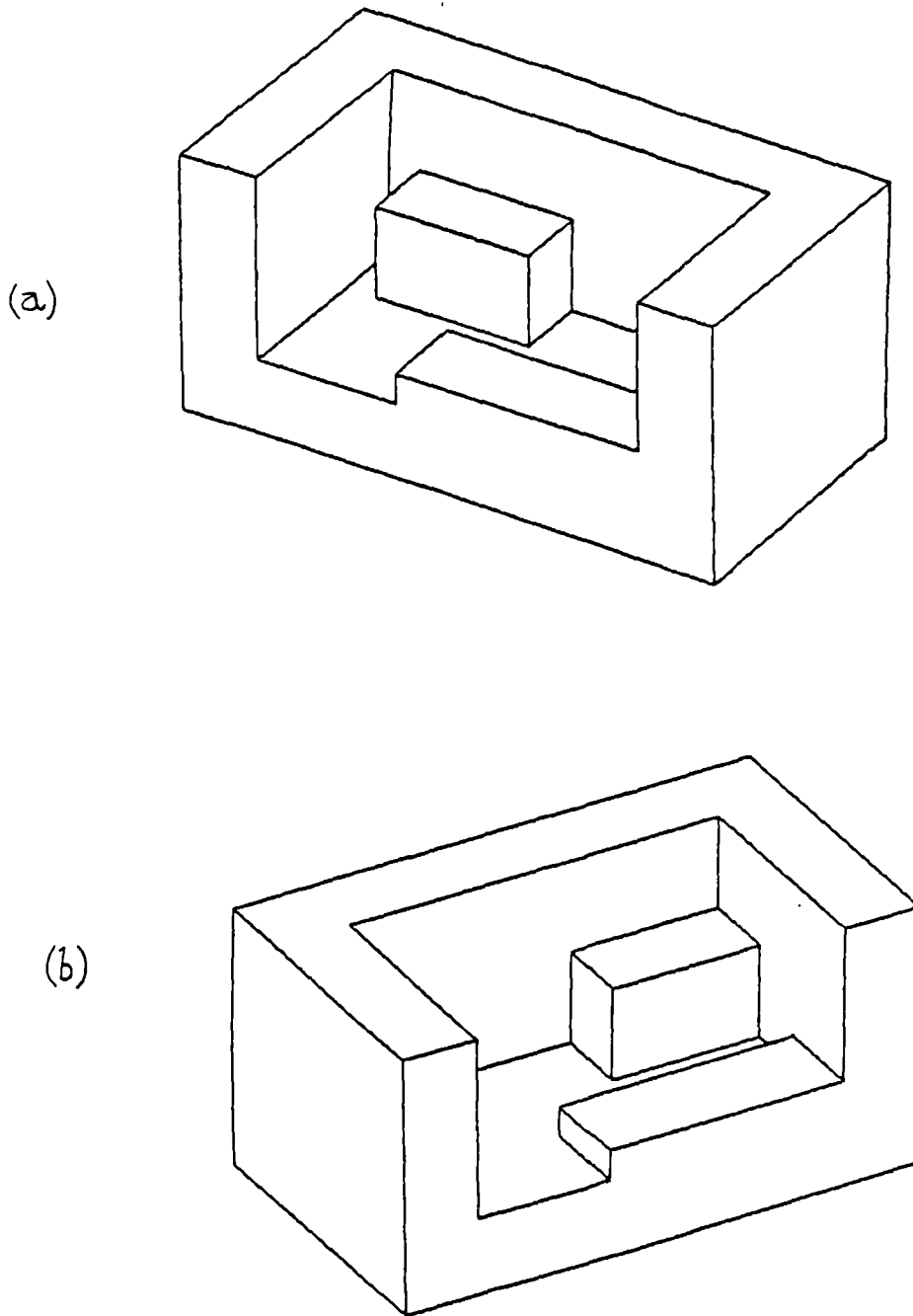
(a) a location on the edge e_2, an instance of a LIN relationship; (b) the final vertex, v_3.

Figure 5–11: A path along edges – 2

Figures 5–10 — 5–11 show a path found by the algorithm. The path consists of three vertices and two edges of the contact space. The first edge, corresponds to a LAD spatial relationship and the second edge corresponds to a LIN spatial relationship. The path is thus

$$v_1 \rightarrow e_1 \rightarrow v_2 \rightarrow e_2 \rightarrow v_3.$$

Figure 5–10 (b) shows an arbitrary location on the edge e_1. Similarly for Figure 5–11 (a). The information in the model about the path is shown in Figures 5–12 — 5–13.
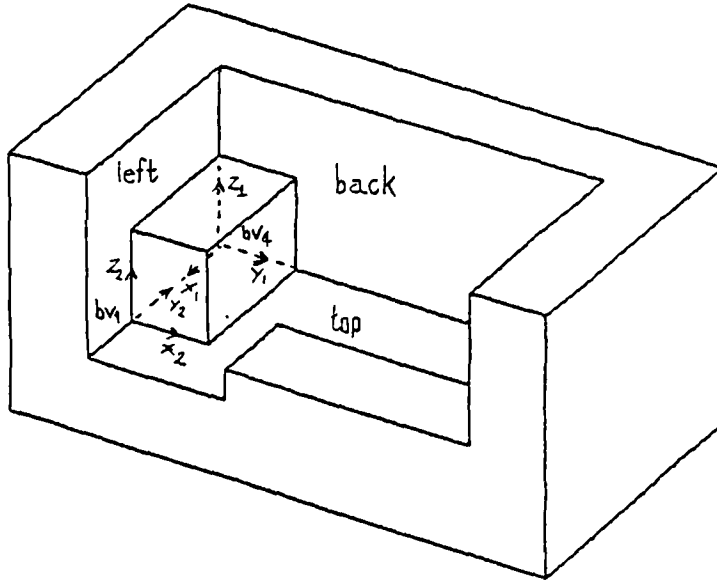
Vertex v_1

Location:

   0.0   1.0   0.0

 -1.0   0.0   0.0

   0.0   0.0   1.0

 10.0   0.0   0.0

Descriptor:

D =   (bv1,bot) $\wedge$ (bv2,bot) $\wedge$ (bv3,bot) $\wedge$ (bv4,bot) $\wedge$ (bv1,left) $\wedge$ (bv4,back)

        $\wedge$ (bv4,left)

Figure 5–12: A path along the edges: First vertex, v_1
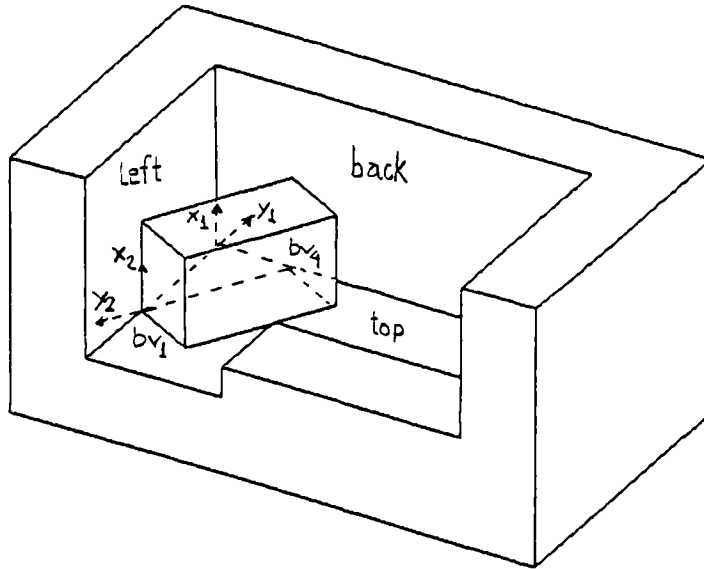
Edge e_1

Spatial Relationship: LAD

	Feature 1				Feature 2		
Dimension:	10.0			Dimension:	0.0		
Location:	0.0	0.0	1.0	Location:	0.0	0.0	1.0
	-1.0	0.0	1.0		0.0	-1.0	0.0
	0.0	-1.0	0.0		1.0	0.0	0.0
	0.0	0.0	0.0		0.0	0.0	0.0

Parameter interval: (1.5708,3.141593)

Bounding vertices: v_1, v_2

Descriptor:

D = (bv1,bot) $\wedge$ (bv2,bot) $\wedge$ (bv3,bot) $\wedge$ (bv4,bot) $\wedge$ (bv1,left) $\wedge$ (bv4,back)

Figure 5–13: A path along edges: First edge, e_1

5.3.2 Reaching a Vertex of the Contact Space

In the previous section I considered the problem of finding a sequence of edges which connect two vertices of the contact space. In this section I consider the problem of finding a path from an arbitrary location to some vertex. If this problem is solved then a path between any two arbitrary locations can be found.

If the graph of all the faces of the contact space had been constructed, this problem would again be a graph searching problem in the graph of faces. A motion plan would then correspond to a sequence of faces of the contact space. By choosing an arbitrary location in each face, we would have a description of the plan as a sequence of locations. However, for the reasons explained, only the graph of edges and vertices has been constructed. For higher dimensional faces, we only know the surfaces on which they lie. Let us, therefore, first consider the problem of finding a sequence of surfaces of the contact space which have to be traversed from some initial location to an arbitrary vertex of the contact space.

The problem is stated as follows:

> Given an initial contact state, find a sequence of surfaces of the contact space which have to be traversed in order to reach some[3] vertex of the contact space

The initial contact state can be described in the following ways:

1. As a location

2. As a set of basic types of contacts that have to be satisfied (5 d.o.f. relationships)

---

[3]Perhaps instead of reaching an arbitrary vertex we want to reach a vertex which is as close to the goal state as possible. The described algorithm can be easily adjusted so that it chooses the vertex which shares the most contacts with the goal state. Similarly for the goal state.

3. As an explicit spatial relationship.

The algorithm for this problem is summarised below:

1. Identify the surface of the contact space for the initial state and find the corresponding node n_i in the lattice of the contact space.

2. Find a vertex of the space whose clauses (5 d.o.f. relationships) includes the clauses of the node n_i. Let n_f be the node of the lattice corresponding to this vertex.

3. Find the chain in the graph from n_i to n_f.

However, a plan consisting only of a sequence of surfaces is not sufficient. The problems arise because a sequence of surfaces cannot be transformed into a sequence of locations, since there is no guarantee that an arbitrary location would correspond to a legal location or to a location for which the objects are in contact.

One solution to the problem would be of course to partition the surfaces into faces. Whether this can be done, however, remains an open issue.

Another way of overcoming the problem is, instead of describing a motion plan as a sequence of locations, to describe it only by means of directions of motions and conditions that have to be specified during the motion and at the end of the motion. This problem is related to the problem of transforming a path in the contact space into a sequence of compliant-guarded motions, which is the subject of the next section.

5.3.3   A Motion Plan as a Sequence of Compliant-guarded Motions

Up to now I have considered the problem of finding a sequence of surfaces of the contact space which have to be traversed in order to reach some final contact state from some initial contact state. The issue considered here is how from such a sequence of surfaces a motion plan can be derived.

Motion in contact involves motions which *comply* with constraints imposed by the geometry of the task. Such motions are called *compliant* motions. Also, motion in contact involves *guarded* motions, that is motions during which a new contact is established. As it has been argued in Section 2.3.2, a plan for a motion in contact cannot be adequately expressed as a sequence of locations. In the presence of uncertainty, force sensing is necessary so that a set of desired contacts are maintained during the motion. A motion plan can thus be expressed as a sequence of compliant-guarded motions. Each compliant-guarded motion can be specified by a template of the form: [Mason 81] [Paul and Shimano 76] [Will and Grossman 75]

MOVE TO [location] WITH [compliance] UNTIL [condition]

The compliance is specified by defining force and torque sensing in the coordinate system of the task, e.g.

FORCE X=0
FORCE Y=0
TORQUE X=0
TORQUE Z=0

The final condition can be specified by defining the force which has to be sensed in the coordinate system of the task, e.g.

FORCE Z=1

Mason [Mason 81] examined the relation between compliant motions and the surfaces of the contact space: motion on a surface of the contact space corresponds to a compliant motion, where freedom of motion occurs along the surface tangents, while freedom of force occurs along a surface normal. A guarded motion occurs when traversing the interface between two surfaces of different dimension. A path in the contact space can be decomposed thus into a sequence of compliant motions joined together by guarded motions.

From the above remarks it follows that a path from a location in some face of the space to a location on the boundary of the face can be translated into a compliant-guarded motion. A path along the edges of the contact space can be translated into a sequence of compliant-guarded motions, where each elementary motion corresponds to a compliant motion along the edge until the next vertex is reached.

In order to formulate the compliant-guarded motions the following information is necessary:

1. A goal location

2. The compliance axes

3. The force condition for stopping the motion

Let us examine through an example how the compliance axes can be inferred from the spatial relationship which is associated with a face of the contact space. Consider the case of planar motion (Figure 5-14). Then, the associated relationship would be an "against plane plane" (AGPP) relationship,

$$p_2 = f_2^{-1} twix(\theta) trans(0, y, z) f_1.$$

The free variables correspond to degrees of freedom of motion expressed in the coordinate system of the feature of the first object, that is in the coordinate system of f_1, as shown in the figure. Thus freedom of motion can occur along the y and z axes and about the x axis. The *non-compliant* axes are then the y-translation,
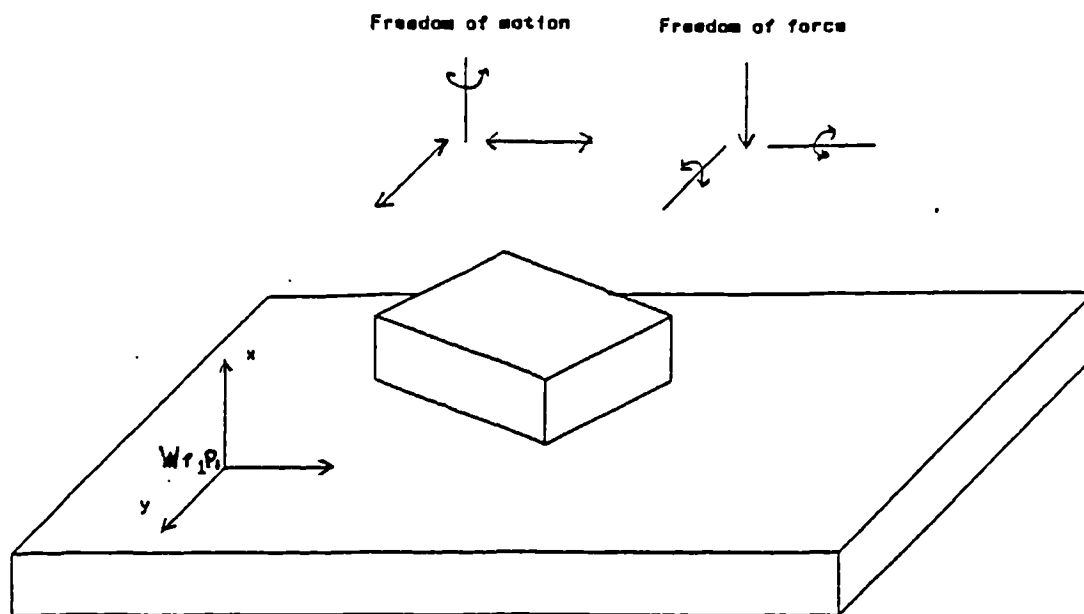
Figure 5–14: Planar motion: freedom of motion, freedom of force

z-translation and x-rotation.[4] The remaining axes, that is y-rotation, z-rotation and x-translation, are the *compliant* axes. As a result, the compliance for a planar motion can be specified by setting

FORCE X=0

TORQUE Y=0

TORQUE Z=0

From the above example it can be concluded that the compliant-axes of a motion can be inferred from the spatial relationship of the node in the model representing the traversed surface of the contact space.

In traversing the interface from a surface to a surface of lower dimension, one or more degrees of freedom of motion are lost. The condition for terminating a

---

[4]The 6 axes in 6-dimensional configuration space are: x-translation, y-translation , z-translation, x-rotation, y-rotation and z-rotation [Mason 81].
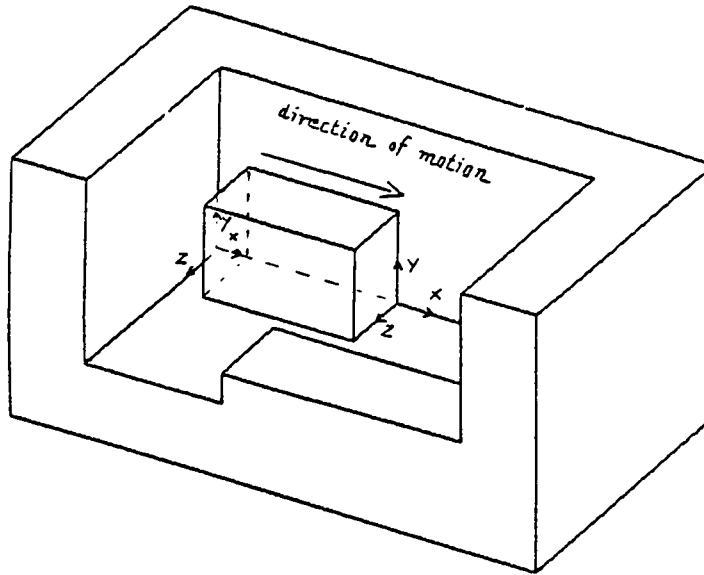
A location on the edge e_2. The motion on this edge is terminated when the right side of the block contacts the obstacle (vertex v_3). The terminating condition is FORCE X = -1.

Figure 5–15: Finding the terminating condition of a motion: LIN relationship

motion specifies for which axes the freedom of motion is eliminated. If there is only one degree of freedom of motion then the terminating condition is simply a statement of the form FORCE K=x (or TORQUE K=x), where K is the non-compliant axis of the motion and x is a number.

Consider as an example a motion from the second to the third vertex of the path found in Section 5.3.1 (Figure 5–15). This motion is a linear motion (LIN relationship) and can be stated as

MOVE TO P2 WITH
>   FORCE Y = 0
>   FORCE Z = 0
>   TORQUE X = 0
>   TORQUE Y = 0
>   TORQUE Z = 0

UNTIL FORCE X = -1

In the above discussion it has been assumed that the spatial relationships correspond to 'real' constraints, that is, the degrees of freedom of motion are physically constrained by actual contacts, as in the examples used above. However this is not necessarily the case, since 'imaginary' faces have been introduced for partitioning the contact space. An example can be found in Figure 3.30, page 90.

It can be concluded that the constructed model of the space can provide some of the information needed for transforming a path of the contact space into a sequence of compliant-guarded motions. There are still though some outstanding issues that have to be considered, but these fall outside the objective of this research.

5.4 Conclusions

In Chapter 5 we have been concerned with deriving a plan for a motion in contact. Working towards this objective the first stage has been to establish the necessary links between the spatial reasoning system and the decomposition of the contact space proposed in Chapter 3. The second stage has been to tackle the issue of implementation of the algorithm for decomposing the space into various dimensional surfaces using the spatial reasoning system. The decomposition of the space reduces the motion planning problem into a graph searching problem. The third and final stage has been to use the model of the space constructed previously to plan a motion in contact.

What has been accomplished is the construction and implementation of an algorithm which finds a path along the edges of the contact space, given an initial and final contact state which correspond either to vertices or edges. In addition, ways have been suggested for using the constructed model of the space in order to find a path between any initial and final states.

The problem of motion in three dimensions with rotations has the intrinsic difficulty that the constraints on the locations of the objects which are in contact are non-linear. What I have demonstrated in this chapter is that it is possible to deal with this problem without having to resort to the solution of algebraic equations. This approach has been made feasible by the use of the spatial reasoning system in the decomposition of the space.

Without doubt there are steps that could be taken that would enhance the efficiency of the algorithm. In particular, what needs to be considered is that the number of vertices of the contact space could be astronomical. Faced with this state of affairs, a possible solution would be to construct the contact space incrementally. This would entail considering a subset of the basic types of contact among the two objects, constructing the model of the contact space for that subset and then adding more relationships until a path can be found. As it stands, the algorithm can easily accommodate this process of building the model

of the space incrementally. The initial subset of constraints can be chosen so that it is the union of the constraints of the initial and the final states. Choosing which constraints to add could be problematic. One possible way of dealing with this is to examine the boundaries of the objects.

The algorithm, as it has been developed, is based on the assumption that the vertices of the space are edge–connected and, therefore, if there is a motion in contact between two vertices, then there is a motion in contact along the edges. Whether this property holds has been considered extensively in Chapter 3 and it was argued that it is only in exceptional cases that a vertex of the space wouldn't lie on the boundary of an edge. If the initial or final states correspond to such vertices then it would be necessary to find a path along a higher dimensional face.

This brings us to the issue of moving along higher dimensional faces of the space. As the implemented algorithm currently stands, the issue of finding this type of path has not been completely resolved. In Section 5.3.2 it has been suggested that perhaps this should be considered in conjunction with the question of transforming a path along the faces into a sequence of compliant-guarded motions and the execution of the motion. This latter question has only briefly been touched upon. In Section 5.3.3. I focussed on whether there is information available in the constructed model to effect this transformation and in what form of representation this information is to be found. In so far as it was shown that such information does exist and, moreover, in a form that is readily accessible, the prospects for effecting the transformation are encouraging.

Chapter 6

Conclusions and Further Research

This thesis has been concerned with the problem of moving a 3-dimensional polyhedral object while maintaining contact with a set of stationary obstacles. The issue of planning a motion in contact arises from the problems encountered in planning parts mating operations in the presence of uncertainty. In an environment about which there is incomplete information, planning parts mating entails planning also the use of force and touch sensors. In this context, motion in contact assumes paramount importance since it allows us to incorporate force information in the planning process.

What has been achieved in this research is the development and implementation of an algorithm which derives a motion plan as a sequence of contacts that have to be established during a motion from some initial to some final contact state. More specifically, the motion plan is a sequence of 0- and 1-dimensional faces of the contact space which have to be traversed.

In this chapter, I shall first recapitulate the salient features of the research and indicate the contributions made, and then I shall suggest areas for further research.

6.1 Summary

6.1.1 The Decomposition of the Contact Space

In the literature, the problem of motion planning has usually been formulated using the notion of configuration space. When dealing with the problem of motion in contact, what is of relevance is a subset of the configuration space, the contact space, i.e. the set of configurations for which the objects are in contact.

The contact space is composed of faces which intersect at lower dimensional faces which, in turn, intersect in still lower dimensional faces. The introduction of the notion of the contact space and its faces has allowed us to reduce the problem of planning a motion in contact to the problem of finding a path on the faces of the contact space.

Following this approach, my objective in this research has been defined as decomposing the contact space into faces and determining the boundary relations which hold between them. For a path-connected contact space, if the 0-dimensional faces (vertices) are shown to be connected by 1-dimensional faces (edges), then the motion planning problem is reduced to the problem of reaching some vertex and then traversing the edges of the space until the final vertex configuration is reached. Accordingly, I have sought to formulate the decomposition in terms such that the property of edge-connectedness obtains. In developing this formulation my starting point has been the work of Hopcroft and Wilfong [Hopcroft and Wilfong 84b].

Hopcroft and Wilfong have examined the case of one or more 2-dimensional polygonal moving objects amidst polygonal obstacles, where the only motions allowed are translations. They proved that for this case the edge-connectedness property holds. In this research, I have sought to extend their approach so as to accommodate the case of 3-dimensional polyhedral objects, when both translations and rotations are allowed. At the same time, I have restricted the number of moving objects to one.

As in [Lozano-Perez 83], [Donald 84], [Canny 84a], three basic types of contact among polyhedral objects are identified. Configurations for which such contacts hold lie on some 5-dimensional surface of the configuration space, which is defined by the constraint equation. The distinction between a configuration for which the objects are actually in contact and a configuration which satisfies the constraint equation of the contact has been formulated in [Hopcroft and Wilfong 84b]. Informally, I refer to this second class of configurations as configurations which satisfy a contact 'kinematically', where by 'kinematics' I refer to equations relating the locations of the objects.

Through a series of counter examples it has been shown that the edge-connectedness property does not hold when the decomposition in [Hopcroft and Wilfong 84b] is extended to 3-dimensional polyhedral objects and rotations. Faced with this problem, I have sought to modify the decomposition. In the modified decomposition I have proposed, the contact space is partitioned into faces by considering the 5-dimensional surfaces defined by the basic types of contact and their intersections. On the basis of the decomposition in [Hopcroft and Wilfong 84b], transition from a face to its boundary implies that one or more contacts have been established or broken. On the basis of the alternative decomposition I have proposed, this may be true but not necessarily so. All that it is required is that one or more additional constraints hold for a configuration in the boundary. The alternative decomposition thus yields a finer partition of the contact space. Through a series of examples, it was argued that this finer partition results in vertices that are edge-connected, with the possible exception of some pathological cases.

It should be noted that a similar approach for the decomposition of the contact space is proposed in [Yap ss], where the case of 2-dimensional polygonal objects which are allowed only to translate is considered.

At a theoretical level, the significance of the decomposition I have proposed is twofold. On the one hand, it has been shown that the vertices of the space will be, in general, edge-connected. This allows us to reduce the original search space – the contact space– to a 1-dimensional subspace –the 1-dimensional skeleton of the

contact space. On the other hand, the decomposition I have proposed is based on the distinction between configurations at which the objects are actually in contact and configurations which satisfy the 'kinematic' equation implied by the contact. This distinction allows us to treat separately the problem of 'kinematics', i.e. the problem of equations in the locations of the objects, from the problem of space occupancy.

6.1.2 The Algorithm for Planning a Motion in Contact

Having defined the decomposition, an algorithm was then presented for finding the faces of the contact space. The algorithm comprises two stages. In the first stage, the graph of the surfaces of various dimensions, on which the faces of the contact space lie, are found. In the second stage, the vertices and edges of the contact space are found.

In the first stage, the implemented algorithm makes use of a spatial reasoning system for finding the intersections of surfaces, while in the second stage it employs a solid modeller for checking physical interference. The introduction of the spatial reasoning system in my approach has been premised precisely upon the fact that it is possible to decouple theoretically the problems of 'kinematics' from the problems related to body occupancy.

Finally, an algorithm has been developed for finding a path between two vertices along the edges of the contact space.

6.1.3 The Spatial Reasoning System

Spatial relationships have been used in this research to represent the constraints on the relative locations of the objects imposed by contacts. Put in a different way, what this means is that a configuration for which two objects are in contact satisfies some spatial relationship between some features of the two objects.

The objective of the spatial reasoning system is to find the spatial relationship and the location of the features between which it holds, for any contact state.

This spatial relationship is arrived at by considering conjunctions of 5 degrees of freedom spatial relationships which describe the three basic types of contact.

The spatial reasoning system presented in this thesis is derived from the more general spatial reasoning system of the RAPT system [Corner, Ambler, and Popplestone 83], which I have adapted to suit the task at hand. The spatial reasoning system is based on a substitution table which holds rules for substituting a pair of spatial relationships by an equivalent, more constrained relationship. The table has entries for various combinations of types of relationships and geometric rules for finding the equivalent relationship and the location of the features between which it holds.

The construction of the table has been premised on the algebraic solution of location equations. The process of solving these equations has been guided by examining the geometric relations of the coordinate systems of the objects.

The substitution table derived in this research differs from the substitution table of the RAPT system primarily insofar as it includes a greater number of types of relationships and it examines an almost complete set of combinations. In contrast with the substitution table of RAPT, the derived table deals only with combinations with a 5 d.o.f. relationship, since this has been sufficient for this research. In addition to these differences, there are certain differences in the methodological approach. In particular, I have arrived at an iterative method for constructing the substitution table. Moreover, insofar as I have developed an algebraic method for finding the equivalent relationship, it became possible to examine carefully all the different cases that may arise from a combination of certain types of relationships.

6.2 Significance and Originality

In this research I have developed and implemented an algorithm for finding a path for a motion in contact along the edges of the contact space. The principal significance of the research lies in the fact that the algorithm deals with 3-dimensional polyhedral objects which are allowed both to translate and to rotate. It is also significant that the algorithm has been developed without resorting to linear approximations of the constraints resulting from rotations.

In addition, several other methodological contributions have arisen from the research:

1. I have proposed the enlargement of the decomposition in [Hopcroft and Wilfong 84b] for the case of 3-dimensional objects which can rotate and translate. This alternative decomposition makes a clear distinction between the 'kinematics' of contacts and body occupancy. It also overcomes, in most cases, the problem of vertices which are not edge-connected.

2. I have introduced the use of a spatial reasoning system in motion planning. With the use of the particular reasoning system, it is not necessary to resort to the solution of algebraic equations while planning a motion. General solutions of the algebraic equations have been formulated in terms of geometric relations between coordinate systems embedded in objects and their features. This approach differs significantly from the approach in [Donald 84], where an algebra system is employed.

3. I have extended the spatial reasoning system of the RAPT system to the point that it can handle a significantly more extensive range of spatial relationships, though still not complete.

6.3   Suggestions for Further Research

In this section I discuss the limitations of the work reported in this thesis and I suggest some possible ways to overcome them. In addition, I indicate topics for further research related to this thesis. In particular, I suggest ways in which this work can be extended towards a practical system which can be used for parts mating operations.

The topics presented in this section examine questions which can be classified into four main categories.

- Implications of the restriction on the domain of the problem: What happens if non-polyhedral objects are considered?

- Direct extensions: Is it necessary to construct the whole graph of the contact space? What are the implications of an incomplete Substitution Table?

- Introducing force sensing: How can optimal paths be found? How can a plan for a motion from some initial contact to some vertex be formulated? How can a sequence of compliant motions be derived?

- Other possible directions: Can the algorithms developed be used for planning a motion in free space?

6.3.1   Non-polyhedral objects

In the research reported in this thesis only polyhedral objects have been considered. The problem of examining the motion in contact of non-polyhedral objects arises from the fact that it is not easy to characterise the possible types of contact and the corresponding d.o.f. of motion. Since one can use polyhedral approximations to model some class of 'real' objects, it is theoretically possible to use the approach described here. There are two main problems in doing this:

(a) the graph of the contact space would consist of a very large number of nodes, since the number of edges and vertices of the objects would be very large; (b) force data could not be used in following a non-real edge, e.g. some edge on the curved surface of a cylinder. The first problem could be partially overcome by building the contact space incrementally, as it is described in Section 6.3.2. A possible solution to the second problem would be to avoid, if possible, curved surfaces.

6.3.2 Building the 1-skeleton Incrementally

It was pointed out in Section 5.4 that the number of vertices of the contact space could reach an astronomical value. This, however, does not present an insurmountable problem, since it is not necessary to construct the entire graph of vertices and edges. The graph can be built incrementally by considering initially a subset of all the possible 5 d.o.f. relationships and then adding more 5 d.o.f. relationships to this subset until a path can be found. The initial set of relationships should definitely include the relationships which are present in the initial and final states. If a sequence of contact states from some initial to some final state cannot be found, then the graph of the contact space can be extended, by adding more 5 d.o.f. relationships. Deciding which relationships to add means deciding which features are likely to be brought into contact. The process of enlarging the set of relationships involves, thus, the examination of the features in the boundaries of the objects. Clearly, the features which are more likely to interact would be features in the vicinity of the features already included in the set. A method for choosing features and relationships to be added for enlarging the graph of the contact space could form the subject of further research.

6.3.3 Extension of the Substitution Table

The substitution table of the spatial reasoning system developed in this research is not complete: firstly, it cannot deal with all possible types of relationships,

and secondly, it cannot deal with all possible combinations of relationships. In particular, it does not deal extensively with relationships for which the degrees of freedom are interdependent. Such relationships are not only computationally difficult, but they are also practically difficult, in the sense that deciding the force sensing required to execute such a compliant motion would be a complicated procedure. For this reason, in planning a motion in contact, it would be preferable to avoid contact states corresponding to these type of relationships. This could be achieved by leaving entries out of the table, but this is not an entirely satisfactory approach. Ideally, the table would be completed, so that a path would always be found if one existed. The criteria described in Section 6.3.4 would then be used to select the best path, so that awkward contact states would be avoided.

The substitution table presented in this thesis has been developed with a view to be used in planning a motion in contact. For this purposes, it has been sufficient to consider combinations of relationships with 5 d.o.f. relationships. While this makes it possible for all types of contact to be represented, it might sometimes be convenient to have an extended table which directly coded combinations of more constrained spatial relationships. RAPT itself uses an extended table so that the programmer can easily describe contact states such as one plane face being against another. However, the extended substitution table is incomplete. It could be completed by using the methods presented in Sections 4.4–4.6.

6.3.4   Choosing a Path Along Edges

The algorithm for finding a path along the edges of the contact space has not considered the problem of optimality. The search for an optimal path can be accomplished if cost functions are assigned to the edges of the graph (i.e. the edges of the contact space). Some criteria for assigning cost functions have been suggested in Section 5.3.1. In particular, two types of optimality criteria have been suggested. The first type is related to the length of the edge and the second

type is related to the cost of force sensing that would be necessary in order to execute the motion. The cost of force sensing clearly depends on the type of motion required to traverse the edge.

In order to derive a more efficient plan for a motion in contact, further research is needed in the following areas: (a) deciding which optimality criteria should be applied; (b) deciding on an appropriate metric which can be used in rotation space; (c) evaluating the cost of force sensing required for a motion.

6.3.5   Moving on Higher-dimensional Surfaces

In this research I have sought to decompose the space in such a way that the vertices of the space will be edge-connected. This method of decomposition has been chosen with a view to avoiding motion on higher dimensional surfaces. There is still, however, the problem of reaching some vertex or edge from the initial and final states. The problem has been examined in section 5.3.2, where a method for achieving this transition was outlined.

The problem of moving on higher dimensional surfaces would be solved if the surfaces of the contact space were partitioned into faces. Developing an algorithm for partitioning the surfaces into connected regions is not a trivial problem, even if the boundaries of the regions can be identified. Moreover, it is not certain that this would be the most efficient way for tackling this problem, since only the transition from the initial state to a vertex has to be considered.

The approach I have outlined involves deciding on the type of motion and on the direction of motion, and then executing the motion until some additional contact is established. For this reason, I believe that the issue of reaching some vertex should be considered in relation with the issue of deriving a sequence of compliant-guarded motions.

6.3.6 Deriving a Motion Plan as a Sequence of Compliant-guarded Motions

The problem of transforming a sequence of contact states into a sequence of compliant-guarded motions has been discussed in Section 5.3.3. It was pointed out that this transformation can be achieved by considering the types of spatial relationships and the locations of the features. There are still, however, a number of outstanding issues. For example, it is unclear how the compliant axes would be defined in the case when the degrees of freedom of motion are interdependent. Furthermore, it is not clear what type of sensing should be used in the case of 'imaginary' edges, that is, in the case where the d.o.f. of motion are not actually constrained by physical contact.

A broad approach for deriving a sequence of compliant-guarded motions has been suggested in Section 5.3.3. The extension of this research in that direction would be the first step towards using the system presented in this thesis for parts mating operations.

6.3.7 Extension to motion in free space

Research in collision-free motion ([Lozano-Perez 83], [Donald 84]) has considered the boundaries of the grown obstacles (the contact space) in order to decompose the free space into connected regions. In this research, a model of the contact space has been constructed. It would be interesting to consider whether the model of the contact space developed in this thesis can be used to decompose the free space into connected regions, and then to investigate planning a motion in free space using the intersections of the surfaces of the contact space.

6.4 Conclusion

This thesis has suggested a method for planning a path for a motion in contact as a sequence of contact states, and presented a partial solution to the parts mating problem in the case of polyhedral objects. Further work along the lines suggested in this chapter should make it possible for a practical solution to the problem to be reached.

Bibliography

[Agoston 76] M.K. Agoston. *Algebraic Topology: a first cource.* Marcel Dekker, Inc., New York, 1976.

[Ambler and Popplestone 75] A.P. Ambler and R.J. Popplestone. Inferring the positions of bodies from specified spatial relationships. *Artificial Intelligence,* 6(2):157–174, 1975.

[Arai 81] T. Arai. *Private communication.*

[Barrett, Ramsay, and Sloman 85] R. Barrett, A. Ramsay, and A. Sloman. *POP-11: A Practical Language for Artificial Intelligence.* Ellis Horwood Limited, 1985.

[Birkhoff and MacLane 65] G. Birkhoff and S. MacLane. *A Survey of Modern Algebra.* Macmillan, New York and London, 1965.

[Boyse 79] J.W. Boyse. Interference detection among solids and surfaces. *Communications of the ACM,* 22(1):3–9, 1979.

[Brady 82a] M. Brady. Trajectory planning. In M. Brady et al., editors, *Robot Motion: Planning and Control,* The MIT Press, Cambridge, Massachusetts, 1982.

[Brady 82b] M. Brady. Parts description and acquisition using vision. In *Proceedings of SPIE,* May 1982.

225

[Brady et al. 82] M. Brady, J. M. Hollerbach, T. Johnson, T. Lozano-Perez, and M. Mason, editors. *Robot Motion: Planning and Control.* The MIT press, Cambridge, Massachusetts, 1982.

[Brooks 82] R.A. Brooks. Symbolic error analysis and robot planning. *International Journal of Robotics Research*, 1(4), December 1982.

[Brooks 83a] R.A. Brooks. Planning collision free motions for pick and place operations. *International Journal of Robotics Research*, 2(4):19–44, 1983.

[Brooks 83b] R.A. Brooks. Solving the find-path problem by representing free space as generalised cones. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-13:190–197, 1983.

[Brooks and Lozano-Perez 83] R.A. Brooks and T. Lozano-Perez. A subdivision algorithm in configuration space for findpath with rotation. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, Karlsruhe, Germany, December 1983.

[Brown 81] C.M. Brown. Some mathematical and representational aspects of solid modelling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(4):444–453, July 1981.

[Cameron 84] S. Cameron. *Modelling Solids in Motion.* PhD thesis, Dept. of Artificial Intelligence, University of Edinburgh, 1984.

[Canny 84a] J. Canny. *Collision Detection for Moving Polyhedra.* AI Memo 806, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, December 1984.

[Canny 84b] J. Canny. On detecting collisions between polyhedra. In *Proceedings of the 6th European Conference on Artificial Intelligence*, Pisa, Italy, September 1984.

[Canny 85] J. Canny. A voronoi method for the piano movers' problem. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 530–535, St. Luis, Missouri, March 1985.

[Chatila 82] R. Chatila. Path planning and environment learning in a mobile robot system. In *Proceedings of the European conference on Artificial Intelligence*, Orsay, France, 1982.

[Cooke and Finney 67] G.E. Cooke and R.L. Finney. *Homology of Cell Complexes*. Princeton University Press, New Jersey, 1967.

[Corner, Ambler, and Popplestone 83] D.F. Corner, A.P. Ambler, and R.J. Popplestone. Reasoning about the spatial relationships derived from a RAPT program for describing assembly by robot. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, Karlsruhe, 1983.

[Donald 84] B.R. Donald. *Motion Planning with Six Degrees of Freedom*. Technical Report 791, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, May 1984.

[Drake 77] S.H. Drake. *Using Compliance in Lieu of Sensory Feedback for Automatic Assembly*. PhD thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, 1977.

[Drysdale 79] R.L. Drysdale. *Generalised Voronoi Diagrams and Geometric Searching*. Technical Report, Department of Computer Science, Stanford University, 1979.

[Dufay and Latombe 84] P. Dufay and J.C. Latombe. An approach to automatic robot planning based on inductive learning. *International Journal of Robotics Research*, 3(4), 1984.

[Erdmann 84] M. Erdmann. *On Motion Planning with Uncertainty*. Technical Report TR-810, Arificial Intelligence Laboratory, Massachusetts Institute of Technology, 1984.

[Fleming 85] A. Fleming. Analysis of uncertainty in a structure of parts. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pages 1113–1115, Los Angeles, California, 1985.

[Gini, Gini, and Somalvico 80] G. Gini, M. Gini, and M. Somalvico. Program abstraction and error correction in intelligent robots. In *Proceedings of the 10th International Symposium on Industrial Robots*, pages 101–108, Milan, Italy, March 1980.

[Goto, Inoyama, and Takeyasu 80] T. Goto, T. Inoyama, and K. Takeyasu. Precise insert operation by tactile controlled robot. *IEEE Transactions on Systems, Man, Cybernetics*, SMC-10(1):19–25, 1980.

[Gouzenes 84] L. Gouzenes. Strategies for solving collision-free trajectories problems for mobile and manipulator robots. *International Journal of Robotics Research*, 3(4), 1984.

[Hanafusa and Asada 77] H. Hanafusa and H. Asada. Stable prehension of objects by the robot hand with elastic fingers. In *Proceedings of the 7th International Symposium on Industrial Robots*, pages 361–368, Tokyo,, October 1977.

[Hopcroft, Joseph, and Whitesides 84] J.E. Hopcroft, D. Joseph, and S. Whitesides. Movement problems for 2-dimensional linkages. *SIAM Journal on Computing*, 13(3), August 1984.

[Hopcroft and Wilfong 84a] J.E. Hopcroft and G.T. Wilfong. On the motion of objects in contact. In *Proceedings of the 2nd International Symposium on Robotics Research*, pages 81–90, Kyoto, Japan, 1984.

[Hopcroft and Wilfong 84b] J.E. Hopcroft and G.T. Wilfong. *Reducing Multiple Object Motion Planning to graph searching*. Technical Report TR 84-616, Computer Science Department, Cornell University, July 1984.

[Hopcroft, Schwartz, and Sharir 84] J.E. Hopcroft, J.T. Schwartz, and M. Sharir. On the complexity of motion planning for multiple independent objects; PSPACE-hardness of the 'warehouseman's problem'. *The International Journal of Robotics Research*, 3(4), 1984.

[Inoue 74] H. Inoue. *Force Feedback in Precise Assembly Tasks.* Technical Report AIM-308, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, August 1974.

[Koutsou 85] A. Koutsou. A geometric reasoning system for moving an object while maintaining contact with others. In *Proceedings of the 1st ACM Symposium on Computational Geometry*, Baltimore, June 1985.

[Latombe and Mazer 81] J.C. Latombe and E. Mazer. LM: a high-level programming language for controlling manipulator robots. In *Proceedings of the 11th International Symposium on Industrial Robots*, Tokyo, Japan, October 1981.

[Lieberman and Wesley 77] L.I. Lieberman and M.A. Wesley. AUTOPASS: an automatic programming system for computer controlled mechanical assembly. *IBM Journal of Research and Development*, 21(4):321–333, July 1977.

[Lozano-Perez 76] T. Lozano-Perez. *The Design of a Mechanical Assembly System.* Technical Report AI TR 397, Arificial Intelligence Laboratory, Massachusetts Institute of Technology, 1976.

[Lozano-Perez 81] T. Lozano-Perez. Automatic planning of manipulator transfer movements. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-11(10):681–689, 1981.

[Lozano-Perez 82] T. Lozano-Perez. Task planning. In M. Brady et al., editors, *Robot Motion: Planning and Control*, The MIT Press,Cambridge, Massachusetts, 1982.

[Lozano-Perez 83] T. Lozano-Perez. Spatial planning: a configuration space approach. *IEEE Transactions on Computers*, C-32(2), February 1983.

[Lozano-Perez and Wesley 79] T. Lozano-Perez and M.A. Wesley. An algorithm for planning collision-free paths among polyhedral objects. *Communications of the ACM*, 22(10):560–570, October 1979.

[Lozano-Perez, Mason, and Taylor 84] T. Lozano-Perez, M.T. Mason, and R.H. Taylor. Automatic synthesis of fine-motion strategies for robots. *International Journal of Robotics Research*, 3(1), 1984.

[Mason 81] M.T. Mason. Compliance and force control for computer controlled manipulators. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-11(6):418–432, 1981.

[Mason 82a] M.T. Mason. Compliance. In M. Brady et al., editors, *Robot Motion: Planning and Control*, The MIT Press, Cambridge, Massachusetts, 1982.

[Mason 82b] M.T. Mason. *Manipulator grasping and pushing operations*. PhD Thesis, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1982.

[Mason 83] M.T. Mason. *Automatic Planning of Fine Motions: Correctness and Completeness*. Technical Report CMU-RI-TR-83-18, The Robotics Institute, Carnegie-Mellon University, December 1983.

[Massey 67] W.S. Massey. *Algebraic Topology: An introduction*. Springer-Verlag, York, 1967.

[Massey 78] W.S. Massey. *Homology and Cohomology Theory*. Marcel Dekker, Inc., New York, 78.

[Mazer 83] E. Mazer. Geometric programming of assembly robots (LM-GEO). In *Proceedings of the International Meeting on Advanced Software in Robotics*, Liege, France, May 1983.

[Meyer 81] J. Meyer. An emulation system for programmable sensory robots. *IBM Journal of Research and Development*, 25(6), November 1981.

[Moravec 80] H.P. Moravec. *Obstacle Avoidance and Navigation in Real World by a Seeing Robot Rover*. AIM-340, Artificial Intelligence Laboratory Stanford University, September 1980.

[Mujtaba and Goldman 79] M.S. Mujtaba and R. Goldman. *AL User's Manual*. Technical Report STAN-CS-79-718, Computer Science Department, Stanford University, January 1979.

[Nevins and Whitney 74] J.L. Nevins and D.E. Whitney. The force vector assembler concept. In *Proceedings of the 1st CISM-IFTOMM Symposium on Theory and Practice of Robots and Manipulators*, 1974.

[Nillson 69] N.J. Nillson. A mobile automaton: an application of artificial intelligence techniques. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 509–520, 1969.

[O'Dunlaing and Yap 85] C. O'Dunlaing and C.K. Yap. A retraction method for planning the motion of a disc. *Journal of Algorithms*, 6:104–111, 1985.

[O'Dunlaing, Sharir, and Yap 83] C. O'Dunlaing, M. Sharir, and C.K. Yap. Retraction: a new approach to motion planning. In *Proceedings of the 15th Symposium on the Foundations of Computer Science*, pages 207–220, 1983.

[O'Dunlaing, Sharir, and Yap 84] C. O'Dunlaing, M. Sharir, and C.K. Yap. *Generalized Voronoi Diagrams for a Ladder: I. Topological Considerations*. Technical Report 139, Computer Science Department, NYU-Courant Institute, November 1984.

[Ohwovoriole 80] M.S. Ohwovoriole. *An Extension of Screw Theory and its Application to the Automation of Industrial Assemblies*. PhD thesis, Department o Computer Science, Stanford University, 1980.

[Paul 81] R.P. Paul. *Robot Manipulators: Mathematics, Programming and Control.* The MIT press, Cambridge, Massachusetts, 1981.

[Paul and Shimano 76] R. Paul and B. Shimano. Compliance and control. In *Proceedings of Joint Automatic Control Conference*, pages 694–699, San Francisco, 1976.

[Pieper 68] D.L. Pieper. *The Kinematics of Manipulators under Computer Control.* PhD thesis, Artificial Intelligence Laboratory, Stanford University, 1968.

[Popplestone and Ambler 83] R.J. Popplestone and A.P. Ambler. A language for specifying robot manipulations. In A. Pugh, editor, *Robotic Technology*, pages 125–141, Peter Peregrinus Ltd., London, UK, 1983.

[Popplestone, Ambler, and Bellos 78] R.J. Popplestone, A.P. Ambler, and I. Bellos. RAPT, a language for describing assemblies. *Industrial Robot*, 5(3):131–137, 1978.

[Popplestone, Ambler, and Bellos 80] R.J. Popplestone, A.P. Ambler, and I. Bellos. An interpreter for a language for describing assemblies. *Artificial Intelligence*, 14(1):79–107, 1980.

[Raibert and Craig 81] M.H. Raibert and J.J. Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement ,Control*, 102:126–133, June 1981.

[Reif 79] J. Reif. On the movers problem. In *Proceedings of the 20th IEEE Symposium on the Foundations of Computer Science*, pages 421–427, 1979.

[Requicha 77] A.A.C. Requicha. *Mathematical Models of Rigid Solid Objects.* Technical Memorandum 28, Production Automation Project, University of Rochester, November 1977.

[Requicha 83] A.A.C. Requicha. *Towards a Theory of Geometric Tolerancing.* Technical Memorandum 40, Production Automation Project, University of Rochester, March 1983.

[Rowat 79] P.F. Rowat. *Representing Spatial Experience and Solving Spatial Problems in a Simulated Robot Environment.* PhD thesis, 1979.

[Salisbury 80] J.K. Salisbury. Active stiffness control of a manipulator in cartesian coordinates. In *19th IEEE Conference on Decision and Control*, Albuquerque, New Mexico, November 1980.

[Schwartz and Sharir 83a] J.T. Schwartz and M. Sharir. On the piano movers' problem: I. the case of a rigid polygonal body moving amidst polygonal barriers. *Communications on Pure and Applied Mathematics*, 36:345–398, 1983.

[Schwartz and Sharir 83b] J.T. Schwartz and M. Sharir. On the piano movers' problem: II. general techniques for computing topological properties of real algebraic manifolds. *Advanced Applied Mathematics*, 4:298–351, 1983.

[Schwartz and Sharir 83c] J.T. Schwartz and M. Sharir. On the piano movers' problem: III. coordinating the motion of several indpendent bodies: the special case of circular bodies moving amidst polygonal barriers. *International Journal of Robotics Research*, 2(3):46–75, 1983.

[Schwartz and Sharir 84] J.T. Schwartz and M. Sharir. On the piano movers' problem: V. the case of a rod moving in three-dimensional space amidst polyhedral obstacles. *Communications on Pure and Applied Mathematics*, 37:815–848, 1984.

[Simunovic 75] S.N. Simunovic. Force information in assembly processes. In *Proceedings of the 5th International Symposium on Industrial Robots*, pages 415–431, Chicago, Illinois, September 1975.

[Simunovic 79] S.N. Simunovic. *An Information Approach to Parts Mating.* PhD thesis, Department of Electrical Engineering, Massachusetts Institute of Technology, 1979.

[Sharir and Ariel-Sheffi 84] M. Sharir and E. Ariel-Sheffi. On the piano movers' problem: IV. various decomposable two-dimensional motion-planning problems. *Communications on Pure Applied Mathematics*, 37:479–493, 1984.

[Sharir and Leven 85] M. Sharir and D. Leven. An efficient and simple motion planning algorithm for a ladder moving in two dimensional space amidst polygonal barriers. In *Proceedings of the 1st Symposium on Computational Geometry*, Baltimore, June 1985.

[Spirakis and Yap 84] P. Spirakis and C.K. Yap. Strong NP-hardness of moving many discs. *Information Processing Letters*, 19:55–59, 1984.

[Taylor 76] R.H. Taylor. *A Synthesis of Manipulator Control Programs from Task Level Specifications*. Technical Report AIM-282, Artificial Intelligence Laboratory, Stanford University, July 1976.

[Taylor, Summers, and Meyer 82] R. Taylor, P. Summers, and J. Meyer. AML: a manufacturing language. *International Journal of Robotics Research*, 1(3), July 1982.

[Udupa 77a] S.M. Udupa. *Collision Detection and Avoidance in Computer Controlled Manipulators*. PhD thesis, Department of Electrical Engineering, California Institute of Technology, 1977.

[Udupa 77b] S.M. Udupa. Collision detection and avoidance in computer controlled manipulators. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, pages 737–748, Massachusetts Institute of Technology, 1977.

[Whitney 82] D.E. Whitney. Quasi-static assembly of compliantly supported rigid parts. *Journal of Dynamic Systems, Measurement and Control*, 104(65), March 1982.

[Whitney and Nevins 79] D.E. Whitney and J.L. Nevins. What is the remote centre compliance and what can it do. In *Proceedings of the 9th International Symposium on Industrial Robots*, pages 135–149, Washington, March 1979.

[Widdoes 74] C. Widdoes. *A Heuristic Collision Avoidance for the Stanford Robot Arm*. Artificial Intelligence Laboratory, Stanford University, 1974.

[Will and Grossman 75] P.M. Will and D.D. Grossman. An experimental system for computer controlled mechanical assembly. *IEEE Transactions on Commputers*, C-24(9):879–888, September 1975.

[Wingham 77] M. Wingham. *Planning how to Grasp Objects in a Cluttered Environment*. M. Phil. Thesis, Department of Artificial Intelligence, University of Edinburgh, 1977.

[Yap 84a] C. K. Yap. *Coordinating the Motion of Several Discs*. Technical Report 16, Robotics Lab., NYU-Courant Institute, February 1984.

[Yap 84b] C. K. Yap. *How to Move a Chair through a Door*. Technical Report , Robotics Lab., NYU-Courant Institute, December 1984.

[Yap 85] C. K. Yap. *An O(nlogn) algorithm for the Voronoi diagram of a set of simple curve segments*. Technical Report 43, Robotics Lab., NYU-Courant Institute, May 1985.

[Yap ss] C.K. Yap. Algorithmic motion planning. In J.T. Schwartz and C.K. Yap, editors, *Advances in Robotics: Vol. 1*, Lawrence Erlsbaum Associates, in press.

Appendix A

The Substitution Table

The conventions used in the following tables are summarised below.

$$
1. \qquad pos2pos1^{-1} = \begin{pmatrix} f_{11} & f_{12} & f_{13} & 0 \\ f_{21} & f_{22} & f_{23} & 0 \\ f_{31} & f_{32} & f_{33} & 0 \\ l_1 & l_2 & l_3 & 1 \end{pmatrix}.
$$

2. E_i is a relationship between features with locations a_i and b_i of objects A and B respectively.

$$
3. \qquad F_1 = a_2 a_1^{-1} = \begin{pmatrix} f_{11} & f_{12} & f_{13} & 0 \\ f_{21} & f_{22} & f_{23} & 0 \\ f_{31} & f_{32} & f_{33} & 0 \\ l_{11} & l_{12} & l_{13} & 1 \end{pmatrix}.
$$

$$
4. \qquad F_2 = b_2 b_1^{-1} = \begin{pmatrix} f'_{11} & f'_{12} & f'_{13} & 0 \\ f'_{21} & f'_{22} & f'_{23} & 0 \\ f'_{31} & f'_{32} & f'_{33} & 0 \\ l_{21} & l_{22} & l_{23} & 1 \end{pmatrix}.
$$

A.1 Geometric functions

Tables A-1 — A-4 present a set of geometry functions which are used in the substitution table for checking the conditions under which two relationships can be substituted by an equivalent relationship. Both the geometric and the algebraic interpretation of the functions are presented.

Function	Geometric Interpretation	Algebraic Interpretation
d-oo	distance from origin of pos1 to origin of pos2	$\sqrt{l_1^2 + l_2^2 + l_3^2}$
d-op	distance from origin of pos1 to x=0 plane of pos2	$-(f_{11}l_1 + f_{12}l_2 + f_{13}l_3)$
d-xo	distance from origin of pos2 to x-axis of pos1	$\sqrt{l_2^2 + l_3^2}$
d-oi	distance from origin of pos1 to intersection of x=0 planes of pos1 and pos2	$\left\|(f_{11}l_1 + f_{12}l_2 + f_{13}l_3)/\sqrt{f_{12}^2 + f_{13}^2}\right\|$
d-xi	distance from origin of pos1 to intersection of x-axis of pos1 with x=0 plane of pos2	$\|(f_{11}l_1 + f_{12}l_2 + f_{13}l_3)/f_{11}\|$
d-yi	distance from origin of pos1 to intersection of y-axis of pos1 with x=0 plane of pos2	$\|(f_{11}l_1 + f_{12}l_2 + f_{13}l_3)/f_{12}\|$
d-cm	length of common perpendicular between x axes of pos1 and pos2	$\left\|(f_{12}l_3 - f_{13}l_2)/\sqrt{f_{12}^2 + f_{13}^2}\right\|$

Table A-1: Geometry Functions

Name	Geometric Interpretation	Algebraic Interpretation
x-a-par †	x axes of body A are parallel	$f_{11} = \pm 1$
x-a-eq	x axes of body A are equal	$f_{11} = 1$
x-a-collin †	x axes of body A are collinear	$f_{11} = \pm 1 \wedge l_{12} = l_{13} = 0$
x-a-perp †	x axes of body A are perpendicular	$f_{11} = 0$
x-a-coplan †	x axes of body A are coplanar	$f_{11} = \pm 1 \vee f_{12}l_{13} = f_{13}l_{12}$
y-a-par †	y axes of body A are parallel	$f_{22} = \pm 1$
y-a-perp †	y axes of body A are perpendicular	$f_{22} = 0$
is-0-a12 †	cos(angle between x axis of a_2, y axis of a_1) $= 0$	$f_{12} = 0$
is-0-a13 †	cos(angle between x axis of a_2, z axis of a_1) $= 0$	$f_{13} = 0$
eq-x-angle	angle between x axes of a_1 and a_2 = angle between x axes of b_1 and b_2	$f_{11} = f'_{11}$
eq-x-axis	x-axis of a_2 is equal to x axis of a_n ‡	$f_{11}f'_{11} + f_{12}f'_{12} + f_{13}f'_{13} = 1$
eq-y-axis	y-axis of a_2 is equal to y-axis of a_n ‡	$f_{21}f'_{21} + f_{22}f'_{22} + f_{23}f'_{23} = 1$
eq-a-origin †	origin of a_1 is coincident with origin of a_2, d-oo$(a_1, a_2)=0$	$l_{11} = l_{12} = l_{13} = 0$

† Similarly for body B. Substitute f'_{ij} for f_{ij} and l_{2j} for l_{1j}

‡ $a_n = b_2 b_1^{-1} a_1$

Table A—2: Geometric Boolean Functions - 1

Name	Geometric Interpretation	Algebraic Interpretation
oa2-on-xa1 †	origin of a_2 on x axis of a_1, d-xo$(a_1, a_2) = 0$	$l_{12} = l_{13} = 0$
oa2-on-ya1 †	origin of a_2 on y axis of a_1	$l_{11} = l_{13} = 0$
oa2-on-za1 †	origin of a_2 on z axis of a_1	$l_{11} = l_{12} = 0$
oa2-on-plxa1 †	origin of a_2 on x=0 plane of a_1	$l_{11} = 0$
oa2-on-plya1 †	origin of a_2 on y=0 plane of a_1	$l_{12} = 0$
oa2-on-plza1 †	origin of a_2 on z=0 plane of a_1	$l_{13} = 0$
oa1-on-pla2 †	origin of a_1 on x=0 plane of a_2	$f_{11}l_{11} + f_{12}l_{12} + f_{13}l_{13} = 0$
oan-on-pla2 ‡	origin of a_n on x=0 plane of a_2	$f_{11}l_{11} + f_{12}l_{12} + f_{13}l_{13} =$ $f_{11}l_{21} + f_{12}l_{22} + f_{13}l_{23}$

† Similarly for body B. Substitute f'_{ij} for f_{ij} and l_{2j} for l_{1j}

‡ $a_n = b_2 b_1^{-1} a_1$

Table A–3: Geometric Boolean Functions - 2

Name	Geometric Interpretation	Algebraic Interpretation
ed-op-op	$\text{d-op}(a_1, a_2) =$ $\text{d-op}(b_1, b_2)$	$f_{11}l_{11} + f_{12}l_{12} + f_{13}l_{13} =$ $f'_{11}l_{21} + f'_{12}l_{22} + f'_{13}l_{23}$
ed-po-po	$\text{d-op}(a_2, a_1) =$ $\text{d-op}(b_2, b_1)$	$l_{11} = l_{21}$
ed-xo-xo	$\text{d-xo}(a_1, a_2) =$ $\text{d-op}(b_1, b_2)$	$\sqrt{l_{12}^2 + l_{13}^2} =$ $\sqrt{l_{22}^2 + l_{23}^2}$
ed-oo-oo	$\text{d-oo}(a_1, a_2) =$ $\text{d-oo}(b_1, b_2)$	$\sqrt{l_{11}^2 + l_{12}^2 + l_{13}^2} =$ $\sqrt{l_{21}^2 + l_{22}^2 + l_{23}^2}$
ed-cm-cm	$\text{d-cm}(a_1, a_2) =$ $\text{d-cm}(b_1, b_2)$	$(f_{12}l_{13} - f_{13}l_{12})/\sqrt{f_{12}^2 + f_{13}^2} =$ $(f'_{12}l_{23} - f'_{13}l_{22})/\sqrt{f'^2_{12} + f'^2_{13}}$
ed-op-oo	$\text{d-op}(a_1, a_2) =$ $\text{d-oo}(b_1, b_2)$	$|f_{11}l_{11} + f_{12}l_{12} + f_{13}l_{13}| =$ $\sqrt{l_{21}^2 + l_{22}^2 + l_{23}^2}$
ld-op-oo †	$\text{d-op}(a_1, a_2) <$ $\text{d-oo}(b_1, b_2)$	$|f_{11}l_{11} + f_{12}l_{12} + f_{13}l_{13}| <$ $\sqrt{l_{21}^2 + l_{22}^2 + l_{23}^2}$
ed-op-xo	$\text{d-op}(a_1, a_2) =$ $\text{d-xo}(b_1, b_2)$	$|f_{11}l_{11} + f_{12}l_{12} + f_{13}l_{13}| = \sqrt{l_{22}^2 + l_{23}^2}$
ed-oi-oo	$\text{d-oi}(a_1, a_2) =$ $\text{d-oo}(b_1, b_2)$	$\left|(f_{11}l_{11} + f_{12}l_{12} + f_{13}l_{13})/\sqrt{f_{12}^2 + f_{13}^2}\right| =$ $\sqrt{l_{21}^2 + l_{22}^2 + l_{23}^2}$
ed-om-xo	$\text{d-oi}(a'_1, a_2) \ddagger =$ $\text{d-xo}(b_1, b_2)$	$(f_{11}(l_{11} - l_{21}) + f_{12}l_{12} + f_{13}l_{13})/\sqrt{f_{12}^2 + f_{13}^2} =$ $\sqrt{l_{22}^2 + l_{23}^2}$

† Similarly for ld-op-xo, ld-oi-oo, ld-op-op, ld-om-xo, ld-xi-xp

‡ $a'_1 = trans(l_{21}, 0, 0)a_1$

Table A-4: Geometric Boolean Functions - 3

A.2   Constructing the new features

Tables A–5 — A–12 present a set of functions for constructing the locations of the features of bodies A and B between which the equivalent relationship holds.

The locations of the features between which the equivalent relationship holds (a_3 and b_3) are constructed from the locations a_1, b_1, a_2 and b_2. In most cases the location of the new feature a_3 of body A will only depend on a_1 and a_2. In the cases, however, that there two sets of solutions to the equations, the location a_3 will also depend on b_1 and b_2.

The new features are constructed using a set of functions which take as arguments two or four locations and produce one or two new locations. The following conventions are used for naming the geometric construction functions:

1. The function names are made up from six characters. The first two define the new origin, the next two define the new x-axis and the final two define the new y-axis. Let us denote a function name by $o_a x_b y_c$.

2. A function name starting with 'o', takes two arguments (pos1 and pos2) and constructs one new location (pos3). A function name starting with 'q', takes four arguments (pos1, pos2, pos3, pos4) and constructs two new locations (pos3, pos4). A function name starting with 'd', takes as arguments two locations and constructs two new locations.

It can be noted that since reversing the x-axes with the y-axes of a location corresponds to the transformation $twix(\pi)XTOY$, if

$$o_a x_b y_c = trans(\mathrm{d})\hat{R}$$

then

$$o_a y_c x_b = trans(\mathrm{d})twix(\pi)XTOY\hat{R}$$

Tables A–5 — A–6 present the geometric interpretation of the construction functions. Tables A–7 — A–12 present the algebraic interpretation of the functions.

o1	origin of pos1
q1	origin of pos1, two solutions
d1	origin of pos1, two solutions
o2	origin of pos2
oi	some point of intersection of x=0 planes of pos1 and pos2
op	intersection of x-axis of pos1 with x=0 plane of pos2
oy	intersection of y-axis of pos1 with x=0 plane of pos2
oo	intersection of x-axis of pos1 with the plane perp to x-axis of pos1 and through the origin of pos2
ox	intersection of x-axis of pos1 with the plane perp to x-axis of pos1 and containing x-axis of pos2
on	intersection of x-axis of pos1 with x=n plane of pos1

Table A–5: Naming conventions for origin of constructed location

x1	x-axis of pos1
x2	x-axis of pos2
xc	direction from pos1 to pos2
xi	direction of intersection of x=0 planes
xp	direction of common perp between x-axes
xj	direction of common perp between parallel x-axes
xo	direction of perp from o2 to x1
y1	y-axis of pos1
z1	z-axis of pos1
ya	any y direction
yc	crossvec of x-axis of pos1 with direction of line joining origin of pos1 to origin of pos2

Table A–6: Naming Conventions for axes of constructed location

Function(pos1, pos2) $= twix(\alpha)$ pos1		
Function	$e^{i\alpha}$	Condition
o1x1y1	0	-
o1x1yt	e^{it} †	-
o1x1x2	$f_{12} + if_{13}$	$\dot{f_{11}} = 0$
o1x1xm	$(f_{12} + if_{13})/\sqrt{f_{12}^2 + f_{13}^2}$	$f_{11} \neq \pm 1$
o1x1xp	$(-f_{13} + if_{12})/\sqrt{f_{12}^2 + f_{13}^2}$	$f_{11} \neq \pm 1$
o1x1xo	$(l_2 + il_3)/\sqrt{l_2^2 + l_3^2}$	-
o1x1yc	$(-l_3 + il_2)/\sqrt{l_2^2 + l_3^2}$	-
o1x1xj	$(l_2 + il_3)/\sqrt{l_2^2 + l_3^2}$	$f_{11} = \pm 1$
o1x1z1	$e^{i\pi/2}$	-

† t is a constant

Table A–7: Geometric Constructions - 1

Function(pos1, pos2) = trans(x,y,z) pos1				
Function	x	y	z	Condition
o2x1y1	l_1	l_2	l_3	-
onx1y1	n †	0	0	-
oox1y1	l_1	0	0	-
oxx1y1	l_1	0	0	$f_{11} = 0$
opx1y1	$(f_{11}l_1 + f_{12}l_2 + f_{13}l_3)/f_{11}$	0	0	-
oyx1y1	0	$(f_{11}l_1 + f_{12}l_2 + f_{13}l_3)/f_{12}$	0	-
otx1y1	n_1 †	n_2	n_3	-

† constant

Table A–8: Geometric Constructions - 2

Function(pos1, pos2) = $twix(\pi + \alpha)XTOYtwix(\beta)$pos1		
Function	$e^{i\alpha}$	$e^{i\beta}$
o1xpx2	$f_{11} + i\sqrt{f_{12}^2 + f_{13}^2}$	$(-f_{13} + if_{12})/\sqrt{f_{12}^2 + f_{13}^2}$
o1ycxc	$(l_1 + i\sqrt{l_2^2 + l_3^2})/\sqrt{l_1^2 + l_2^2 + l_3^2}$	$(-l_3 + il_2)/\sqrt{l_2^2 + l_3^2}$
o1y1z1	$e^{-i\pi/2}$	

Table A–9: Geometric Constructions - 3

Function(pos1, pos2) = trans(0,0,d)$twix(\alpha)$pos1		
Function	$e^{i\alpha}$	d
oix1xi	$(-f_{13} + if_{12})/\sqrt{f_{12}^2 + f_{13}^2}$	$-(f_{11}l_1 + f_{12}l_2 + f_{13}l_3)/\sqrt{f_{12}^2 + f_{13}^2}$

Table A–10: Geometric Constructions - 4

Function(pos1, pos2, pos3, pos4) $=twix(\pm\beta)twix(\alpha)$pos1		
Function	$e^{i\alpha}$	$e^{i\beta}$
qlxlps	$(l_{23} - il_{22})/(\sqrt{l_{22}^2 + l_{23}^2})$	$-\dfrac{f_{11}(l_{11}-l_{21})+f_{12}l_{12}+f_{13}l_{13}+i...}{\sqrt{f_{12}^2+f_{13}^2}\sqrt{l_{22}^2+l_{23}^2}}$
qlxlpd	$(f_{12} + if_{13})/\sqrt{f_{12}^2 + f_{13}^2}$	$\dfrac{f_{11}l_{11}+f_{12}l_{12}+f_{13}l_{13}+i...}{\sqrt{f_{12}^2+f_{13}^2}\sqrt{l_{21}^2+l_{22}^2+l_{23}^2}}$
qlxlpl	$(l_{22} + il_{23})/(\sqrt{l_{22}^2 + l_{23}^2})$	$\dfrac{f_{11}l_{11}+f_{12}l_{12}+f_{13}l_{13}+i...}{\sqrt{l_{22}^2+l_{23}^2}}$

Table A-11: Geometric Constructions - 5

d-functions	
dly1x1(pos1,pos2)	$= $ oly1x1$(twix(\pm\pi/2)\ twix(\pi/2)$ pos1,pos2)
dlycx1(pos1,pos2)	$= $ olycx1$(twix(\pm\pi/2)\ twix(\pi/2)$ pos1,pos2)
dlzly1(pos1,pos2)	$= [XTOY\ XTOY]$ olzly1(pos1,pos2)
dlxox1(pos1,pos2)	$= [XTOY\ XTOY]$ olxox1(pos1,pos2)
dnxly1(pos1,pos2,k,n)	$= trans(k \pm n)$ pos1

Table A-12: Geometric Constructions - 6

A.3 Spatial Relationships

Table A-13 presents the set of spatial relationships which has been derived in the process of constructing the substitution table. The following conventions have been followed for naming the spatial relationships:

- The letters 'AG' stand for 'against'.

- The letters 'P', 'E', and 'V' stand for 'plane', 'edge' and 'vertex' respectively. For example 'AGPV' stands for 'against plane vertex'.

- The letters 'ROT' or 'R' imply a rotational degree of freedom. For example, the relationship 'ROT2' has two rotational degrees of freedom.

- The letters 'LIN' or 'L' imply a translational degree of freedom. For example the relationship 'RRL' has two rotational and one translational degrees of freedom.

- A relationship name starting with the letter 'G' implies that the relationship is a 'general' form of some other relationship. For example the relationship denoted by 'GAGPE' is a general form of the 'AGPE' relationship in the sense that the axes of rotations are not necessarily perpendicular.

- If a relationship name starts with the letter 'F' then the degrees of freedom of the relationship are interdependent.

Figures A-1 — A-4 show some of these relationships.

d.o.f.		Name	Algebraic Form
R	T		
3	2	AGPV	$twix(\theta)XTOY\,twix(\phi)XTOY\,twix(\psi)trans(0,y,z)$
3	2	AGVP	$twix(\theta)XTOY\,trans(x,0,z)twix(\phi)XTOY\,twix(\psi)$
3	2	AGEE	$twix(\theta)XTOY\,twix(\phi)XTOY\,trans(x,0,z)twix(\psi)$
3	1	AGEV	$twix(\theta)XTOY\,twix(\phi)XTOY\,twix(\psi)trans(x,0,0)$
2	2	AGPE	$twix(\theta)XTOY\,twix(\phi)trans(0,y,z)$
2	2	GAGPE	$twix(\theta)\hat{E}twix(\phi)trans(0,y,z)$ *
3	0	AGVV	$twix(\theta)XTOY\,twix(\phi)XTOY\,twix(\psi)$
2	1	RRL	$twix(\theta)XTOY\,twix(\phi)trans(0,y,0)$
2	1	GRRL	$twix(\theta)\hat{E}twix(\phi)trans(0,y,0)$
1	2	AGPP	$twix(\theta)trans(0,y,z)$
2	0	ROT2	$twix(\theta)XTOY\,twix(\phi)$
2	0	GROT2	$twix(\theta)\hat{E}twix(\phi)$
1	1	ROTYLIN	$twix(\theta)trans(0,y,0)$
1	1	FITS	$twix(\theta)trans(x,0,0)$
1	1(1)**	ROTLIN †	$twix(\theta)trans(x,y(x),0)$
1(1)	1	FROTLIN †	$twix(\theta)XTOY\,twix(\phi(\theta))trans(0,y,0,)$
2	0(1)	ROTLAD †	$twix(\theta)XTOY\,twix(\phi)trans(0,y(\phi),0)$
1	0	ROT	$twix(\theta)$
1	0(1)	LAD	$twix(\theta)trans(0,a\cos\theta+b\sin\theta,0)$
1	0(1)	SCR	$twix(\theta)trans(a\cos\theta+b\sin\theta,0,0)$
1(1)	0	FROT †	$twix(\theta)XTOY\,twix(\phi(\theta))$
1(1)	0	GFROT†	$twix(\theta)\hat{E}twix(\phi(\theta))$
0	1	LIN	$trans(x,0,0)$

* $\hat{E}$ is a constant matrix: $\hat{E} = XTOY\,twix(\omega)XTOY$, $e^{i\omega} = a + ib$

** (n) means n dependent variables

† This relationship has been derived but is not used in the substitution table

Table A–13: Set of Spatial Relationships
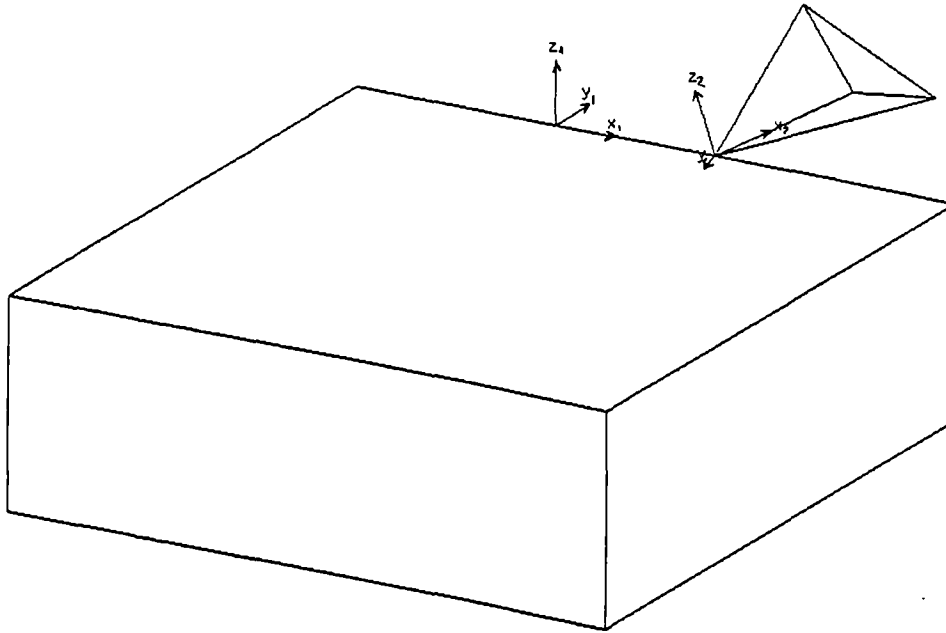
$$twix(\theta)XTOYtwix(\phi)XTOYtwix(\psi)trans(x,0,0)$$

Figure A–1: AGEV relationship
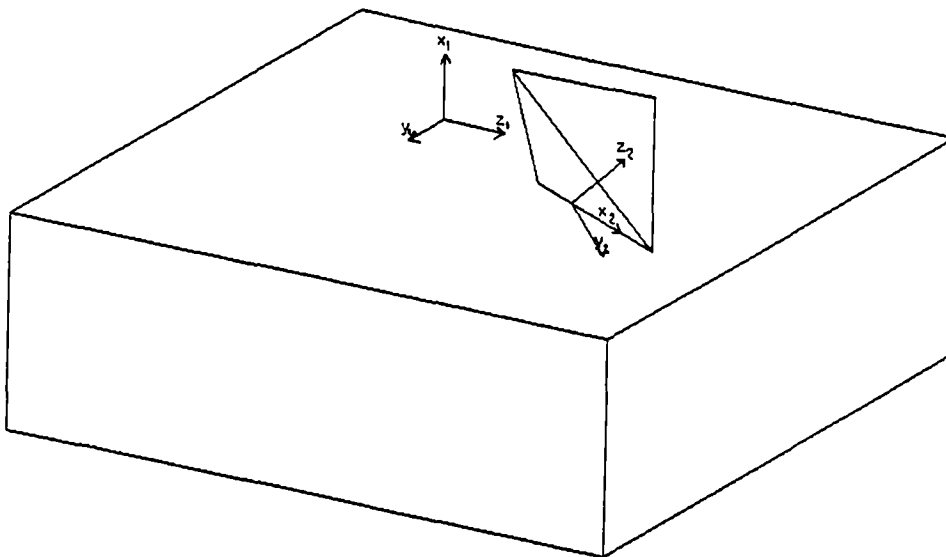
$$twix(\theta)XTOYtwix(\phi)trans(0,y,z)$$

Figure A–2: AGPE relationship
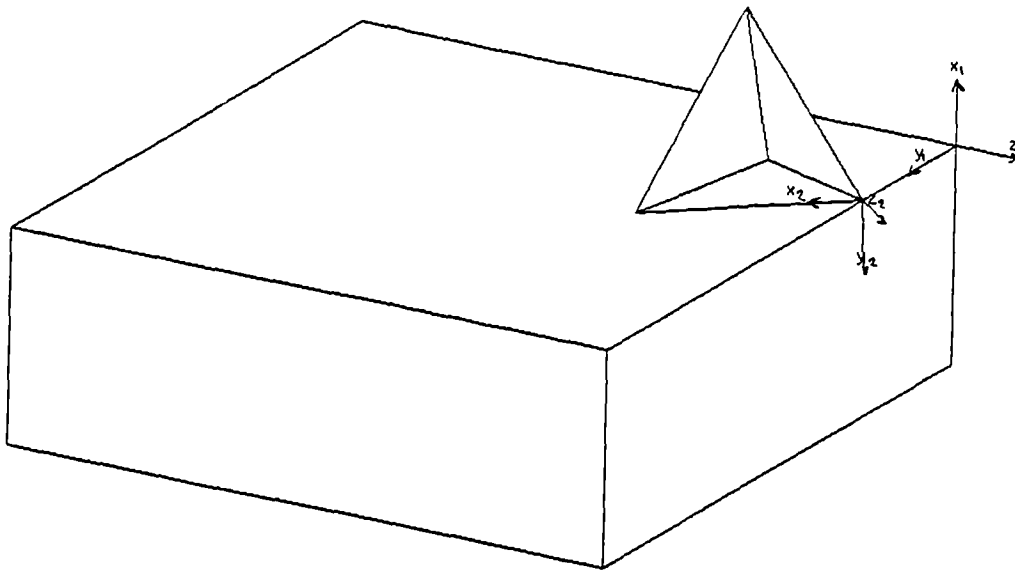
$twix(\theta)XTOYtwix(\phi)trans(0, y, 0)$

Figure A–3: RRL relationship
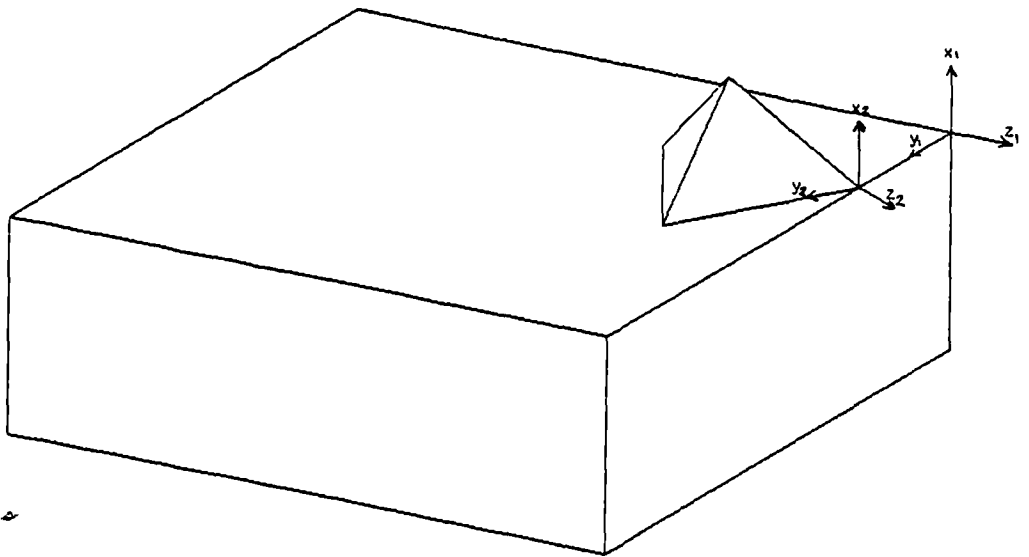
$twix(\theta)trans(0, y, 0)$

Figure A–4: ROTYLIN relationship

A.4 The Substitution Table

Tables A-14 — A-30 present the rules for substituting two relationships by an equivalent relationship. The interpretation of the table is explained below.

1. A table is selected by matching the pair of relationships to E_1 and E_2.

2. A row is selected by examining the entries in the column 'Condition 1' until a match is found.

3. If 'Condition 2' is satisfied then the relationships are consistent. In this case the equivalent relationship is given by E_e and the new features are constructed using the functions in the column 'New Features'.

4. If 'Condition 2' is not satisfied then the two relationships are inconsistent.

E_1	E_2	E_e	Condition 1	Condition 2	New Features $\frac{a_3}{b_3}$
AGPV	AGPV	AGPV	x-a-par eq-b-origin	oal-on-pla2	a_1 b_1
		AGPE	x-a-par oal-on-pla2		a_1 olxcyc
		AGPP	x-a-par ed-op-oo		a_1 olxcyc
		GAPPE†	x-a-par	ld-op-oo	a_1 olxcyc
		AGEV	eq-b-origin		oixixl b_1
		-	general		

† The constants of the relationship are given by:

a=d-op(a_1, a_2)/d-oo(b_1, b_2) and $b = \sqrt{1 - a^2}$

Table A-14: Substitution Table: AGPV-AGPV

E_1	E_2	E_e	Condition 1	Condition 2	New Features $\frac{a_3}{b_3}$
AGPE	AGPV	AGPE	x-a-par ob2-on-xb1	oa1-on-pla2	a_1 b_1
		AGPP	x-a-par ed-op-xo		a_1 o1xox1
		AGPP*	x-a-par	ld-op-xo	a_1 q1plx1
		RRL	ob2-on-ob1		oix1xi o2x1y1
		-	general		-

* An asterisk denotes that there are two sets of solutions

Table A–15: Substitution Table: AGPE-AGPV

E_1	E_2	E_e	Condition 1	Condition 2	New Features $\frac{a_3}{b_3}$
GAGPE [1]	AGPV	GAGPE	x-a-par ob2-on-xb1	oan'-on-pla2 †	a_1 b_1
		AGPP	x-a-par ed-oi-oo ‡		a_1 o1ycx1
		AGPP*	x-a-par	ld-oi-oo ‡	al -
		GRRL	general		-

[1] a and b are the constants of the relationship

† $a'_n = trans(-a\, l_{21}, 0, 0)a_1$

‡ Distance from origin of a_1 to intersection of $x = 0$ plane of a_2 with $x = al_{21}$ plane of a_1

Table A–16: Substitution Table: GAGPE-AGPV

E_1	E_2	E_e	Condition 1	Condition 2	New Features $\begin{smallmatrix}a_3\\b_3\end{smallmatrix}$
AGEV	AGPV	AGEV	x-a-perp eq-b-origin	oal-on-pla2	a_1 b_1
		RRL	x-a-perp oal-on-pla2		olx2xl olxcyc
		ROTYLIN	x-a-perp ed-op-oo		olx2xl olxcyc
		GRRL†	x-a-perp	ld-op-oo	olx2xl olxcyc
		AGVV	eq-b-origin		opxlyl b_1
		-	general		-

† The constants of the relationship are given by:

a=d-op(a_1, a_2)/d-oo(b_1, b_2) and $b = \sqrt{1 - a^2}$

Table A–17: Substitution Table: AGEV-AGPV

E_1	E_2	E_e	Condition 1	Condition 2	New Features $\frac{a_3}{b_3}$
AGVV	AGPV	AGVV	eq-b-origin	oa1-on-pla2	a_1
					b_1
		ROT2	oa1-on-pla2		o1x2yc
			ob2-on-xb1		b_1
		ROT2	oa1-on-pla2		o1x2yc
					o1xcyc
		ROT	ed-op-oo		o1x2yc
			ob2-on-xb1		b_1
		ROT	ed-op-oo		o1x2yc
					o1xcyc
		GROT2†	general	ld-op-oo	o1x2yc
					o1xcyc

† The constants of the relationship are given by:

a=d-op(a_1, a_2)/d-oo(b_1, b_2) and $b = \sqrt{1 - a^2}$

Table A–18: Substitution Table: AGVV-AGPV

E_1	E_2	E_c	Condition 1	Condition 2	New Features $\genfrac{}{}{0pt}{}{a_3}{b_3}$
RRL	AGPV	RRL	x-a-par ob2-on-xb1	oa1-on-pla2	a_1 b_1
		ROTYLIN	x-a-par ed-op-xo		a_1 o1xox1
		ROTYLIN*	x-a-par	ld-op-xo	a_1 q1plx1
		RRL	is-0-a12 eq-b-origin	oa1-on-pla2	a_1 b_1
		FITS*	is-0-a12 ob2-on-xb1 oa1-on-pla2		d1y1x1 b_1
		ROTYLIN	is-0-a12 ob2-on-xb1 ed-oi-oo		o1z1x1 b_1
		ROTLIN*	is-0-a12 ob2-on-xb1	ld-oi-oo	q1pdx1 b_1
		ROTYLIN**	is-0-a12 x-a-perp ob2-on-plxb1 oa1-on-pla2		d1z1y1, a_1 b_1 , d1xox1
		FROTLIN	is-0-a12		- -
		ROT2	eq-b-origin		oyx1y1 b_1
		ROTLAD	ob2-on-xb1		- -
		-	general		- -

** Four sets of solutions

Table A-19: Substitution Table: RRL-AGPV

E_1	E_2	E_e	Condition 1	Condition 2	New Features $\begin{smallmatrix}a_3\\b_3\end{smallmatrix}$
GRRL [1]	AGPV	GRRL	x-a-par ob2-on-xb1	oan'-on-pla2 †	a_1 b_1
		GRRL	is-0-al2 eq-b-origin	oal-on-pla2	a_1 b_1
		ROTYLIN	x-a-par ed-oi-oo ‡		a_1 olycx1
		ROTYLIN*	x-a-par	ld-oi-oo †	al -
		FITS	is-0-al2 ob2-on-xb1		$\hat{E}a_1$ b_1
		-	general		-

[1] a and b are the constants of the relationship

†$a'_n = trans(-a\,l_{21}, 0, 0)a_1$

‡Distance from origin of a_1 to intersection of $x = 0$ plane of a_2 with $x = al_{21}$ plane of a_1

Table A–20: Substitution Table: GRRL-AGPV

E_1	E_2	E_e	Condition 1	Condition 2	New Features $\begin{smallmatrix}a_3\\b_3\end{smallmatrix}$
AGPP	AGPV	AGPP	x-a-par	oan-on-pla2	a_1 b_1
		ROTYLIN	general		oix1xi(a'_1, a_2)† o2x1y1

†$a'_1 = trans(l_{21}, 0, 0)a_1$

Table A–21: Substitution Table: AGPP-AGPV

E_1	E_2	E_e	Condition 1	Condition 2	New Features $\begin{matrix}a_3\\b_3\end{matrix}$
ROT2	AGPV	ROT2	eq-b-origin	oa1-on-pla2	a_1 b_1
		ROT	ob2-on-xb1 ed-oi-oo		o1x1xm b_1
		ROT	x-a-par ed-op-xo		a_1 o1xox1
		ROT*	ob2-on-xb1	ld-oi-oo	q1pdx1 b_1
		ROT*	x-a-par	ld-op-xo	a_1 q1plx1
		ROT	x-a-perp ed-op-op		o1x2x1 b_1
		FIX	ed-op-oo		o1x2yc -
		FROT	general		-

Table A–22: Substitution Table: ROT2-AGPV

E_1	E_2	E_e	Condition 1	Condition 2	New Features $\genfrac{}{}{0pt}{}{a_3}{b_3}$
GROT2 [1]	AGPV	GROT2	eq-b-origin	oal-on-pla2	a_1 b_1
		GROT2	x-a-par ob2-on-xb1	oan'-on-pla2 †	a_1 b_1
	.	ROT	ob2-on-xb1 ed-oi-oo ‡		$\hat{E}$olx1xp b_1
		ROT*	ob2-on-xb1	ld-oi-oo ‡	q1x1ps b_1
		ROT*	general		-

[1] a and b are the constants of the relationship

†$a'_n = trans(-a\,l_{21}, 0, 0)a_1$

‡Distance from origin of a_1 to intersection of $x = 0$ plane of a_2 with $x = al_{21}$ plane of a_1

Table A–23: Substitution Table: GROT2-AGPV

E_1	E_2	E_e	Condition 1	Condition 2	New Features $\genfrac{}{}{0pt}{}{a_3}{b_3}$
ROTYLIN	AGPV	ROTYLIN	x-a-par	oan-on-pla2	a_1 b_1
		ROTYLIN	is-0-a12 ob2-on-xb1	oan-on-pla2	a_1 b_1
		LIN	is-0-a12 ed-om-xo		oly1x1 olycx1
		LIN*	is-0-a12	ld-om-xo	oly1x1 q1psx1
		ROT	ob2-on-xb1		oyx1y1(a_n, a_2)† o2x1y1
		LAD‡	general		oyx1y1(a_n, a_2)† oox1y1

† $a_n = b_3 b_1^{-1} a_1$

‡ The constants of the relationship are given by:

$a = -(f_{12} l_{22} + f_{13} l_{23})/f_{12}$

$b = (f_{12} l_{23} - f_{13} l_{22})/f_{12}$

Table A–24: Substitution Table: ROTYLIN-AGPV

E_1	E_2	E_e	Condition 1	Condition 2	New Features $\frac{a_3}{b_3}$
FITS	AGPV	FITS	x-a-perp ob2-on-xb1	oal-on-pla2	a_1 b_1
		LIN	x-a-perp ed-op-xo		o1x1x2 o1x1xo
		LIN*	x-a-perp	ld-op-xo	o1x1xp q1x1ps
		ROT	ob2-on-xb1		opx1y1 o2x1y1
		ROT	x-a-par		opx1y1 oox1y1
		SCR	general		opx1y1 oox1y1

Table A–25: Substitution Table: FITS-AGPV

E_1	E_2	E_e	Condition 1	Condition 2	New Features $\frac{a_3}{b_3}$
ROT	AGPV	ROT	ob2-on-xb1	oan-on-pla2	a_1 b_1
		ROT	x-a-par		a_1 b_1
		FIX	x-a-perp ed-op-xo		o1x1x2 o1x1xo
		FIX*	x-a-perp	ld-op-xo	o1x1xp q1x1ps
		FIX	ed-om-xo		o1x1xp o1x1yc
		FIX*	general	ld-om-xo	o1x1xp q1x1ps

Table A–26: Substitution Table: ROT-AGPV

E_1	E_2	E_e	Condition 1	Condition 2	New Features $\begin{smallmatrix} a_3 \\ b_3 \end{smallmatrix}$
LIN	AGPV	LIN	x-a-perp	oan-on-pla2	a_1 b_1
		FIX	x-a-par		opx1y1 oox1y1
		FIX	ob2-on-x-b1		opx1y1 oox1y1
		FIX	general		opx1y1(a_n, a_2)† o2x1y1

†$a_n = b_3 b_1^{-1} a_1$

Table A–27: Substitution Table: LIN-AGPV

E_1	E_2	E_e	Condition 1	Condition 2	New Features $\begin{smallmatrix} a_3 \\ b_3 \end{smallmatrix}$
LAD	AGPV	LAD	x-a-par	oan-on-pla2	a_1 b_1
		LAD	is-0-a12‡ ob2-on-xb1	oan-on-pla2	a_1 b_1
		FIX	is-0-a12 ed-om-xo		ony1x1† o1ycx1
		FIX*	is-0-a12 oan-on-pla2		dny1x1† d1ycx1
		FIX*	is-0-a12	ld-om-xo	dny1x1† q1psx1
		FIX*	general		-

‡ Similarly for the case is-0-a13.

Substitute b_1 for $b_1' = trans(0, -a, b)\, b_1$,

where a, b are the constants in E_1

† $n = (-al_{23} + bl_{22})/\sqrt{l_{22}^2 + l_{23}^2}$

Table A–28: Substitution Table: LAD-AGPV

E_1	E_2	E_e	Condition 1	Condition 2	New Features $_{b_3}^{a_3}$
AGEE	AGEE	AGEE	x-a-collin		a_1
			x-b-collin		b_1
		AGPE	x-a-par		olycxl
			x-b-colin		b_1
		AGPE	x-a-coplan		olxlxp
			x-b-collin		b_1
		AGPE	x-b-par	.	a_1
			x-a-colin		olycxl
		AGPE	x-a-coplan		a_1
			x-b-collin		olxlxp

Table A–29: Substitution Table: AGEE-AGEE

E_1	E_2	E_e	Condition 1	Condition 2	New Features $_{b_3}^{a_3}$
AGPV	AGEE	AGPE	oa2-on-plal		a_1
			x-a-perp		b_2
			ob1-on-xb2		
		-	general		-

Table A–30: Substitution Table: AGPV-AGEE

A.5 Equivalent Relationships

Table A-31 presents the rules for deciding whether or not two relationships are equivalent. The interpretation of this table is similar to the interpretation of the substitution table, i.e. relationships E_1 and E_2 are equivalent if 'Condition 1' is satisfied. If 'Condition 1' is satisfied but 'Condition 2' is not satisfied then the relationships are inconsistent.

E_1	E_2	E_e	Condition 1	Condition 2
LIN	LIN	LIN	x-a-par	oan-on-pla2
ROT	ROT	ROT	x-a-collin	x-b-collin ed-po-po
LAD †	LAD	LAD	x-a-par y-a-par ob2-on-xb1	x-b-par oan-on-pla2 oa2-on-plza1
FITS	FITS	FITS	x-a-collin	x-b-collin
ROTYLIN	ROTYLIN	ROTYLIN	x-a-par y-a-par ob2-on-xb1	ed-po-po oa2-on-plza1
ROT2	ROT2	ROT2	x-a-par x-b-par eq-b-origin	eq-a-origin
AGPP	AGPP	AGPP	x-a-par	ed-po-po
RRL	RRL	RRL	x-a-par x-b-par eq-b-origin y-a-par	oa2-on-ya1
AGVV	AGVV	AGVV	eq-a-origin	eq-b-origin
AGPS	AGPS	AGPS	x-a-par eq-b-origin	oa1-on-pla2
AGEE	AGEE	AGEE	x-a-collin x-b-collin	

† Similarly for the case x-a-par and y-a-perp. The same conditions apply if b_1 is substituted for $b_1' = trans(0, -a, b) \, b_1$, where a, b are the constants of E_1

Table A–31: Equivalent Relationships

A.6 Parameters of One Degree of Freedom Relationships

Given a one degree relationship E holding between features with locations a_1 and b_1 and a location p, Table A–32 summarises the conditions under which the location satisfies the relationship and the value of the parameter of the relationship at that location. Let a_2 be the location b transformed by p, i.e.

$$a_2 = b\,p.$$

E_1	Condition	Parameter
LIN	x-a-eq y-a-eq oa2-on-xa1	$x = \text{d-op}(a_2, a_1)$
ROT	eq-a-origin x-a-eq	$\cos\theta = f_{22}$ $\sin\theta = f_{23}$
LAD	x-a-par oa2-on-ya1 $l_{12} = a\, f_{22} = b\, f_{23}$	$\cos\theta = f_{22}$ $\sin\theta = f_{23}$

Table A–32: Parameterising one degree of freedom relationships