**Inspection plan and code generation for coordinate measuring machines in a product modelling environment**

This item was submitted to Loughborough University’s Institutional Repository by the/an author.

**Additional Information:**

- A Doctoral Thesis. Submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy of Loughborough University.

**Metadata Record:** [https://dspace.lboro.ac.uk/2134/11813](https://dspace.lboro.ac.uk/2134/11813)

**Publisher:** © Michael J. Corrigall

Please cite the published version.
This item was submitted to Loughborough University as a PhD thesis by the author and is made available in the Institutional Repository (https://dspace.lboro.ac.uk/) under the following Creative Commons Licence conditions.

For the full text of this licence, please go to: http://creativecommons.org/licenses/by-nc-nd/2.5/
<table>
<thead>
<tr>
<th>VOL. NO.</th>
<th>CLASS MARK</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>date due:</th>
<th>date due:</th>
<th>date due:</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>loan copy</th>
<th>loan copy</th>
<th>loan copy</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 JUL 1993</td>
<td>11 JUL 1993</td>
<td>18 MAR 1994</td>
</tr>
</tbody>
</table>

LOUGHBOROUGH UNIVERSITY OF TECHNOLOGY LIBRARY

AUTHOR/FILING TITLE

Corrigan, M J

ACCESSION/COPY NO.

036000161

VOL. NO.

036000161

CLASS MARK

036000161
Inspection Plan and Code Generation for Coordinate Measuring Machines in a Product Modelling Environment

by

Michael J. Corrigall

A Doctoral Thesis submitted in partial fulfilment of the requirements for the award of

Doctor of Philosophy of the Loughborough University of Technology

July 1990

© by Michael J. Corrigall 1990
à Valérie
Acknowledgements

This work has been carried out as part of a project funded by the ACME Directorate of the Science and Engineering Research Council and the Department of Trade and Industry entitled Information Support Systems for Design and Manufacture, which is supported by a group of industrial collaborators.

The author would like to acknowledge the professional and knowledgeable community of researchers of this research project at both Loughborough University of Technology and the University of Leeds.

The author would like to pay an additional tribute to the following individuals who have each made a measurable contribution to the Inspection Planning research:

- **Professor R. Bell** for continuous support and encouragement throughout the project and for the substantial effort that he devoted to acquiring a Coordinate Measuring Machine for this research.
- **Professor A. de Pennington** for discussion and constructive criticism of the research.
- **Dr M. Sabin** and **Dr A. Hope** for providing knowledgeable and constructive criticism in their rôles as members of the SERC/ACME Review Panel.
- **Laurence Wickens** for valuable discussions and guidance on both inspection and his research field of Dimensions and Tolerances.
- **Pete Dawson** and **Paul Bell** for their assistance and advice regarding Ada and the Spatially Divided Solid Modelling environment.
- **Alison McKay** for assistance and advice regarding the Structure Editor and for generously implementing the geometry of the Wrist Motor Body case-study.
- **Bob Young** for discussions regarding the planning of fixturing and component setups.
• Gregorio Varvakis for discussions regarding the literature survey of this thesis.

• Trevor Downham and Jagpal Singh for their generous assistance and advice on practical engineering matters.

• Dave Walters for assistance with computational matters and for helping to establish a file-transfer link to the CMM controller.

• Ross Harris of GEC(FAST) for supporting this work at GEC(FAST), for providing case-study material on the Wrist Motor Body and for assisting the transfer of the Ferranti CMM to Loughborough.

• Phil Williams of Renishaw for supporting this work at Renishaw and for generously providing Renishaw probing equipment throughout the project.

• Mel Fletcher of LK Tools for valuable discussions on inspection and in particular DMIS, NDF and Valisys and for the provision of CMM training and access to an LK CMM.

• Graham Corrigall for his encouragement during the writing of this thesis and for the excellent computerised transcription of the figures.

• My parents for their continuous support, advice and encouragement throughout the research and during the writing of this thesis.
Abstract

This thesis describes research into Inspection Plan and Code Generation that has been carried out as part of a research project investigating Information Support Systems for Design and Manufacture. The major theme of this project has been the creation of a skeletal Design to Manufacture Environment which incorporates a feature-based design system, a Machine Planner and Cutter Path Generator for machining an Inspection Plan and Code Generator (the subject of this thesis), and a Manufacturing Data Analysis facility. This experimental environment is supported by a Product Modelling System that permits all geometrical and technological information required to design and manufacture the product to be represented so that full integration can be achieved in the Design to Manufacture Environment.

The survey of literature in this thesis covers the work of researchers in the field of Product Modelling Environments, in addition to work based on advanced Design to Manufacture systems, before concentrating on research directly concerned with Inspection Plan and Code Generation. The main body of the thesis begins by stating and explaining the objectives of the research and lists the issues that need to be addressed in order to meet these objectives. This is followed by a description of the experimental Design to Manufacture Environment, which includes an explanation of the Product Model and the interaction between it and the inspection application. The higher level issues of Inspection Planning are then discussed before attention is focussed on the individual planning activities that represent the main thrust of the research. Frequent references are made throughout these sections to a series of case-studies (Appendices C and D) based on components supplied by industrial collaborators and processed by an experimental Inspection Plan and Code Generator modelled on the theories promoted by this work. The conclusion of the thesis describes what has been learnt by this research and discusses how adequately the research objectives have been achieved.
The Contents

| Chapter 1 - Introduction                      | 1 |
| Chapter 2 - Geometric and Product Modelling  | 5 |
| 2.1 Introduction                              | 5 |
| 2.2 Geometric Modelling                       | 5 |
| 2.2.1 Wire-frames                             | 7 |
| 2.2.2 Polygonal Schemes                       | 8 |
| 2.2.3 Sculptured Surfaces                     | 8 |
| 2.2.4 Solid Modelling                         | 8 |
| 2.2.5 Problems with Solid Modelling           | 10 |
| 2.2.6 Dual Representation and Hybrid Solid Modelling | 11 |
| 2.3 Tolerancing                               | 13 |
| 2.3.1 Representation schemes for constraints  | 15 |
| 2.3.2 Analysis of tolerances for design and manufacture | 18 |
| 2.4 Product Modelling                         | 18 |
| 2.4.1 Neutral File Formats                    | 19 |
| 2.4.2 PDES/STEP                               | 20 |
| 2.4.3 Product Modelling Environments          | 21 |
| 2.4.3.1 University of Tokyo                   | 22 |
| 2.4.3.2 Universities of Leeds and Loughborough | 23 |
| 2.4.3.3 Research Elsewhere                   | 25 |
| Chapter 3 - The Design to Manufacture Environment | 28 |
| 3.1 Introduction                              | 28 |
| 3.2 The Design Environment                    | 28 |
| 3.2.1 Specification                           | 29 |
| 3.2.2 Detail Design                           | 29 |
| 3.2.3 Design Analysis                         | 31 |
| 3.3 The Manufacturing Environment             | 33 |
| 3.3.1 Computer-Aided Process Planning (CAPP)  | 33 |
| 3.3.1.1 Variant CAPP Systems                  | 35 |
| 3.3.1.2 Generative CAPP Systems               | 36 |
| 3.3.1.3 Semi-Generative CAPP Systems          | 37 |
| 3.3.2 Cutter Path Generation                  | 37 |
| 3.3.3 Cutter Path Verification                | 40 |
| 3.4 Integration in the Design to Manufacture Environment | 41 |
| 3.4.1 Integration of CAD and CAM              | 42 |
| 3.4.1.1 Design Concurrency in CAD/CAM         | 42 |
| 3.4.2 Computer Integrated Manufacture         | 44 |
| Chapter 4 - Inspection Plan and Code Generation | 48 |
11.5 The Method ..................................................................................... 165
11.6 Fixture Design ................................................................................. 168
11.7 The Results ...................................................................................... 170
11.8 Discussion ........................................................................................ 171

<table>
<thead>
<tr>
<th>Chapter 12 - Probe Setup Planning .................................................... 174</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.1 Introduction .................................................................................. 174</td>
</tr>
<tr>
<td>12.2 The Objectives ............................................................................... 174</td>
</tr>
<tr>
<td>12.3 Research Elsewhere ....................................................................... 175</td>
</tr>
<tr>
<td>12.4 The Assumptions and Limitations ................................................. 177</td>
</tr>
<tr>
<td>12.5 The Method used for Probe Configuration Determination .......... 178</td>
</tr>
<tr>
<td>12.5.1 The Stylus Tip ........................................................................... 178</td>
</tr>
<tr>
<td>12.5.2 The Probe Stylus ....................................................................... 180</td>
</tr>
<tr>
<td>12.5.3 The Probe Body .......................................................................... 180</td>
</tr>
<tr>
<td>12.6 The Method used for Probe Orientation Determination .............. 181</td>
</tr>
<tr>
<td>12.7 The Results .................................................................................... 182</td>
</tr>
<tr>
<td>12.8 Discussion ...................................................................................... 182</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 13 - Probing Point Planning ................................................. 184</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.1 Introduction .................................................................................. 184</td>
</tr>
<tr>
<td>13.2 The Objectives ............................................................................... 184</td>
</tr>
<tr>
<td>13.3 Research Elsewhere ....................................................................... 185</td>
</tr>
<tr>
<td>13.4 The Assumptions and Limitations ................................................. 187</td>
</tr>
<tr>
<td>13.5 The Method .................................................................................... 187</td>
</tr>
<tr>
<td>13.6 A Theoretical Method for Determining Ideal Point Distribution . 190</td>
</tr>
<tr>
<td>13.7 The Results .................................................................................... 191</td>
</tr>
<tr>
<td>13.8 Discussion ...................................................................................... 192</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 14 - Probe Path Planning ...................................................... 195</th>
</tr>
</thead>
<tbody>
<tr>
<td>14.1 Introduction .................................................................................. 195</td>
</tr>
<tr>
<td>14.2 The Objectives ............................................................................... 195</td>
</tr>
<tr>
<td>14.3 Research Elsewhere ....................................................................... 196</td>
</tr>
<tr>
<td>14.4 The Assumptions and Limitations ................................................. 198</td>
</tr>
<tr>
<td>14.5 The Method .................................................................................... 198</td>
</tr>
<tr>
<td>14.6 An Alternative Heuristic Method ................................................. 202</td>
</tr>
<tr>
<td>14.7 The Results .................................................................................... 203</td>
</tr>
<tr>
<td>14.8 Discussion ...................................................................................... 204</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 15 - Inspection Code Generation ........................................... 207</th>
</tr>
</thead>
<tbody>
<tr>
<td>15.1 Introduction .................................................................................. 207</td>
</tr>
<tr>
<td>15.2 The Objectives ............................................................................... 207</td>
</tr>
<tr>
<td>15.3 Research Elsewhere ....................................................................... 208</td>
</tr>
<tr>
<td>15.4 The Assumptions and Limitations ................................................. 209</td>
</tr>
<tr>
<td>15.5 The Method .................................................................................... 210</td>
</tr>
<tr>
<td>15.6 The Results .................................................................................... 214</td>
</tr>
<tr>
<td>15.7 Discussion ...................................................................................... 214</td>
</tr>
</tbody>
</table>
C.3.2 The Inspection Code Generator ................................................. 260
C.3.3 The Inspection Machine Controller ............................................ 260
C.4 Discussion .......................................................................................... 261
C.4.1 The Inspection Machine Planner .................................................... 261
C.4.1.1 Data Retrieval ........................................................................... 261
C.4.1.2 Operation Type Determination ............................................... 262
C.4.1.3 Setup Determination ................................................................ 263
C.4.1.4 Operation Data Determination ................................................. 264
C.4.2 The Inspection Code Generator ................................................. 264
C.4.3 The Inspection Machine Controller ............................................ 264
C.5 Conclusions ....................................................................................... 264
C.6 The CMM Part Program ................................................................... 266

Appendix D - The Users Guide ........................................................... 269
D.1 Introduction ...................................................................................... 269
D.2 Before running the software ............................................................ 269
D.3 Entering the Product Modelling environment ................................ 270
D.4 Entering the Inspection Planning environment ...................... 271
D.4.1 Initialize ....................................................................................... 272
D.4.2 Get CMM Data ........................................................................... 272
D.4.3 Create CSG tree .......................................................................... 272
D.4.4 Get object bounds ....................................................................... 273
D.4.5 Decompose object ...................................................................... 273
D.4.6 Get Centre of Gravity .................................................................. 274
D.4.7 Draw Wire-frame ........................................................................ 274
D.4.8 Draw SDSM ................................................................................. 274
D.4.9 Walk Relationship Graph ............................................................ 274
D.4.10 Plan Inspection ........................................................................ 275
D.4.11 Generate Part Program ............................................................. 276
D.4.12 Get Measurements .................................................................... 276
D.4.13 Return to Main Menu ................................................................ 276
D.5 The Inspection Planning process ...................................................... 276
D.5.1 Retrieving surface data ............................................................... 276
D.5.2 Rejecting relationships with invalid surfaces ......................... 278
D.5.3 Generating Probe Approach Directions ..................................... 278
D.5.4 Rejecting relationships with invalid approach directions .......... 281
D.5.5 Identifying datum setting operations ........................................ 281
D.5.6 Identifying measuring and probing operation types .................. 282
D.5.7 Identifying the number of sub-operations .................................. 284
D.5.8 Setting best approach directions ............................................... 284
D.5.9 Identifying setups ....................................................................... 285
D.5.10 Identifying master probing operations ...................................... 287
D.5.11 Identifying operation groups ..................................................... 288
D.5.12 Identifying the probing data for each sub-operation ............... 289
D.5.13 Identifying probing operation reference numbers ................... 290
D.5.14 Identifying sub-operation links ............................................... 290
D.6 The Part Program for the first Setup ............................................. 292
D.7 The Part Program for the second Setup .......................................... 295
Appendix E - Glossary ................................................................. 298
Figures .......................................................................................... 304
Chapter 1

Introduction

Arguably the most significant manufacturing concept that has emerged in the last decade has been Computer Integrated Manufacturing (CIM). CIM is an advanced business philosophy that demands close interaction between every aspect of the manufacturing system from sales and marketing, through design and manufacture, to delivery and after-sales service. It is a concept whose major objectives are to improve quality levels and increase manufacturing efficiency.

One area of business that is particularly affected by this philosophy of integration is the Design to Manufacture environment. Many companies are experiencing the substantial benefits of shorter lead-times and higher levels of quality through introducing advanced CAD/CAM systems into their design and manufacturing departments. However it is argued that even greater gains can be made by making more use of the data that is generated by the CAD/CAM activities. The Product Model concept represents a basis for a more efficient Design to Manufacture system by providing a data structure that describes both the component and its method of manufacture. The Product Model allows CAD/CAM applications of different origin to be integrated, simply by interfacing them to the common data structure. It also allows a higher level of automation to be achieved by the CAD/CAM applications as more detailed component data is readily available and also more generated data can be represented and used by other applications.

This thesis describes research which is part of a larger project investigating the effect of Product Model Data on a skeletal Design to Manufacture Environment. The so-called skeletal Design to Manufacture Environment is realised as a prototype software system that allows a user to design a single component, machine it on a three-axis vertical machining centre, inspect
it on a Coordinate Measuring Machine (CMM) and then analyse the inspection results to correct any errors. This skeletal system represents only part of the functionality of a full Design to Manufacture system, but does provide a practical environment for carrying out experiments relating to the beneficial effects of Product Model Data.

That part of the work addressed by this thesis is concerned with the effect that a Product Model has on the automation of the Inspection Planning and Inspection Code Generation tasks which together represent the inspection activity of the Design to Manufacture process. Inspection Planning is an activity that determines how the component should be inspected by analysing the geometry and by reference to various rules and algorithms, whereas Inspection Code Generation is a down-line activity that uses the Inspection Plan to generate a part program suitable for a specified CMM.

Most manufacturing tasks need some degree of planning and one as complicated as dimensional inspection is certainly no exception. Even though the objective of inspection is apparently straight-forward, that is to determine the values of a number of dimensions on a component, a great deal of planning is required to ensure that the measurements are as accurate as can be reasonably expected from the machine. The time factor of Inspection Planning is of secondary importance but nevertheless must be seen in the light of the increasing pressure to reduce batch sizes to satisfy the demands of the market-place. Ironically there is little difference in the time taken to plan the manufacture of a batch of one or ten thousand and so ways must be found of reducing the planning time to make the smaller batch more competitive. In a machining environment this pressure is felt more keenly by Inspection Planning as some success has already been achieved in the automation of planning for machining. This has been achieved with the emergence of generative process planners and the increasingly more sophisticated computer-aided part programming tools, but this progress will eventually be limited by the scarcity of similar tools for the Inspection Planning activity.
Chapter 1 - Introduction

The continuing search for higher levels of quality is also having an important effect on Inspection Planning. It is recognized that the most effective way of improving manufacturing quality is to manufacture the component right first time. This is a major objective of process and machine planners in a quality-conscious company. However it is not always possible to completely avoid the risk of defects and therefore the next best approach is to inspect the component as close as possible to the source of a potential defect in the manufacturing process. This approach undoubtedly improves the quality of the component but does increase the inspection frequency and hence the load on the Inspection Planning activity.

Inspection Code Generation is under the same pressures to reduce cycle time and increase quality levels as Inspection Planning. However it does have the additional requirement of the flexibility to generate part programs for CMMs of the same type but with different programming languages. This opportunity does not arise in the conventional environment as the part program is often the Inspection Plan and would be awkward to translate into a different CMM programming language. With an automated system the Inspection Plan is machine-independent and so part programs can be generated for machines of any origin.

This thesis describes the research into Inspection Plan and Code Generation with respect to the rest of the work of the project, as it has played a fundamental rôle in both the shape of the Product Model and the effectiveness of the Design to Manufacture system.

The survey of literature therefore includes the work of other researchers who are investigating Product Modelling Environments as well as work based on advanced Design to Manufacture systems and finally focuses on those concerned with Inspection Plan and Code Generation.

The main body of the thesis begins by stating and explaining the objectives of the research and the issues that need to be addressed in order to meet these objectives. This is followed by a description of the experimental environment of the work which includes an
explanation of the Product Model and the interaction between it and the inspection application.

Each aspect of Inspection Plan and Code Generation is addressed in detail with frequent references to case-studies which can be found in the appendices. These case-studies were based on components supplied by the industrial collaborators of the project and were processed by an experimental software package modelled around the ideas promoted by this research.

The thesis is concluded by a discussion of the extent to which the research objectives were satisfied and a summary of the research achievements.
Chapter 2

Geometric and Product Modelling

2.1 Introduction

The ability to represent the geometry of a product in a computer-readable form has arguably been the most important development in the continuing automation of design and manufacturing applications. The geometry represents a significant proportion of the information required to describe a product and is essential for manufacturing applications such as process planning, NC code generation and inspection planning. However, as researchers strive to increase the automation of these applications, it is increasingly becoming clear that greater levels of information are required. For example, if a process planning system is expected to determine the manufacturing processes required to produce a part then it must have access to tolerance information, which is currently represented in only a handful of geometric modelling environments. The logical extension of this argument is, of course, the modelling of all information required to design and manufacture a product and this concept is embodied by recent research into Product Modelling Environments.

The objectives of this chapter are in the first instance to survey the literature relating to geometric modelling and tolerancing, because of their recognized importance in the automation of inspection planning and secondly to summarize the literature that has so far been published in the emerging field of product modelling.

2.2 Geometric Modelling

Ever since Stone-age Man used flints to make tools and implements four million years ago, some form of geometric modelling has been used for design and manufacture. In the earliest times, the products were simple and the geometry could either be
remembered by the designer or could take the form of a previously manufactured product that others could copy [1]. Gradually as products increased in complexity, sketches were made both to assist the design process and to represent the finished product for manufacture. Documentary evidence of this trend appears much later in the fifteenth century when Leonardo da Vinci produced detailed technical sketches of prototype aircraft. However it was not until the Industrial Revolution that the roots of our current technical drawing practices can be seen, with the emergence of multiview parallel projection as shown by Trevithick in his designs for steam engines. The one element of current draughting practice that did not find usage until the beginning of this century was that of tolerancing. Until this time, drawings were scaled so that their dimensions could be retrieved through the use of proportional dividers. The theory of geometric and true position tolerancing appeared before the Second World War but has only gained wide acceptance in industrial practice since the relevant standards emerged during the 1970s.

If the Industrial Revolution accelerated the pace of design technology from simple sketches to the roots of drawing practice as we now know it, then the introduction of computers into industry during the 1950s had a similar accelerative effect that has yet to reach its peak. Computers have an enormous impact on both design and manufacture, which manifested itself in the first instance as simple computer graphics for design and Numerical Control for manufacture. These were major break-throughs because they promised to automate the menial tasks of design and manufacture allowing engineers to devote more time to the creative side of their work. On the design side the early work [2] soon separated into four largely independent streams of development [3] as shown in figure 2.1.
2.2.1 Wire-frames

The wire-frame approach to Computer-Aided Design is closely based on the traditional drawing techniques and as such it is very popular with experienced designers. The first wire-frame systems were simple 2D systems that were used for designing PC boards and mechanical parts in multiple views. The user would design a part by linking together lines and arcs that were stored as lists that could be scanned by application software. Typical applications for these early wire-frame representations were part programming systems, which would ask the user to drive cutters to lines and arcs selected by the user with the cursor [4]. These part programming systems were highly successful in their semi-automatic role and are still very popular in manufacturing industry. Three dimensional wire-frame systems appeared in the early 1970s, when the 2D lines and arcs were generalized into 3D space curves, which could be computationally projected to produce multiple orthographic, isometric and perspective views. This development greatly aided the visualisation of objects but also marked the peak of wire-frame technology mainly through the inherent lack of strength of the internal representation, which meant that it was possible to create both ambiguous (figure 2.2) and nonsense objects (figure 2.3) [3]. It was also difficult for manufacturing applications to make much sense of the object description [5]. The visualisation of wire-frame objects was also flawed when three dimensional views of curved surfaces were attempted (figure 2.4) because the wire-frame is literally a list of lines and arcs that represent edges of surfaces and it is therefore not possible to add view-dependent lines for visualisation [3]. Even with these limitations, the wire-frame representation remains popular in industry as there are so many designers that are trained to use this type of system and also because so many parts have already been defined using this representation.
2.2.2 Polygonal Schemes

This type of representation is based on the principle that polygons can be used to represent objects in an approximate sense, which although not accurate, can be used to visualize quite complicated objects. It is for this reason that this technique finds applications in flight simulators, animation and research into visual perception [6]. Research has concentrated on fast algorithms and dedicated hardware for generating displays from lists of polygons. Some of this work has been adopted in solid modelling systems for producing approximate Boundary Representations quickly.

2.2.3 Sculptured Surfaces

The sculptured surfaces representation was one of the first computer-based representations to appear and marked a watershed in the definition of components that could previously only be approximated using drawing board techniques [4]. The first design applications appeared in the 1950s and 1960s when Coons and Bezier and a few other researchers sought to replace the lofting and clay modelling practices of the aeronautical, marine and automotive industries with computerized descriptions of doubly-curved surfaces [3]. Since then there has been a significant amount of research carried out in the support [7] and architecture [8] of such systems.

2.2.4 Solid Modelling

Research into solid modelling began in the early 1960s when several systems were built that were of an experimental nature. Formal solid modelling theories began to appear during the 1970s from European, Japanese and American Universities and in the early 1980s the first generation of commercial solid modellers came on to the market [3,9].
Chapter 2 - Geometric and Product Modelling

From the start the major difference between Solid Modelling and the other forms of Geometric Modellers was the fact that the representation of the object was valid and unambiguous and was therefore a true representation that could be used by automatic applications [9,10]. Two solid representations have been researched: Boundary Representations (B-rep) and Constructive Solid Geometry (CSG) as shown in figure 2.5. B-reps are built from collections of surfaces, edges and vertices that are linked together in a topologically consistent way [11]. In CSG the geometry is represented as a tree structure whose leaves are simple solid primitives which interact under the control of Boolean operators at the nodes [11].

For completeness, it is worth mentioning four other solid representations that are used in addition to the two main representations. In spatial enumeration the object is represented as a union of quasi-disjoint box-shaped cells that may vary in size and are usually organized into quadtrees for 2D objects and octrees for 3D objects [12,13,14]. This type of representation appears to be particularly receptive to the type of queries common in machining and inspection planning but suffers from being computationally intensive. Special hardware solutions promise to alleviate this problem [10]. Cell decomposition is a technique that is mainly used in the field of finite element meshing and again is a union of quasi-disjoint cells which may have varying shape providing they are homeomorphic to a sphere [6]. Sweeping is commonly used to model the movement of industrial robots and other such applications as it is based on the sweeping of an area or solid through a spatial trajectory to generate another solid [6]. However it does not enjoy a wide geometrical domain and for this reason is seldom used as a primary representation, but because it is nevertheless a powerful method for describing certain types of geometry it is often used as a volatile input mechanism [3]. Finally primitive instancing is a formalization of the family-of-parts concept where a
solid is instanced from a parametrically defined member of a family, such as a single diameter round shaft with oil-grooves [6].

2.2.5 Problems with Solid Modelling

Although it is now accepted that solid modelling is the only viable means of geometric representation for automated manufacturing applications [4,6] it is also understood that there are limitations to the original solutions that must be circumvented before the full value of the concept can be exploited. The two major limitations are firstly the implicit restrictions of supporting a single geometric representation, which have now been countered by the use of multiple-representation and hybrid modellers, and secondly the difficulty of integrating a satisfactory tolerance model with the geometry [8,10,15]. Both of these subject areas play a crucial rôle in current Product Modelling trends and therefore merit more detailed discussion in the following sub-sections. However there are a number of other limitations, the foremost of which being the limited geometric domain of commercial solid modellers [6] that does not yet extend to modelling sculptured surfaces and therefore renders the modeller incapable of representing the complete geometrical spectrum. The PADL-1 development team at Rochester found that roughly 40% of the parts designed by a range of companies could be represented by block and cylinder primitives, whereas 90% of components could be modelled when further primitives (cones, spheres and tori) were used [13]. The remaining 10% of components were found to require sculptured surfaces. It has also been found that certain configurations of geometry can be modelled using primitives but are particularly difficult to do so; these include the blends and fillets which frequently occur on machined, cast, moulded and forged parts [10]. One possible solution to this problem, which could also encompass the separate problem of thread modelling, is to label the geometry using notation analogous to a note on a
drawing [9]. Solid Modellers are also slow through the computationally intensive nature of their operation, but this is likely to become less of a problem in the future as critical algorithms become more efficient and the trend of increased computing power continues [6]. However there is a subtle adjunct to this problem which concerns the complexity of the parts that industry would like to model, which appears to be increasing faster than the ability of the commercially available systems. This is due in large part to the fact that current systems consider that all parts of the definition have equal significance, from a small hole to the largest macro-feature [6]. Humans cope with this by ignoring those parts that are irrelevant and those which are too detailed and it is anticipated that future research will explore similar avenues of thought.

2.2.6 Dual Representation and Hybrid Solid Modelling

The experimental solid modellers of the 1970s normally had a single representation scheme, which was either boundary type, or CSG. Either of these schemes were powerful in their own right, but it was evident that they had their individual strengths and weaknesses. For example, CSG representations are concise and guarantee a topologically consistent and bounded model, but it is computationally expensive to generate wire-frame displays [11]. Boundary representations, on the other hand, require a large amount of data and care must be taken to ensure that they correspond to valid physical models, but they are well suited to the generation of hidden-line displays and they contain face-relationship information that is useful in various applications including assembly [11]. As a result of this, dual representation modellers were developed, which incorporated a primary representation that could be converted into another (or several others in the case of multiple representation schemes) auxiliary representation depending on the application. Two schools of thought have emerged as to the type of
primary internal representation that should be used but as yet there is no general consensus as to which of the two architectures is superior [3]. Those who argue the case for an primary representation based on CSG supported by a boundary representation state that:

- CSG is concise
- it guarantees that objects are valid
- CSG and boundary representations can be constructed with the same large domain
- reliable algorithms for converting CSG to boundary representations are known
- efficient CSG-based algorithms exist for certain applications such as computing mass-properties
- good CSG-based interfaces can be designed

Conversely, those who believe that a boundary representation should be the primary representation argue that:

- boundary representations are informationally complete and relate more closely to traditional draughting techniques
- for the applications studied to date boundary-based algorithms are competitive with those based on CSG
- dual primary representations cause difficult consistency maintenance problems and preclude direct operations on boundaries (e.g. chamfering edges)
- available sculptured surface technology is surface-oriented and therefore may be easier to incorporate in boundary-based systems than in CSG-based systems.

Substantial research has been directed towards the maintenance of consistency between the representations, and the possibility of converting only parts of the model
when, for example, a small engineering modification has been carried out which does not merit the reconversion of the complete model.

There is some inconsistency in the literature regarding the difference between dual-representation and hybrid schemes [8] but for the purposes of this discussion it will be assumed that a scheme is termed hybrid when generically different facilities or constructs are incorporated in the same structure in order to widen the domain. For example, the solid shown in figure 2.6a cannot be represented by the simple sweeping representation as the object is not translationally or rotationally symmetric [14] (figure 2.6b). For this reason the additional gluing facility, which is a restricted form of set union, can be incorporated to broaden the domain (figure 2.6c) [3,14,16].

The overview of dual representation and hybrid schemes presented in this section is necessarily brief but it is worth noting, in conclusion, that these schemes form part of a major strand of research into the architectural issues of solid modelling [8], which promise to revolutionise the power and applicability of such systems.

2.3 Tolerancing

Dimensional and geometrical tolerances are the means by which the designer controls the function, quality, reliability, manufacturability and cost of a product [17] and as such they play a critical rôle in the product model. However, it is clear that although traditional design and manufacture methods take account of tolerancing practices, very few computer based design, process planning and code generation systems make any provision for their integrated application.

Conventional linear and angular dimensions and tolerances have been in use since the beginning of the century [6] but it was only after the Second World War that the standards on this subject were formulated [18]. Currently the use of dimensional
Chapter 2 - Geometric and Product Modelling

and geometrical tolerances is principally controlled by the BS308 series in the UK [19,20,21], ANSI Y14.5M in the United States [22] and various ISO standards. They describe general engineering drawing practice, dimensioning and tolerancing of size and geometrical tolerancing for form, attitude, location and the composite tolerance of run-out. The practice of geometrical tolerancing emerged during the early '60s [18] and is generally accepted to be an additional and powerful form of constraint as shown by the example given by Gladman [23], which constrains the position of a hole with respect to two faces. Figure 2.7a shows the most simple method of constraint, where the centre-line is independently constrained by the minimum distance to a point on each of the two planar surfaces. This definition does not consider any deviation of form on either of the two surfaces or any deviation of the perpendicularity between them. Figure 2.7b shows a system of constraint based on geometrical tolerances where the true position of the centre-line is constrained within a rectangular system of coordinates, which are constrained to be flat and perpendicular to each other. The practice of geometrical tolerancing has recently enjoyed wider usage in the engineering community both for the reasons given above and also because computer controlled measuring machines are being deployed that are capable of measuring geometrical tolerances quickly and automatically. However it has been noted that there are serious discrepancies between the results obtained from different measuring machines [24,25] and this has been explained by differences in the types of software algorithms used by the machine controller.

In spite of the well intentioned and diligent efforts of the various standards committees, the tolerancing conventions for dimensional and geometrical tolerances remain open to misinterpretation because there is no mathematical theory to ensure unambiguity, uniqueness, consistency or completeness within a product drawing [26].
In fact it has been argued by many researchers [4,17,25,27,28] that the current standards are insufficient and need to be reconsidered from first principles.

Two of the main areas of research involving dimensional and geometrical tolerances (hereafter known as constraints) include representation schemes for constraints within geometrical modellers and the analysis of tolerances for design and manufacture. Both strands of work are attracting increasing degrees of interest and are widely acclaimed as critical to the effectiveness of CIM and product modelling solutions.

2.3.1 Representation schemes for constraints

The previous section (2.2) described geometrical modelling systems and argued that solid modellers will undoubtedly form the basis of the geometric requirements of automatic manufacturing applications, primarily because solid models are unambiguous and cannot describe nonsense objects. Automatic manufacturing applications, most notably inspection planning, will also require relationship information and, for the same reasons as above, this must also be represented in such a way that it is unambiguous and realizable.

At first glance this does not seem a difficult problem and several commercial systems have implemented solutions that allow the designer to attach constraints as textual annotations to the geometry [29]. This approach is based on conventional tolerancing practice, which has the advantages that it is simple and well understood, but by the same token it suffers from the same problems of mathematical ambiguity as described above and therefore does not extend beyond the limitations of the current approach.

Probably the earliest and most comprehensive research in this field was carried out by Requicha [30] in his theory for tolerancing, which is based on the assumption
that an object satisfies a tolerance specification if its boundaries are within defined regions of space called tolerance zones. However other researchers [24,31,32] have found that although the theory is a worthwhile and necessary attempt to formulate a theory for tolerances it is flawed by the inadequacy of its description of part size and position tolerances with respect to functionality.

Ranyak and Fridshal [27] propose a feature-based hierarchy, which incorporates:

i) toleranced features at the primitive level, ii) design and manufacturing features at the next level, which are themselves composed of toleranced features and iii) group technology type features at the highest level. Each toleranced feature has a type of tolerance and pointers to each surface, which are qualified according to the portion of the surface that is constrained. The system is configured so that it is impossible for the designer to construct invalid tolerance situations with reference to ANSI Y14.5M and it has been tested on the XPS-2 process planner from CAM-I.

Gossard and Zuffante [33] have researched a hybrid solid modelling system that allows dimensions to be represented by a relative position operator. This approach confers the advantages of dimensionally driven geometry where the designer can alter the size and shape of the object by adjusting the values of the dimensions. Previously the designer would be obliged to edit the surfaces, lines and arcs within the geometrical environment before adjusting the dimensional values to suit. However there are important issues to be observed with this sort of approach as it is possible to inadvertently create a nonsense object as its topology (the numbers of faces, edges and vertices and their connectedness) will remain constant while the dimensions are changed. This can be seen by imagining a simple object constrained by a number of constraints (figure 2.8a), where relationship B can be adjusted in order to fine-tune the design (figure 2.8b) but if the variation exceeds a certain margin (figure 2.8c) then the object
becomes non-realizable. For this reason a variable topology modelling environment was researched based on a dual-representation modeller combining CSG and boundary representations.

Hillyard and Braid [34] describe a mechanical solution to the representation of constraints, which models the edges and vertices of the object as extendible members and pin-joints. Dimensions are represented as stiffeners which fix the length of members or the angles between them depending on their type and tolerances are modelled as the stiffness of the dimensions. Once a dimensioning scheme has been specified for a component it can be verified by building and analysing a flexibility matrix, which determines whether the object is over or under-constrained.

Imamura et al [16] are concentrating their efforts in the area of product modelling and as such are considering the representation of both the geometry and the constraints. The geometry is represented in the form of a G-rep (gluing representation), which is based on the gluing of linearly and vertically swept plane figures. The dimensioning scheme is loosely based on Hillyard's stiffener concept but checks for over or under dimensioning by the use and analysis of flags at each node. The system also adheres to the concept of Dimension Driven Geometry championed by Gossard and Zuffante, which allows the designer to modify the geometry by editing the dimensions. This is done by constraint propagation where the dimensions are represented as first order linear equations, the variables of which correspond to the coordinates of 3D geometrical objects. When the dimensions are changed, the system generates a new set of equations and solves them so that the revised coordinate values can be passed back to the geometry.

There have been many other attempts at defining a theory for modelling tolerances but as yet a comparative study of the strengths and weaknesses of these
approaches does not exist [29].

2.3.2 Analysis of tolerances for design and manufacture

Tolerance analysis is a technique used to reconfigure the designer's relationship scheme so that it is in a form suitable for manufacturing processes such as assembly or machining [26]. This is a frequent requirement as the tolerance information necessary to determine assembly or machining operations is rarely available from the original design and must therefore be calculated.

Weill [17] suggests a tolerance transfer method that optimizes the manufacturing tolerances using a tolerance matrix so that the largest tolerance range is used, which therefore reduces the cost of the operation.

Xiaoqing [35] uses the tolerance chart technique, which provides the process planner with a precise method for establishing maximum allowable tolerances and an easy method for establishing the relationship between working dimensions and drawing dimensions.

Parkinson [36] has been working on two related techniques that firstly permit rapid assessment of tolerances on a set of component dimensions by indicating the magnitude of risks of unsatisfactory assembly and secondly by optimizing the tolerances to minimize the overall cost of manufacture (to a given accuracy) and assembly (including cost of rejection).

2.4 Product Modelling

Product modelling has been practised in the literal sense since man started to make objects by either remembering the design, materials and method of manufacture or by constructing a physical model that could be copied. The concept is now becoming more formalised as it has recently attracted an increasing amount of attention from
an engineering community that is striving for more efficient ways of integrating design and manufacture.

The concept of Product Modelling facilitates integration because it is a data structure that completely describes the information required to design and manufacture a product [16,37,38,39]. This makes it possible for design and manufacture applications of any origin to be interfaced to this data structure so that they can retrieve the data they require and write back the data that they generate.

2.4.1 Neutral File Formats

This type of approach to integration is not entirely new as it has been embodied by the IGES standard for integration between CAD systems since the late 1970s [40]. IGES is the Initial Graphics Exchange Specification and is a data format for describing product design and manufacturing information [40]. It is used to facilitate the exchange of data, which has been created and stored in a CAD/CAM system in a computer readable form. The format is designed to be independent of all CAD/CAM systems, hence the term neutral, but to have sufficient functionality to capture all the information likely to be stored in the current generation of CAD systems. The early versions of IGES were capable of representing three-dimensional wire-frame models with planar or curved surfaces and included a range of standard dimensioning formats. Later versions were revised to permit the modelling of CSG and boundary representations.

Reports from users reflect a mixed reception to the standard ranging from 100% effective to totally ineffective. This is supported by a recent project devised by the Society of Motor Manufacturers and Traders (SMMT), which identified some seventy problems, from which a few were attributed to the inadequacies of the specification, approximately one third were due to the interpretation of the specification and the rest
Chapter 2 - Geometric and Product Modelling

were software errors in either pre or post-processors [41]. The solutions to many of these problems have been found through thorough validation testing of translators, which typically involves verifying the transfer of each entity in ascending order of complexity [42]. Other validation methods include reflection testing [42], which involves sending a model through a pre-processor and feeding it back through a post-processor to examine any differences and falsification, which attempts to show that an implementation is incorrect [40].

Several other neutral file formats are used in industry and these are described below. The Data Exchange Format (DXF) is a vendor specific format developed by Autodesk Inc. in the United States for their AutoCAD product, but is limited to 2D models. The German VDA-FS standard was designed for the exchange of free-form surface data and was developed by the automobile industry as an alternative to IGES, which at the time of its inception could not represent precise surface definitions. The French Standard d'Exchange et de Transfert (SET) was developed to provide a fast and efficient translation, to be easily expandable and to minimize the required storage volume. SET supports the same entities as IGES but has been shown to have superior performance [41]. The Product Definition Data Interface (PDDI) was initiated by the US Air Force ICAM project and goes a step beyond the IGES standard by incorporating more manufacturing information, such as machined, turned and sheet metal features, geometric tolerances and datum positions. It was the first specification to mention product data and product life cycle data [40].

2.4.2 PDES/STEP

The ISO sub-committee TC184/SC4 was formed in July 1984 to centralise all data exchange development and decided that the needs of industry could not be satisfied by enhancing any one standard and that a new specification should be developed which is
now known as PDES/STEP. The American Product Data Exchange Specification (PDES) and the identical [43] European Standard for the Exchange of Product Model Data (STEP) have now reached the stage in standardisation where they can be registered as a Draft International Standard [41].

The objective of PDES/STEP is to represent a complete product model with sufficient information content to facilitate interpretation by advanced CAD/CAM applications, which may include generative process planning, group technology coding, finite element analysis, NC code generation etc. [44]. It has also been designed to provide the data exchange requirements for a variety of disciplines, including architecture, mechanical engineering, construction and electronics on any one of four levels:

- a passive file exchange, which is the more traditional level of exchange processor
- a working form exchange in which the data is translated into and out of a temporary working database, accessed using standard software function calls
- a database exchange level, where product data is translated into and out of a database management system
- the Integrated Product Knowledge level, where product data is translated into and out of a knowledge-base management system.

Translators are not expected to be available for at least two years after delivery of the standard, but they are expected to be able to handle sophisticated and complex translation tasks and in so doing will supersede all existing data exchange standards [41].

2.4.3 Product Modelling Environments

Most of the academic research in the area of product modelling has been directed towards establishing product modelling environments and experimenting with the
integration of automated applications. The following sections describe some of this work.

2.4.3.1 University of Tokyo

A significant amount of research into product modelling environments has been carried out and reported by Tokyo University [37,45,46] and this section aims to provide a summary of the direction their work is taking. They have defined a product modelling environment that uses first order predicate logic and object-oriented programming techniques in a structure that is comprised of basic, engineering and application models (figure 2.9). The basic model is used to describe the fundamental characteristics and properties of a object, which are likely to include the geometry, mathematical formulae for describing constraints, finite element mesh models and any other information that is deemed to be independent from particular engineering processes. The engineering model is used to represent all the basic engineering knowledge, which includes dimensions and tolerances, assembly and kinematic relationships, materials and manufacturing methods etc. The application models are previously defined models of products, which can be used to assist the modelling of similar future products [37]. One of the earlier case-studies that they carried out was based on a computer disk drive spindle assembly, which they were able to define within their product modelling environment as it is equipped with tools to describe face and part kinematic relationships. An assembly verification application is supported, which checks that there is at least one sequence in which the product can be disassembled into its component parts [37]. Later work [45] concentrated on constraint propagation and variational design, which allowed the designer to translate the product specification into a set of engineering constraints that were subsequently propagated through the layers of the model described above until they were simple enough to be satisfied by
primitive design solutions. This theory is supported by a simple case-study based on the design of a material handling device that must satisfy certain performance criteria. The most recent work of this laboratory [46] continues the theme of variational geometry, which is embodied in a new modelling system designed to solve the limitations of their previous modeller GEOMAP-III. These limitations were centred around the difficulties experienced with propagated constraints that were either too complicated or too weak to be converted into geometric solutions. A revised modelling system was based on GEOMAP-III but incorporated geometric constraint representations in a PROLOG-like sub-system, which allows the geometry to be controlled by simple constraints. Good examples of the power of this facility were given by models of flexible objects such as wires or textiles, where the constraints were based on physical characteristics such as the effect of gravity, stiffness, deformation characteristics etc. Another interesting feature of the variational modeller was the ability to model products in a combination of geometric representations, which permits the designer to impart as much or as little detail as necessary to the product.

2.4.3.2 University of Leeds and Loughborough University of Technology

The research described in this thesis formed part of the Information Support Systems for Design and Manufacture research project, which was funded to investigate Product Modelling Environments and the benefits that they can impart on the integration of highly automated design and manufacture applications. The purpose of this section is to describe the research into product descriptions systems and product data models.

The interest in product modelling arose from previous research in geometric modelling systems and the realization that geometry alone is insufficient to support the requirements of engineering applications. This led to the development of a prototype tool for handling structured data, known as the Structure Editor (SE), which relies on
the principle of the meta-structure to define the structure of data, which is itself
defined by another meta-structure that defines the data modelling capabilities of the
SE [38]. The SE facilities include:

- unstructured objects
- relationships
- abstractions
- networks or hierarchies
- derivation/inheritance
- insertion/deletion/modification constraints
- degree of expression of relationship semantics
- dynamic modelling
- sharing
- parametrization

Although the SE has been designed as a generalised tool for defining data struc-
tures it has proved particularly appropriate for defining the sub-set of Product Data
Models and in this guise forms the basis of a Product Description System (PDS) [47].
The PDS allows the user to define product specific data, company specific data and
reference data. The product specific data applies to a single product, for example its
geometry, whereas the company specific data applies to more than one product, for
example the manufacturing facilities. Reference data is more general, for example
material properties and data which is held in standards. The PDS also provides a
parametric definition capability, which is claimed to be more purposeful than the
current range of parametric modellers on the market. Whereas the typical parametric
modeller allows the user to alter only numeric parameters, the PDS environment
permits the user to define and alter structures that are controlled by both simple (integers, reals and strings) or structured (lists, choices and records) parameters [47]. An example of this capability is given based on a component that has been defined in terms of its geometry, material and cost. A mild steel component would have the cost and geometry as parameters, whereas a turned mild steel component would have costs and its profile as parameters and so on. Finally the researchers involved in this work have proposed a CIM Applications Architecture (CAA) that represents the long-term view of the work (figure 2.10) [47]. The shaded central box represents the architecture into which closely-coupled applications are integrated, which are assumed to be written with an intimate knowledge of the this CIM architecture and will able to fully exploit its capability. They also argue that there will significant use of software systems, which are implemented outside of the CAA for at least the next decade and such systems will need some customised interface through which they can be loosely-coupled [47]. The capabilities offered by these customised interfaces is anticipated to vary widely from simple data awareness flags through to more powerful data translators that will permit bi-directional communication.

2.4.3.3 Research Elsewhere

Spur et al at the Technische Universität of Berlin have also investigated the architectural issues of product modelling [48] and have defined a product model concept that incorporates an organization layer, an information layer and an information link-layer (figure 2.11a). As figure 2.11b indicates there may be many information layers, each of which represent information regarding a certain characteristic of either the product or its manufacturing environment such as geometry, functionality, manufacturing methods and facilities, historical and statistical data etc. Each information layer can be interrogated via the information link-layer and the organization layer both controls this
activity and represents a logical view of the product which can be related to other product models. This architecture is claimed to be capable of supporting hybrid geometrical models by installing each representation as an information layer, which is interconnected to the other representations by the information link-layer. By the same token, variational geometry can be supported by incorporating a variant information layer in each part model, which is linked through the organization layer to the variant layers of other parts in the assembly. Therefore if the product is altered the modifications can propagate down the assembly structure to each affected part. A simple case-study involving a product model based on a machine tool is used to illustrate this concept.

Imamura et al of the Yasukawa Co. Ltd. have implemented a product modelling system using the object-oriented approach [16], which supports a hybrid two and a half dimensional solid modeller and a dimensioning capability that attaches dimensional information to geometric entities. The dimensional information can be analysed for insufficiency or redundancy and the geometry can be modified by altering the dimensions using the constraint propagation method.

Finally Björk describes a product modelling environment suitable for the construction industry [49], which is reported here nevertheless, because it is a good example of the product modelling concept being used outside of the electromechanical industry. The model is designed to contain all the information required to construct and maintain a specific building, which includes the design, location, relations between building parts, physical properties, materials used and price. It should also be capable of expansion and alteration throughout the construction process so that at any one time it can give an accurate reflection of the building's condition. The model also embodies the concept of an abstraction hierarchy, which allows the designer to employ top-down design methods, which are particularly suited to the con-
struction process. This means that at the briefing stage of construction the customer can be presented with an accurate but high-level description of the building, which is only detailed during the subsequent design, construction and maintenance processes.
Chapter 3

The Design to Manufacture Environment

3.1 Introduction

A substantial amount of research has been directed towards the engineering disciplines of design and manufacture since the 1950s with the emergence of both computer graphics and the first programming languages for the then new NC machine tools [6]. Computer-Aided Design (CAD) has developed significantly since then, from the early wire-frame systems through to the current powerful solid modelling systems that are enjoying an increasingly wider domain (refer to chapter 2 for a more substantial discussion of this subject). Computer-Aided Manufacturing (CAM) has evolved in a similarly rapid fashion but on an even wider scale with a broad range of CAM products now available on the market, including process planning systems and NC code generators for milling, turning, grinding, wire-cutting, stamping and, in fact, for virtually any machine tool controlled by a computer. However the promise of fully automatic CADCAM systems and unmanned factories is still a long way from being realized and this has been explained by the lack of integration that exists between these and other advanced software and hardware tools. The objective of this chapter, therefore, is to discuss the research issues that are currently being addressed in CAD and CAM and to summarize the progress that has been made in integrating these disciplines.

3.2 The Design Environment

Design is the first major step in a product's life-cycle, and is arguably the most influential as it has a direct impact on the product's quality, functionality, manufacturability, saleability, serviceability and longevity [6]. In the light of this observation it
is therefore surprising that the creative part of the design process, where the designer converts a set of functions and constraints into a design that can be manufactured into a satisfactory product is still so poorly understood [6,50]. However at its highest level design can conveniently be separated into three stages [51] as shown in figure 3.1.

3.2.1 Specification

The first stage is to set the specification, which is essentially an agreement between the customer and the supplier and is usually based on some form of written contract, which may also include drawings, costs, conditional clauses, delivery dates etc. There are no directly applicable software tools available to automate the specification setting process, but if the product is similar to any previous products manufactured by the company, then CAD and process planning data-bases can be interrogated to provide information that may be relevant to the present specification [51].

3.2.2 Detail Design

This is the creative phase of the design process, where the designer uses his skill and experience in an effort to achieve the performance characteristics stated in the specification, whilst accommodating any design constraints that might have been applied. The result of this process is a finished design, which incorporates four coupled bodies of information [6]:

- ideal-form (shape) specifications for the component parts
- associated variational specifications (tolerances)
- component-combination specifications (assembly drawings)
- material and finish specifications.
How the designer chooses a particular ideal-form to satisfy a function is a mystery that has yet to be solved, but in the case of a completely new component it does appear that a substantial amount of trial and error is involved. Typically the designer will use his skill and experience to draw up an initial design proposal and this will be analysed using a variety of computer-based and practical methods so that any improvements can be fed-back to the design. In this type of environment CAD does provide an ability to rapidly draught the design but rather than substantially reducing the lead-time it has been observed that the use of CAD allows the designer to investigate a wider range of design proposals and also to attempt more complicated designs [2]. However conventional CAD products are not suited to the early conceptual stages of design [6] because they are not yet capable of allowing the designer to sketch ideas and also do not have the functionality to accommodate the sweeping changes in geometry and topology that a designer can carry out so easily with a pencil and notepad.

The design of a completely new component is a rare occurrence [52], because there may be similar components used in other products made by the company or more often some feature of the design is shared elsewhere. Under these circumstances the task is that much easier as CAD and Computer-Aided Process Planning (CAPP) data-bases can be interrogated to provide the relevant information [51]. This process can also be automated through the use of classification and coding systems that allocate a descriptive code to a component [53,54,55,56] so that similar or partially similar components can be retrieved from the part data-base along with the relevant design and manufacturing information. A logical extension of this concept is the use of part families, where a generic part is parametrically defined in terms of geometry and manufacturing method so that a new part can be instantiated simply by
supplying the appropriate parameters to the part family. The Canon Co. [50] use this type of approach for optical lens design where the parameters include the curvatures of individual lenses, kinds and thicknesses of the glass, distance between the lenses etc. Other examples of typical part families include mechanical bearings, fasteners, drill bits etc. More general parametric design tools have also been researched [46,47,52,57] and CAD systems with parametric capabilities are beginning to appear on the market [15], which allow the designer to define the part families described above and also to define segments of geometry, commonly known as form features, that can be controlled parametrically. This class of CAD environment is potentially very powerful as it also permits manufacturing knowledge to be attached to part families and form features, which can also be controlled parametrically and subsequently inherited by the finished design [5,57].

In conclusion, it is clear that current CAD systems do assist the designer to draught designs quickly and accurately and as will be shown in the next section they are more frequently being equipped with time-saving design analysis tools, but they do not yet have the capability to generate a design automatically from a set of performance and constraint criteria. However research teams are investigating the possibilities of generative design and so far some success has been achieved in the design of both software and VLSI (Very Large Scale Integrated circuits) [50].

3.2.3 Design Analysis

The ultimate method for design analysis is to fabricate the design and to observe, for example, whether, for example, the the product is strong enough or light enough to meet its performance and constraint criteria [6]. Unfortunately this is not always a practical solution as it is often expensive and the results may not be totally accurate if the product has had to be manufactured by an alternative method, which might occur
if the original method of manufacture required expensive tooling.

The ideal alternative is to analyse the design using computational methods, which allow the designer to examine a growing range of performance characteristics, including, mass-properties (if solid modelling has been used), tolerances (refer to section 2.3.2) and the effects of stress and strain.

The most celebrated analysis tool is the method of finite element analysis (FEA), which is used to analyse the effect of loading or deformation on a design and can also show, through colour codes, variations in properties such as thermal and electrical conductivity [53]. The underlying assumption is that the component can be described in the form of a structure of finite elements over which the unknown is considered to vary in a known continuous manner. This can be done because each element is treated as a simple beam, shell or rod to which classical mechanics theory can be applied [51]. Currently most systems require the user to generate the mesh manually, which is a tedious and error-prone exercise [10], but it is anticipated [6] that automatic finite-element mesh generators will be available during the 1990s based on the principles of spatial enumeration as described in chapter 2.

Another method of analysis available to the designer is that of flow analysis for products that need to be cast or moulded. A product like MOLDFLOW [4,58] allows the designer to simulate cavity filling, taking account of the material properties and the gate and runner configuration.

Finally product visualisation is often used to analyse aesthetically-sensitive products such as telephone hand-sets [58], which are manufactured using the injection-moulding process and are therefore difficult to prototype. Using a solid model representation, a colour shaded image can be generated, which can be viewed from any position under a variety of synthetic lighting conditions.
Chapter 3 - The Design to Manufacture Environment

3.3 The Manufacturing Environment

The objective of this section is to describe the current research and industrial status of Computer-Aided Manufacturing (CAM) systems that are directly connected with the Design to Manufacture environment. These systems therefore include Computer-Aided Process Planners and Automatic Cutter Path Generators and Verifiers.

3.3.1 Computer-Aided Process Planning (CAPP)

Process Planning is a critical activity in the link between design and manufacture [17,34,59,60] and it is argued that automatic process planning systems represent one of the cornerstones of the Computer Integrated Manufacturing (CIM) initiative [17,60,61,62]. For these reasons, CAPP attracts a substantial amount of academic and industrial research and the purpose of this section, therefore, is only to give a brief review of the trends and directions that are emerging. This section does not aim to survey individual CAPP initiatives as that information is already available in any one of several publications [59,63,64,65].

The purpose of process planning is to prepare a set of instructions that describe how to fabricate a part or build an assembly which will satisfy the design specification [66]. Usually the task is carried out in a sequence of steps, which starts with the planner analysing the part drawing in order to determine the types of processes that should be involved in its manufacture. This decision is based on a great many factors, not least of which being standard company practices, but including [64]:

- geometric configuration
- dimensions and tolerances
- surface roughness constraints
Chapter 3 - The Design to Manufacture Environment

- raw material properties
- batch size
- heat treatment and hardness

Once the types and sequence of manufacturing processes have been determined, the planner identifies the methods and operations that need to be carried out and this depends on the type of process involved. For the purposes of this brief overview, the process shall be assumed to be machining and in these circumstances the following items of information will need to be determined [66]:

- operation sequence
- machine tools
- cutting tools
- materials
- machining tolerances
- cutting parameters (speeds, feeds, depths of cut)
- jigs and fixtures
- operation times
- setup details
- inspection criteria
- inspection gauges

Computer-Aided Process Planning (CAPP) systems fall largely into two categories, the variant and the generative, but with the rapid development of new techniques, many CAPP systems do not fit exactly into either category and are therefore described as semi-generative [64].
3.3.1.1 Variant CAPP Systems

The underlying method used in variant CAPP systems is similar to that used by conventional manual methods in that a process plan for a new part is created by recalling an existing plan for a similar part and then, if necessary, by modifying it accordingly. The approach is based on the concept of Group Technology [67], which is a manufacturing philosophy that suggests that there are advantages to be gained from clustering parts into groups with similar attributes [51]. In the case of CAPP, the parts are clustered according to similarities in their manufacturing methods and are identified through the use of a classification code that describes the geometry and other relevant attributes. A master process plan is assigned to each part grouping and this is defined by an experienced engineer, who is thus able to specify a near-optimum plan and incorporate any appropriate company practices. Inevitably not every part will map directly to this master process plan and so editing facilities are provided, which are often little more than word-processor type editors [66]. The principle strength of the variant approach, therefore lies in its information management capabilities rather than in any data-processing or algorithmic methods, but this approach has proved remarkably successful mainly as a result of its inherent simplicity, which makes it easy to use and cheap to purchase. The disadvantages are that the quality of the process plan still depends on the knowledge and experience of both the engineer who created the master plan and any other personnel who made subsequent modifications, and this can lead to consistency problems [64]. It has been observed [68] that if several process planners are given the same part they will most probably specify different plans and, more alarmingly, if the same planner is required to generate a plan on different occasions then these too are likely to be different.
3.3.1.2 Generative CAPP Systems

The operating principle of generative systems is based on the use of rules and algorithms to analyse the component geometry in order to determine its method of manufacture automatically. This is often achieved through the use of form features, which are either used to construct the component geometry, or are extracted from a more conventional representation. Form features have the advantage in this context that they are inevitably less complex than the combined object and are therefore less difficult to handle.

Research into generative CAPP systems has proceeded on a number of broadly parallel fronts including:

- CAD/CAPP interface [61,62,67,69,70,71,72,73,74]
- feature extraction/recognition [62,70,71]
- process selection [69,70,75]
- process sequencing [61,75]
- dimension and tolerance interpretation [17,27,35,60]
- fixture design [76,77]
- operation sequencing [61,73]
- tool selection [74,75]
- cutting parameter determination [67,72]
- cutter path generation [71,72,74,75]
- inspection [71,75]

Despite the early progress made by various research teams, generative process planners are still at the experimental stage of their life-cycle and this is due in no small part to the complexity of the process planning activity. However this type of approach
promises to provide a fully automatic and consistent process planning tool and is attracting increasing attention from the larger companies that can both benefit from its strength in accommodating a broad and varied product range and can afford its extended implementation and development costs.

3.3.1.3 Semi-Generative CAPP Systems

As generative process planners are not yet a commercial reality, a third class of CAPP system is beginning to emerge that combines the concepts of both generative and variant planners [59,64]. This type of planning system is known as semi-generative and is perceived as a powerful transitional tool for use in industry until reliable generative systems become available. These systems serve to reduce user interaction through such features as standard operation sequences, decision tables, and mathematical formulae, but do not guarantee to generate a perfect plan. Instead they rely on user interaction to edit the plan, although to a lesser degree than the editing activity of the variant approach.

3.3.2 Cutter Path Generation

Once numerical control had been developed by the Massachusetts Institute of Technology and the US Air Force in the 1950s [2], it soon became clear that a high-level programming language would be required to accommodate the complex part geometries designed by the Air Force. They, therefore, supported a follow-up research project, which resulted in Automatically Programmed Tool system (APT); the precursor of all subsequent high-level programming languages. As well as removing the NC programmer from the drudgery of calculating the coordinates of tool paths, APT also had the advantage that it could generate code for any machine tool for which it had a post-processor. This was made possible by an intermediate language, known
as cutter location data (CLData), which later became an industry standard [78].

Since then several other similar languages have been developed (COMPACT, GNC etc.), but the first major change came with the arrival of simple wire-frame based CAD systems that allowed the NC programmer to drive the cutter around a geometric representation built by the designer using menus and a mouse. These systems still generated output in APT or COMPACT, which was subsequently converted into CLData that could be post-processed into NC code. They also had the advantages that the NC programmer could use the original geometry, so saving time and avoiding re-transcription errors, and that it was no longer necessary for the programmer to have expertise in a high-level programming language. There are now dozens of commercially available CAD-driven NC part programming systems, which are enjoying very wide usage in industry, mainly through their recent availability on cheaper Personal Computers. However, they are also approaching the peak of their development as they are essentially based on 1950s technology and are no longer attracting research interest, which has instead turned to fully automatic cutter path generation.

Automatic cutter path generation is still in the early stages of research and is consequently a long way from being a commercial reality. It has been observed that current research is following a variety of different solutions, which is reflected by the three representative systems described below.

Armstrong et al [79] base their approach on the analysis of a spatially ordered geometrical representation in order to derive cutter paths for side and end milling. This spatially ordered representation allows the object to be divided into either stock, part or semi-part cells so that low-level algorithms can be used to determine whether a particular tool path is viable or not. Using this method, they have been able to derive component setup information and roughing and finishing cuts, but they admit that the
system does not consider fixturing methods and does not necessarily generate an optimal cutting strategy.

Ssemakula and Sivac [68] describe a feature-based code generator that forms part of a generative process planning system, which determines the most suitable machine tool, the cutting parameters and the finishing operations for each feature and then determines the machining sequence using rules. The system can generate code in either COMPACT II or APT and this is done by issuing the setup statements from the machine tool database, the geometry definitions from the features and the cutter paths from subroutines attached to the features. Current results show that the system is capable of generating reliable code for workpieces with clearly defined features and an example is given of such a situation involving a simple part.

Choi et al [7] present a method for modelling and machining compound surfaces commonly found in die cavities and punches. The surfaces are modelled using a CSG scheme that consists of planar surface elements, general quadratic surface elements and composite parametric surfaces. Cutter paths are computed according to the Cartesian machining method, where the tool paths are planned to be parallel straight lines on the xy-plane and then projected on to the surface in question in order to determine the actual tool paths (figure 3.2). This results in cutter contact data, which must be converted into cutter location data by determining the normal vector at a given point on the surface and then by using vector analysis to determine the appropriate cutter position. This technique is based on the assumption that all machining is carried out using a ball-nosed cutter. Choi et al are in the process of carrying out further research to solve the problem of gouging that is inherent with this type of approach and early experiments indicate that possible solutions exist.
3.3.3 Cutter Path Verification

One of the problems of generating NC code either manually or automatically is that it cannot be assumed that the code is correct. The program must therefore be verified, which is both expensive and time-consuming as it occupies the time of a machine tool and a skilled operator. Several researchers have investigated the possibility of verifying the NC code by computational means and early attempts resulted in graphical displays of the tool path, which a skilled operator could examine and verify [4]. At the very least this approach did not occupy the valuable time of a machine tool, but the resulting displays are often confusing and not especially helpful. Recent research initiatives, however, have looked at the problem in more detail and have developed systems that represent the cutter path as a series of swept volumes, which can be differenced from a model of the workpiece blank and then compared with a model of the required part [4]. This approach is theoretically sound but suffers from the following problems:

- the limited domain of solid modelling systems that may not be capable of modelling all the shapes that a cutter can make [4],
- the model of the part is based on nominal dimensions, which implies that the cutter must machine to perfect size for it to be acceptable [80],
- the limited capacity of solid modelling systems that may not be able to accommodate the enormous amount of data that is inevitably generated [4], which is reported to be proportional to the fourth power of the number of tool movements [80].

Hunt and Voelker suggest an alternative incremental method that avoids the size limitation of solid modelling systems by examining each cutter path separately and subjecting it to a number of tests:
• checking that cutters cut in allowable directions and with allowable parts of their volume (i.e. parts with teeth)
• evaluating speeds, depths of cut and feeds against economic models of the cutting process
• looking at the deformations of the cutter and workpiece under cutting forces
• modelling the flow of coolant and swarf.

These processes are ordered in increasing complexity and have yet to be completely solved.

3.4 Integration in the Design to Manufacture Environment

The previous sections of this chapter have described some of the many computer-aided systems that are available within the design and manufacturing planning environments. However these are not the only areas of the business in which computers have made an impact as many manufacturing devices can now be computer-controlled including robots, CNC machine tools, Coordinate Measuring Machines Automatically Guided Vehicles (AGVs), transfer lines etc. Computer-based systems such as Materials Requirements Planning (MRP), forecasting analysis, process simulation are also used in the non-production areas of the business such as administration, purchasing, sales and marketing. Each of these systems has had a substantial impact on the performance of the activity in question and indeed has often generated unquantifiable but real benefits in other areas of the business such as improved quality or more accurate cost information. However it has also become apparent that the real company-wide benefits of these systems can only be realised if they are integrated with their related activities as it has been observed that the communication of accurate and detailed data to and from a process is just as important as the automation of the pro-
cess itself [81]. These automated but poorly integrated processes have often been likened to Islands of Automation, [81,82,83] which must be bridged by full data integration for the effects of automation to be globally realised.

3.4.1 Integration of CAD and CAM

Computer-Aided Design (CAD) and Computer-Aided Manufacture (CAM) in the form of wire-frame based draughting systems, NC part programming systems and NC machine tools were amongst the first automated systems to be introduced into industry and were subsequently the first systems that were seen to need integration in order to fully realise their potential [61,84]. NC part programming systems greatly reduce both the time taken to generate a part program and the risk of mathematical error, but their performance is improved even further if the geometrical representation created by the designer can be used directly by the part programming system, which obviates the need to redefine the geometry and consequently reduces transcription errors [84]. Similarly the transfer of the part program between the code generation system and the NC machine tool was fraught with difficulties when the only means of transfer was by tape punch and reader. This method of transfer was both time-consuming and error-prone and was eventually superceded by a direct computer link between the host and the controller, otherwise known as Direct Numerical Control (DNC). In common with the other methods of integration, DNC reduces both the time taken to transfer the program and the risk of transmission errors, but because it is now so easy to transfer information, many other data can be returned to the host computer, such as tool lengths, fixture offsets etc.

3.4.1.1 Design Concurrency in CAD/CAM

Nowadays virtually every CAD/CAM system on the market claims at least to
have integrated CAD to CAM on the uni-directional data-transfer level described above, but there are more subtle integration issues involved in design and manufacture that are now beginning to be investigated [85,86] such as design concurrency. This concept aims to further integrate the processes of design and manufacture by providing the designer with access to information regarding the producibility and usability of the object under consideration during the design process [81,85,87]. However this type of facility can only be provided in a highly-integrated manufacturing environment as the information required to support the designer must be accurate and detailed in order to be of any benefit and this quality of information is not often available in traditional manufacturing systems [85].

The designer is already equipped with tools for analysing various physical aspects of the object as described in section 3.2.3 and concurrent design is an extension of this basic approach facilitated by the provision of additional tools to analyse the suitability of the part geometry for machining, assembly, casting, inspection etc. This type of approach has a profound bearing on the cost and quality of the product because the part can be designed so that it is easy to manufacture by taking into account the manufacturing facilities and methods available within the company at that time [15]. Much of this information is currently held in the minds of the production engineers and machinists of the company and is usually liberated once the part has been designed and submitted for production. If it is found, at this stage, that the part is difficult to manufacture then it must literally be returned to the drawing board to be modified, which greatly increases the lead-time and cost of the product. Research into concurrent design is now more timely than ever as it has been observed that whereas designers may have traditionally had some manufacturing experience through either working on the shop-floor or through their apprenticeship, the skills of the present-day
designer are more specialised on a technical level and as a result the designers are often recruited from other fields [88].

3.4.2 Computer Integrated Manufacture

The benefits of integrating CAD and CAM are now well recognized and by a simple extension of the argument, it can be reasoned that the integration of the entire business should increase the company's efficiency even further. This concept of company-wide integration is known as Computer Integrated Manufacture (CIM) and is achieved through careful analysis and rationalisation of all business activities so that an efficient and effective system can be integrated [89]. This integration is made possible by advances in computing technology; specifically hardware, software, database management systems and telecommunications [53].

Planning is arguably the most critical stage [90] in the implementation of the CIM philosophy as there is no advantage to be gained from integrating a poorly designed system. For example a manufacturing system that has a quality problem before it CIM is implemented will only produce more scrap at a faster rate if the system is integrated [83]. Idef0 (ICAM definition method zero) is an established method for modelling and analysing complex processes and has consequently been used to great effect by companies [91] and CIM consultancies such as the Advanced Manufacturing Technology Centre (AMTeC) [92]. Idef0 was commissioned by the US Air Force in the early 1970s and is a methodology for modelling the functional aspects of a business in terms of activities, which use, produce and are controlled by data [93]. AMTeC use Idef0 to model existing manufacturing practice in order to produce an as is viewpoint, which can be used as a basis for rationalisation and improvement. The Idef0 model assists them to discover the organisational changes necessary to support integrated manufacturing and to derive an as should be view. The as should be view in
turn forms the basis of a CIM strategy and an understanding of the entire company's activities from which decisions on new system implementation and organisational development can be based [92].

The importance of the modelling, analysis and rationalisation of the business activity cannot be over-estimated because each company is unique [89] and requires its own customised solution to CIM implementation [81]. The strategy or architecture of CIM implementation should reflect this uniqueness and should also be capable of accommodating future changes in a step-wise fashion as integration is necessarily an evolutionary process [81].

A strategy for CIM inevitably involves the storage and transfer of vast amounts of product and company data, which need to be carefully managed if the system is to realise its full potential. Typically this is achieved through the implementation of distributed computers, a communication system and a common data or knowledge base [94].

At the management level of manufacturing, a main-frame may operate in batch mode for such functions as pay-roll, personnel records and market forecasting. In the various design and manufacturing departments, powerful engineering graphics workstations are used, and on the shop-floor there are multi-tasking and real-time process control computers [94].

Communication systems generally include local area networking concepts such as baseband for small operations (Ethernet, Decnet etc.) and broadband communication networks for large computer systems [94]. The Manufacturing Automation Protocol (MAP) is also beginning to make an impact and it is expected that this initiative will solve many of the low-level integration problems [82,95].
The most basic requirement of a CIM strategy is an effective and well-designed engineering data-base [81], but it has been observed that current data-base technology cannot easily accommodate the complexity of product and company data structures required by CIM systems [94]. The efforts of the research into Product Modelling Environments described in section 2.4.3 are expected to satisfy this requirement.

The effects of CIM on a company are centred on the greater efficiency of a thoroughly planned and integrated business environment that manifests itself in:

- higher quality levels [83,93]
- shorter design to manufacture lead-times [89]
- reduced resistance to market changes [89]
- reduced inventory levels [89]
- increased asset utilisation [91]
- more accurate and detailed management information [53,81,93]

The problems involved in implementing CIM are:

- rationalising and organising the business so that it can fully benefit from the effects of CIM [54,89,92]
- integrating computer systems from different vendors, which were originally designed as stand-alone systems with no consideration given to high level integration with other systems [54,81,90]
- software transportability [54]
- ensuring data consistency throughout the business [81,90]
- controlling data so that it can only be accessed by authorised systems and personnel [81]
• changing existing and traditional working practices [94]
• accommodating external influences including political movements, legal enactments, social factors, ethics etc. [94]
Chapter 4

Inspection Plan and Code Generation

4.1 Introduction

Inspection Plan and Code Generation is a relatively new field of manufacturing research that is concerned with the automatic generation of an inspection plan and control code for a measuring device by reference to component geometry and constraints. In many respects it is broadly similar to the parallel activity of process planning and code generation for machining, as many of the steps involved in the two processes are equivalent. In both cases the following information must be determined depending on the type of machine under consideration:

- operations
- component datums for machine reference
- component setups
- fixturing methods
- tool/sensor configurations
- tool/sensor orientations
- safe rapid paths
- traverse speeds

However, in contrast to the machining process, relatively little research effort has been directed towards Inspection Plan and Code Generation, until recently when it has been observed that the number of research programmes concentrating in this area has increased markedly. This has possibly been brought about by a combination of factors related to the market forces operating on industry and the amount of progress that has
been made in the parallel research area of process planning and code generation for machining. Consumer-led pressure for higher levels of quality and greater product variability have forced the manufacturer to increase the level and frequency of inspection and to reduce batch sizes, which both serve to increase the load on the inspection planning activity. Increasingly more automated planning systems for machining have also had the effect of reducing the lead-time for machining but as yet without a corresponding reduction in the lead-time for inspection. There are indeed many similarities between the problems that must be solved in machining and inspection, but as yet very little effort has been devoted to transferring this knowledge between the disciplines. As a result of this, the solutions that have appeared so far vary quite substantially in both the scope of inspection planning covered and the techniques used to solve the problem.

In order to gain a complete impression, this chapter will examine and critically review the published literature of each programme of research in this field in ascending order of the date of publication. In the several cases where there are more than one publication, the earliest publication date is used.

Section 4.15 concludes the chapter by briefly reviewing a representative selection of commercially available inspection planning systems.

4.2 Kawabe, Kimura and Sata, 1980 [96]

System Description. The Department of Precision Machinery Engineering at Tokyo University have long been involved in the field of product modelling (refer to section 2.4.3.1) but their research into the automatic generation of NC commands for Coordinate Measuring Machines (CMM) pre-dates this by several years. Their experimental system automatically obtains a geometrical representation of the component from the GEOMAP CSG-based solid modeller, which was also developed by the same
Chapter 4 - Inspection Plan and Code Generation

department. The constraints to be measured are specified by the user and are limited to linear, angular and radial dimensions of planar and cylindrical surfaces. The user selects the type of constraint to be measured and then nominates the constrained surface/s using the cross-hair cursor of the display device.

The next stage of the process is the generation of the probe path and this is initially achieved by manually ascertaining the location and orientation of the component on the CMM table so that the data tables of the planning system can be transformed accordingly. Probing points for each surface are retrieved from surface-specific algorithms and paths are generated between each of the points (figure 4.1a). The points are verified for collision avoidance by constructing a geometrical model of the volume the probe would occupy and then by checking this for intersection against the model of the component (figure 4.1b). If a probe path is found to collide with the component (figure 4.1c) then the probe is instead retracted to the safe space above the component (figure 4.1d), where no collision is possible. A part program can then be generated for the CMM, which in the case of these experiments was an early three-axis Mitutoyo machine with no probe re-orientation capability.

System Critique. This is an early and far-reaching piece of research that supercedes many later experimental inspection planners by the very fact that it uses a solid model representation, which allows it to check for probe/component collisions, albeit on a rather approximate basis. However it has no geometrical tolerancing measurement capability and the dimensions and tolerances are not integrated with the solid modeller. It also uses manual input to determine datums and axis systems and does not incorporate a component or probe setup capability.
4.3 Hopp and Lau, 1983 [97,98]

System Description. A hierarchical task-decomposition control system for inspection is proposed by Hopp and Lau, which is discussed with reference to a CMM, but can be configured to control any measurement device. The system consists of a hierarchy of control levels, each of which is responsible for transforming an input goal into a set of simpler goals that are issued to lower levels of control.

The principle function of the control system for a CMM is to decompose the goal of Inspect Part down to the level of individual servo commands and this is carried out in terms of the following seven hierarchical levels of control:

1. **Tolerances.** This control level is equivalent to the Inspect Part control level, and therefore sub-divides its goal into tolerances that should be inspected in order to satisfy the inspection requirements. It also receives reports from the lower levels regarding how well a tolerance has been satisfied according to the measurement on the CMM.

2. **Features.** This control level receives inspect tolerance goals from above and decomposes them into the features that must be inspected for the tolerance to be satisfied. If the datum reference frame for a tolerance has not been determined this level defines sub-goals to measure each datum feature. Once the feature has been inspected by lower control levels and the result propagated up the hierarchy then the feature control level can be configured to determine whether the result is satisfactory and whether the feature should be re-measured taking more points.

3. **Surfaces.** The features determined by the previous level may not always equate to surfaces on the part, as in the case of centre-lines, edges or projected entities. This control level decomposes each feature into a surface that can be measured.
4. **Probing Points.** The purpose of this control level is to determine the points on a surface provided by the previous control level and to fit them to a mathematical representation of the feature so that the surface can be measured.

5. **Probing Paths.** This control level is responsible for probing a point on the component without causing collision in the process. The collision-free path is a sequence of straight-line motions to be carried out by lower levels and can be generated by several methods. An explicit path can be prescribed by the operator, but this has the disadvantage that it limits the flexibility of the system to alter the inspection sequence based on higher level considerations. Another method is to store a set of *safe motion volumes* for the component through which the probe can travel safely, and the final method is to use the part geometry to determine safe paths. The authors admit that path-finding is still an open problem, but suggest a method based on a *generate-and-test* algorithm. The first stage is to check if the straight-line describing the path intersects with the object, and if it does, a point is added to the path just off the edge of the surface so that the line can now be considered as two lines and consequently two simpler problems. The process is repeated until a safe path is found (figure 4.2). The method is acknowledged to be currently restricted to the two-dimensional situation and potentially expensive with respect to *cpu* time.

6. **Machine Motion.** This control level converts the coordinates of the probing path into machine coordinates that are appropriate to the CMM. This is generally a simple activity but can become more complicated if the CMM has more than three axes.

7. **Servo Commands.** This final control level coordinates the actions of the various servo motors on the machine so that the required machine motions are carried
System Critique. This hierarchical approach provides a rigorously defined and powerful tool that appears to be capable of generating an optimum CMM part program by constantly reviewing inspection methods during the planning process through interaction with the user and simulation of the CMM's activities. It currently does not provide an automatic component setup facility and appears to use a single probe configuration for the entire inspection cycle. Dimensions and tolerances are not integrated with the geometrical representation and are selected in the same manner as the previous system of Kawabe, Kimura and Sata (section 4.2). No case-studies are described, but the system is certainly sufficiently detailed to generate an inspection plan although the degree of automation with which that is carried out is not clear. It is implied that the complete system is automatic once the component has been set up in an appropriate location and orientation on the CMM but there is not an adequate explanation of how the number of points, the point distribution and the type of mathematical entity for fitting the surface are selected. The system was, nevertheless, the first inspection planner to rigorously define the steps involved in this process and remains to this day the only system that is capable of modifying its inspection methods during the inspection planning process. It is also sufficiently powerful to be able to control the CMM directly, thus by-passing the CMM's on-board controller, and avoiding the discrepancies that have been observed between the measurements obtained from CMMs of different manufacturers [24,25].

4.4 Duffie, Bollinger, Piper and Kroneberg, 1984 [99]

System Description. A method is presented for CAD-directed inspection and error analysis of sculptured surfaces, defined using bicubic parametric surface patches. The bicubic parametric surface patch is the lowest order of parametric surface
representation that can conveniently be used to describe non-planar shapes. A smaller number of patches may be needed with patches of a higher order than three, but these require the definition of many more controlling parameters, which subsequently increases the difficulty of surface manipulation. In addition, unwanted oscillations in surface shape can occur more easily when higher order polynomials are used.

The inspection process is initiated by the user selecting the patch or group of patches that require measurement so that the inspection planner can automatically determine the coordinates of the points to be measured. This is done by spreading the points uniformly over each patch in a grid-like formation dependent on the size and proportions of the patch (figure 4.3a). For each surface point specified by the system another two points are determined which lie on the surface normal and are equidistant from the surface; one behind and one in front (figure 4.3b). During the measurement cycle the probe traverses rapidly to the point in front of the surface and then traverses slowly to the point behind so that it touches the surface somewhere between the two points. The coordinates of the measured point are returned by the CMM and are corrected by the inspection planner in order to determine the actual surface point, which is necessary because the CMM only returns the centre-point of the undeflected stylus ball. These coordinates can be corrected because the radius of the stylus ball is constant and the approximate deflection along the normal can be derived empirically. However this assumes that the calculated surface normal was correct, which is an unfortunate Catch 22 situation as that is one of the surface characteristics that is effectively under examination. The authors discuss the effects of surface normal errors on the measurement and conclude that they can be reduced in magnitude by using the smallest possible stylus ball and also that it might be possible to estimate the actual surface normal by taking more than one measurement.
System Critique. This is an early and thorough attempt to solve the problem of automatic sculptured surface inspection and addresses all of the pertinent issues with alacrity. The inspection of sculptured surfaces represents quite a different problem to the more conventional planar, cylindrical, spherical and conic surfaces as the CMM cannot be programmed to fit a sculptured surface to a pre-defined mathematical entity for further analysis. Instead the CMM can only return uncompensated points, which the inspection planner must interpret and correct in order to determine a surface point that can be related to the expected coordinates. This approach proposes a simple but effective method for compensating the measured values and acknowledges the problems of surface normal errors. However the implied use of a constant value for probe deflection may not be valid as the magnitude of deflection varies according to the attack angle [100]. This inspection planner does not determine probe setups, component setups or collision-free safe rapid paths but this is not a major deficiency with this type of application as many components with sculptured surfaces do not require more than either one setup or one probe and do not pose the sort of collision problems that would trouble this simple approach. Typical examples of such parts include dies for automotive body panels and telephone hand-sets.

4.5 Van den Berg, 1987 [101]

System Description. Van den Berg describes a research programme that has been investigating closed loop inspection in a small manufacturing cell incorporating a five-axis machining centre, a six-axis robot, a CMM and a cell controller. The purpose of the cell is to manufacture small batches of precision parts, which can be represented by sculptured surfaces, and the author cites the manufacture of turbine blades as an example. The objective of the research is to develop a fully automatic method for manufacturing the precision parts, which incorporates the following characteristics:
• automatic machining

• automatic material transfer

• automatic inspection

• automatic error analysis (estimate the cause of error and determine a correction strategy)

The inspection and error analysis cycles should also take less time than the machining in order to maximise the productivity of the cell.

Inspection of the part is carried out using both the CMM and the machining centre, which is equipped with a probe, although at the time of writing the inspection software for the machining centre had not been implemented. A menu-driven system allows the user to interactively develop inspection programs and as with the previous system of Duffie et al (section 4.4) only points can be measured as the component types can only be represented by sculptured surfaces. It is not made clear whether there is any intelligent assistance in the determination of probing points, but the user is apparently able to specify a hopping motion to facilitate the measurement of a number of points on the surface.

Two methods of error analysis are employed within the system; the first is tolerance analysis, which determines whether the measured point is out of tolerance and the second is manufacturing analysis, which determines the cause of the error and a correction strategy. The tolerance analysis facility is capable of checking for position, orientation and form errors by calculating the error vector of each point with respect to the surface and then by determining the transformation matrix necessary to fit the measured points to the defined surface. The position and orientation errors can be extracted from the net rigid body fitting transformation, while the form errors can be extracted from the error vectors.
The manufacturing analysis is a two step process. Firstly the observed errors must be matched against the possible sources of error in the manufacturing process model, which creates a model of how the errors observed in the measurements were caused. The second step involves the application of corrective strategies to the manufacturing process.

System Critique. The inspection planning aspect of this research is minimal as it is based on the CMM programming language software, which relies heavily on user-interaction. However this is the earliest example of a fully-integrated machining and inspection cell, albeit only in a partially implemented state at the time of writing. This research shows that it is possible to implement a fully automatic manufacturing cell for a limited domain of parts, and initial results indicate successful analysis of form errors, but with more research required for position and orientation errors.

4.6 ElMaraghy, 1987 [102,103,104]

System Description. ElMaraghy and his colleagues from McMaster University in Canada have been researching a variety of subjects in the field of automated manufacturing, but have recently paid particular attention to the problems faced by automated inspection. The most relevant research to the work of this thesis is a generative feature-based inspection planning system for CMMs [102] that uses an expert system to determine feature accessibility, surface representations and priority of datum features.

The basis of operation of the planner is the automatic classification of the component to facilitate feature extraction so that an expert system can manipulate the inspection methods attached to the features and determine how the part should be inspected. The part is geometrically defined using conventional modelling tools, and dimensions and geometrical tolerances are allocated to the component model on an
interactive basis with assistance from an Expert Tolerancing Consultant for determining optimum geometrical tolerances. The features are then extracted from the component using the method of Syntactic Pattern Recognition, which finds those features from the system feature base that most closely match the elements of the component. This is done by determining the minimum Levenschtein distance between the syntactic pattern classification of both the component elements and the features of the system feature base, where the Levenschtein distance is defined as the minimum number of transformations required to make the two syntactic patterns identical. Once the features have been extracted from the component and the part family has been determined, the inspection methods associated with the features can be inherited by the component.

The first activity of the inspection planning process is to determine those features of the part that cannot be inspected using a CMM, by reference to both the inspection methods associated with each feature and a rule-base. An example is given whereby features which correspond to gear-type parts or features that are constrained by the run-out geometrical tolerance are designated as non-CMM features.

Component setup is carried out by the operator, but once the part is setup on the CMM table, the inspection planner determines the accessibility of each feature in that particular component orientation. This is again achieved using a set of rules, which analyse each feature to determine whether they will be obscured by any other part of the component.

The problem of determining the inspection sequence of the features that are accessible in a particular component orientation is interrelated to the problem of selecting probes, the solution of which is based on the geometrical properties of the feature and the corresponding tolerances. The inspection tasks are sequenced by
selecting the important features first such as the primary datums and then by selecting all the other features that depend on them. The final activity of the inspection planner is to collate all the surface measurements in order to generate the actual values of the dimensions and geometrical tolerances attached to that part.

System Critique. This inspection planner is a truly generative solution that uses an expert system to determine the measurement process, feature accessibility, probe types and the inspection sequence for conventional coordinate measurement tasks. This planner does not consider component orientation determination or fixture design, but it can be assumed from the structure of the approach that these activities could be incorporated into the planning rationale. However it is a feature-based system and the inspection planning rules are heavily dependent on these features, which must make the addition of new features cumbersome and the analysis of geometry outside the domain of features impossible. All of the case-studies examined by this approach are of a rotational nature but the system is claimed to be equally effective for prismatic parts. However, without detailed evidence this is doubtful, as the rotational case is essentially a two-dimensional one and represents a substantially less complex environment for solving, for example, the feature accessibility problem. In a rotational environment accessibility can be determined without excessive difficulty because the majority of inspection tasks are performed with the probe parallel to the centre-line axis, which consequently reduces the problem to checking the order of the features along the axis and their maximum radii. With prismatic parts, it is difficult to imagine how a similar approach could be adopted with the same degree of effectiveness.

ElMaraghy's subsequent research in this area includes a highly automated vision-based surface defect inspector for engine valves [103] and a formulation for evaluating composite geometrical tolerance values for cylindrical features from measured coordi-
4.7 Chang, Anderson and Mitchell, 1988 [75,105,106]

System Description. The work of Chang, Anderson and Mitchell is concentrated on the development of an integrated manufacturing cell in a similar vein to the research of Van den Berg (section 4.5) but with further reaching objectives. The manufacturing cell is known as the Quick Turnaround Cell and incorporates four tightly coupled modules: a feature-based design system, an automatic process planning and part programming system, a cell controller and a vision monitoring and inspection system [75,105].

One of the primary objectives of the system is that user interaction should only be necessary during the design stage of the manufacturing process and even this is intended to be as user-friendly as possible. The design module allows the designer to construct a component from features, which are loosely related to machining operations, such as holes, slots or countersinks. In order to provide enough information for down-line activities such as process planning and inspection, dimensions and tolerances are associated with the size parameters of each feature, such as the depth, radius or relative position. Once the component has been built from features it can be transferred to the TWIN boundary representation environment for graphical display and verification, and this secondary representation can also be used during subsequent stages of the manufacturing process. The process planner is known as AMPS (Automatic Machining Planning System) and consists of process selection, tool selection, process sequencing, fixturing method planning and NC cutter path generation. The cell controller controls and coordinates the manufacturing cell and provides an interactive interface, which guides the operator through tool and part loading/unloading.
Chapter 4 - Inspection Plan and Code Generation

The inspection module of the Quick Turnaround Cell is vision-based and has specifically been designed to cover a wider scope than most inspection planners by incorporating tool inspection, fixture inspection and manufacturing process analysis as well as the conventional final inspection. Only the tool inspection and final inspection activities have been implemented at this stage and they both utilise CCD video cameras and VME board level image processing components. The tool inspection system uses a camera mounted above the tool magazine and analyses the image with algorithms using grey-level morphology to adaptively segment the tool from the background under a variety of lighting conditions. This approach is not sufficiently accurate to measure tool wear, but is perfectly capable for checking tool identity and tool breakage.

The finished part inspection system [106] operates in the following three stages:

- part recognition,
- position estimation,
- inspection planning.

One element of information that is common to all three stages is the view directions of each feature, which are determined during the up-stream process planning activity that also determines approach directions and feed directions for machining. These directions are all determined using techniques similar to those used for feature recognition [75], and enable the planner to determine how the feature interacts with the surrounding geometry. For example in figure 4.4 the designer has intersected a block with a slot, and depending on the parameters of the two features, three quite different surface topologies can be generated.

The view directions are used by the part recognition activity in order to synthesize a strategy tree for matching a two-dimensional image against a three-
dimensional geometric model in the following manner:

1. select focus features
2. generate hypotheses from consistent matches of focus features
3. verify hypothesis through edge comparison
4. refine the parameters of the current hypothesis

Position estimation is inevitably determined during the part recognition activity and the result of this is a set of rotational and translational transforms.

The inspection planning activity is understandably more complex than the part recognition and position estimation activities and is carried out in the following six stages:

1. A list of all the dimensions and tolerances that require measurement on the part is compiled from the parameters of the feature representation of the component and rationalised to eliminate any tolerances that might be redundant through feature interaction (figure 4.4).

2. The required component orientations are determined by compiling lists of tolerance measurements that can be carried out for each possible component orientation with respect to view directions. The component orientation sequence is then determined by counting the number of tight tolerances for each orientation using the following criterion:

\[
A_k = \sum_{i \in F_k} \frac{1}{\text{tolerance}_j}
\]

where,

\( F_k \) is the index of features in orientation \( k \)
Chapter 4 - Inspection Plan and Code Generation

$T_i$ is the index of tolerances in feature $i$

$\alpha_i$ is the weighting factor for each design feature

$\beta_j$ is the weighting factor for each tolerance

The component orientations can then be sequenced so that the orientation with the greatest $A_k$ is carried out first and so on until all tolerance measurements have been allocated.

3. Datum reference frames are established for each component orientation.

4. A rule-based system decides, which surfaces need to be measured in order to satisfy every dimension and tolerance of every feature. In most cases this is straight-forward, but occasionally dimensions will constrain constructions of more than one surface, for example edges or vertices.

5. The CAD data-base is searched to select the edges from the Boundary Representation model, which will be used to measure each surface. This is facilitated by the fact that the symbolic information associated with each feature is appended to every internal data structure of the b-rep model, which allows the system to select the b-rep edge of each feature that has a valid view direction. After finding the approximate orientation and location of the part, the system runs precision measurement algorithms over small windows, where these edges are expected to appear.

6. Constraint algorithms are chosen to calculate the actual value of each dimension and tolerance depending on their type. For example, a constraint algorithm for measuring a linear dimension between two parallel planar surfaces will not need to compute all distances between the two edges, whereas an algorithm for measuring straightness will be required to analyse every point on the edge.
System Critique. The Quick Turnaround Cell research programme represents a thorough analysis of the issues involved in implementing a highly automated design to manufacture environment and is also the first attempt to establish an inspection planning facility for a highly variable part range using vision techniques. Many successful automatic inspection systems have been reported using vision techniques [107,108,109,110,111,112,113,114], but the majority of these installations are designed for geometrically restricted part ranges. This limitation is caused by the problems of part identification and the variability of illumination requirements. This research has overcome these difficulties through the design of a system that inspects tools, fixtures, cutting conditions as well as finished parts. The finished part inspection activity determines measurement and constraint calculation algorithms, component orientations and sequences, and datum reference frames for each component orientation. Unlike inspection planning for CMMs, the vision-based approach does not need to compute collision-free probe paths or the number of points necessary to accurately represent a surface. However, in comparison with CMM inspection planners, vision-based systems suffer from the problems of inspecting a three-dimensional object from a two-dimensional image, which can only be carried out accurately when the object is orthogonal to the viewing direction.

The inspection planner is substantially aided by the dual feature/b-rep modelling environment but appears to be unnecessarily restricted by the assignment of dimensions and tolerances to the feature parameters. This either limits the designer to a dimensioning scheme that is essentially enforced by the feature, or to a multitude of geometrically identical features with different dimensioning schemes. The latter situation is untenable as it will inevitably result in an unmanageable quantity of features and the former would severely limit the designer's ability to influence the function of
the component [115].

4.8 Galm and Merat, 1988 [116]

System Description. The inspection planning system of Galm and Merat is revolutionary inasmuch as it can plan for the part to be inspected by a combination of fast low-accuracy devices such as range imagers and higher-accuracy but relatively slow CMMs.

The first stage of the planning process is to build what is termed a reference model from the CAD description of the object transmitted via IGES (refer to section 2.4.1). As the version of IGES used for this research could not support a solid model representation, a level of interpretation was required in order to construct the internal representation, which was based on a surface adjacency graph. Geometrical Dimensioning and Tolerancing (GDT) is applied according to ANSI 14.5M [22] and its rules are embedded within the planning system using the predicate calculus facility of PROLOG.

A complementary object model is constructed in the first instance from the measurements obtained from a fast low-accuracy sensor, which has sufficient information density to permit recognition of object features in order to estimate part orientation. The object model is also represented as a surface adjacency graph, which is built by converting sharp discontinuities in point depth values into vertices, smoothly connected points into edges and regions of points with similar characteristics such as normals of curvatures into surfaces. The object model is be compared against the reference model in order to determine the object orientation.

The measurement of the object according to its GDT specification is carried out by analysing each feature of the surface adjacency graph. If the tolerance associated
with a constraint is found to be tighter than the measurement accuracy of the inspection device then the feature is added to a refinement list for re-measurement by a higher-accuracy device.

System Critique. The authors admit that this inspection planning system is only partially implemented, but the planning rationale described is certainly indicative of the types of systems that might be expected in the future. This is because the planning system is able to exploit the strengths of vision (speed of operation and the ease with which it can determine certain physical characteristics such as component orientation) and the strengths of coordinate measurement, which include greater levels of accuracy and a true 3d measurement capability. These two dimensional inspection methods complement each other in many areas and it is expected that combined inspection devices will begin to appear on the market in greater numbers as their advantages become better realised [117,118].

This inspection planning system concentrates on the vision measurement tasks and as yet there is very little detail regarding the planning of CMM tasks except for an indication that it should be possible to minimize probe paths.

4.9 Alisto, 1989 [119]

System Description. Alisto describes a set of experiments based on an inspection machine planner for a 3D vision system, where the designer selects high-level device-independent inspection functions for each surface of the part that requires measurement. The resultant measurements are then compared with the original CAD data so that accept/reject decisions can be made by the inspection expert.

System Critique. This is a simple but effective approach that does not advance the level of automation in vision-based inspection planning, but does introduce the
concept of design/inspection concurrency.

4.10 Traband and Medeiros, 1989 [31]

System Description. An approach is described, which allows the designer to specify tolerances within a CAD system so that they can be automatically extracted by an inspection planning system that generates output for a vision system.

The tolerance specification system has been built on to the AutoCAD two-dimensional CAD system, so that when the designer is ready to allocate the tolerances, menus can be called up that are interfaced with AutoCAD's internal tolerance specification system that operates using simple textual annotation. At the same time, when tolerances are selected from the menus, an ASCII text file is constructed that describes the type of tolerance that has been assigned, its parameters and the geometric entities that are constrained.

The inspection planner operates in the first instance by sequencing all of the tolerances that require measurement, so that datums are measured first, then geometric tolerances, followed by conventional size tolerances. At each stage tolerances that are constructed from more than one geometrical entity are sequenced to be carried out after the simple tolerances so that duplication of measurements can be avoided. The next stage is to plan the operations required to calculate the relationships between the geometric entities measured by the previous stage so that the actual values of the dimensions and geometric tolerances can be determined and compared with the specified values. Finally the part program is generated and transmitted to the inspection device for measurement to take place.

System Critique. The tolerance specification system is a basic solution to the problem of tolerance representation that does at least allow an automatic application to
retrieve tolerance information. Tolerance retrieval is not practical in the standard
CAD environment because the information is represented as leader lines, arrowed
lines and textual notes, and cannot easily be differentiated from the elements of the
CAD model. However the storage of tolerance data in an ASCII data file as proposed
by this approach, is not conducive to tolerance retrieval or analysis and it is not
apparent how the constrained geometric entities can be adequately represented in this
format.

This is essentially a two-dimensional solution to inspection planning for vision
systems, limited by the dimensionality of the AutoCAD wire-frame representation and
the Vidicom Qualifier vision system. It does not consider component orientation
determination, but it does select the measurement and relationship algorithms
automatically and determines a sequence that avoids measurement duplication. As
with the work of Ailisto (section 4.9), this research does not make any advance on the
efforts of Chang, Anderson and Mitchell (section 4.7) but it does introduce the advan-
tages of planning the object measurement operations in such a way that measurement
duplication can be avoided. Future research is likely to consider two-dimensional
inspection planning for CMMs with emphasis on minimizing the probe path through
analysis of the operation sequence and also the possibility of rejecting a part as soon as
a feature is measured out of tolerance.

4.11 Atkins and Derby, 1989 [39]

System Description. Atkins and Derby have been researching a partially-
automated off-line inspection planning tool, which is integrated with the US Air Force
sponsored Product Definition Data Interface and its extension the Geometric Model-
lng Applications Interface Program. In fact the raison d'être of their research is to
demonstrate the feasibility of integrating inspection systems into the CIM
environment using Product Definition Data.

The first step of the inspection planning process is to retrieve the part geometry and tolerance information from the Product Definition Data and to sequence the tolerances and the features that they constrain so that tolerances that depend on the construction of several features are carried out after those features have been measured.

The definition of the probe path can then be achieved by either one of two methods. If the feature to be measured is sufficiently simple like a hole, then a parametrically controlled macro can be associated with the feature and called up by the operator when the probe path is being determined. Otherwise, in the case of a feature such as a plane, the operator is requested to specify the probe path manually, although this process is assisted by the automatic provision of additional coordinate systems relative to neighbouring features so that the operator is not restricted to moving the probe with respect to the master coordinate system.

Once the probe path has been completely defined, the operator is able to animate the motions of the probe so that a visual verification can be carried out to ensure that the probe path is collision-free. The animation can be halted at any point to eliminate unnecessary moves or to adjust clearances.

The final step of this application is to generate a DMIS output file, which is based on the Dimensional Measuring Interface Specification of the CAM-I organisation [120] and can be post-processed into the appropriate CMM programming language.

System Critique. This system does not attempt to fully automate the inspection planning task, but does strive to exploit the advantages of Product Definition Data and the Dimensional Measuring Interface Specification, which are both beginning to create a significant impact on inspection planning research. The use of these neutral data formats to represent both the data required and generated by inspection planning,
facilitate the integration of this activity with the CIM environment, which has already been established as an essential objective of automated manufacturing applications (refer to section 3.4.2).

Product Definition Data allows the inspection planning application complete access to a geometrical representation of the product and the tolerances and datum reference frames associated with it. It also facilitates the analysis of this data through the hierarchical organisation imposed on the data by the Product Definition Data Interface. An example of this is provided by the analysis of the tolerances which are dependent on other features during the tolerance sequencing phase of the inspection planning process.

The inspection planning system itself is no more than a computerised aid to the operator, although it is clear that further research could automate the other elements of the inspection planning process. Currently, the system retrieves and sequences the tolerances and generates the probe paths for certain features. The operator is expected to determine the component orientations, fixturing requirements, probe configurations and also to ensure that the probe path is collision-free assisted by an animated sequence of the probe motions.

The researchers admit that user-interaction is hindered by the use of a wireframe geometric representation, which offers only a marginal improvement to product visualisation over conventional methods. They suggest that a solid model representation will alleviate this problem with its hidden-line removal capability and that this type of representation will permit further automation of the inspection planning activity.

4.12 Mullineux et al, 1989 [121,122]

System Description. The main purpose of the research at Brunel is to investi-
gate the benefits of installing a bi-directional intelligent link between design and manufacture, and in particular between a CAD system and a CMM. The approach adopted by the team is based on the concept of a blackboard, which is an idea that has arisen from work into Artificial Intelligence and is used to refer to a central data structure capable of communicating with a number of other systems. The blackboard data structure supports six data types, which include integers, real numbers, angles, strings, pointers and constraints. The constraint type is a combination of a string and a real number and is the means by which the blackboard represents truth, which occurs when the real number evaluates to zero. The blackboard also consists of a number of other tools (figure 4.5), the most powerful of which being the obey/resolve module. The obey part of the function allows command strings, and in particular calculations, to be interpreted and performed. The significance of the resolve part is that it can deal with constraints, by attempting to make all of the constraints submitted to the function true before completing the evaluation. Through this means, the blackboard has its own problem solving mechanism.

The inspection planner is designed to show how each step of the process can be dealt with, albeit in a sometimes simplistic manner. However the key-point is that the blackboard has the capability of undertaking these tasks, which are under a continuous schedule of improvement and expansion. The first stage in the process is for the user to select the features to be measured through the CAD system and then to nominate a local origin to be used during the inspection process. The user then supplies approach vectors and probing data for each feature, for example the probing depth in a hole, which are then passed to the blackboard and stored in a feature file. The blackboard can then generate probing paths in the form of vectors and coordinates and these are sent to the CAD system graphics screen for verification by the operator.
Chapter 4 - Inspection Plan and Code Generation

The inspection code is then generated and transmitted to the CMM along with instructions for setting up the machine for the operator, and once the part has been inspected, the results are returned to the blackboard for comparison with the CAD data. The example tested by the system was a die block with a number of holes in the top face and the results from this experiment were compared with the CAD data by using rules to align the two sets of holes so that each measured hole fell within the tolerances prescribed by the CAD data. If the holes could not be aligned, further tests could be carried out to determine the cause of the error, starting with an analysis of the method of manufacture to see how this could have caused the error.

System Critique. The Brunel inspection planner has clearly achieved its initial objective of investigating the use of the blackboard concept to automate the CAD to CMM link and has in the process researched a particularly interesting method for analysing the CMM output. The ability to manipulate the CMM results so that they fit the CAD data will allow the inspector to fully utilize the available tolerance bands and will provide a more accurate analysis of whether the part will satisfy its function. The error analysis activity appears to be in the early stages of investigation but also promises to provide a revolutionary solution to this notoriously complicated field.

The inspection planner itself is only partially automated and relies on the user to supply a substantial amount of planning information. However, it does generate unchecked probe paths and the inspection code and certainly forms the basis of a more automated approach.

4.13 Tang and Davies, 1990 [123]

System Description. The research at UMIST is concentrated on a knowledge-based expert process planning system for turned parts, which is known as EXCAP. This system is capable of generating process plans automatically from its own rule-base
Chapter 4 - Inspection Plan and Code Generation

and from geometry supplied by a CAD system and transmitted via IGES. The NC code is then generated automatically by submitting the process plan to TECHTURN, which was also developed at UMIST. The objective of their research into inspection planning is to develop an expert system for planning both in-process and post-process inspection. The planning requirements of both these types of inspection are broadly similar, but in-process inspection is by definition carried out on the machine tool and therefore does not have access to the range of software tools commonly available on CMMs.

The research team have identified that an inspection planner should be capable of determining the following functions, which include:

- when the inspection operation should be carried out during the machining of a component,
- whether dimensions should be inspected on the machine or the CMM,
- how the measurements should be carried out on the individual features of the component,
- how the measurement steps can be sequenced for the inspection operation
- how the component should be held on a CMM for the inspection operation.

However, at this stage in the research programme they have been concentrating on the types of knowledge representations that are most appropriate to inspection planning, which include:

- if-then or production rules, which define simple logical relationships between concepts in the problem domain,
- decision nets, which sequence the evaluation of the rules in an efficient manner,
- the problem reduction approach, which structures a problem so that it consists of
Chapter 4 - Inspection Plan and Code Generation

a number of simpler sub-problems and a selection of operators for solving them, (similar to the hierarchical task-decomposition approach of Hopp and Lau described in section 4.3)

- the AND/OR graph, which allows the relationship between sub-problems to be described more accurately.
- a combination of depth-first and breadth-first searching with backward chaining to provide a more efficient method of searching the decision nets.

A simple example is described, which involves these techniques in the inspection planning domain for solving the problem of how the component should be held on a CMM for the inspection operation.

System Critique. This research is clearly in the early stages of its development, but promises to provide a powerful environment for solving the inspection planning problem. In contrast to the machining area, very little research has been directed at fixturing strategies in inspection planning and although this considers the less complex turning problem it nevertheless represents a valid attempt to establish a rule-based strategy in this poorly researched area. The inspection planner will also benefit significantly from the integration with the process planning system, which will be able to provide information regarding the manufacture of the part and the reasoning involved in selecting the inspection device and measurements.

4.14 Spyridi and Requicha, 1990 [115,124]

System Description. Requicha's research team at the University of Southern California is developing an inspection planning system, which uses a solid model representation of the component together with a set of surface features and their associated tolerances. They propose that a complete inspection planner would need to
incorporate the following tasks:

1. selection of workpiece orientations,
2. selection and placement of fixtures and clamping devices,
3. machine selection,
4. probe selection,
5. sample-point generation,
6. generation of probe trajectories (paths),
7. generation of servo commands for the CMM controller.

The work that they have carried out so far has concentrated on solving the accessibility problems that are concerned with items one and four listed above.

The accessibility of a probe to a surface feature is determined by considering a simple abstraction of the probe as a half-line or ray with an end-point at the probe's tip and extending to infinity along the probe's axis (figure 4.6a). It is reasoned that a probe can inspect a surface if the half-line abstraction of the probe can be placed at every point on the surface without interfering with the workpiece. This analysis is carried out in two stages. The first stage is termed accessibility analysis and involves the construction of a direction cone for each feature, which consists of a collection of all the accessible probe abstractions for that feature as shown in figure 4.6b. The next stage is termed clustering, which selects a minimal set of directions sufficient to inspect all features.

The accessibility analysis module operates by considering the local and global accessibility of each feature, where the local accessibility is concerned with obstacles in the immediate neighbourhood of a point and the global accessibility takes the entire workpiece into account.
Chapter 4 - Inspection Plan and Code Generation

Local Accessibility Cones (LACs) are created by considering an infinitesimal region $\Delta F$ of a feature $F$, and constructing all the probe abstractions that can inspect that region, which is effectively all the half-lines that make an angle of $\pi/2$ or less with the region normal, $\mathbf{n}$. This LAC for the region $\Delta F$ is therefore equivalent to a planar half-space. For a probe to inspect an entire feature $F$ it must inspect all of the feature's elemental regions $\Delta F$ and the result of this is the intersection of all the planar half-spaces for each region $\Delta F$. Algorithms are being developed to compute LACs for the geometric domain of solids having planar and natural quadric faces (i.e. cylindrical, conical or spherical), but exact algorithms for more general domains are not known.

Local accessibility does not suffice to guarantee that a surface feature can be reached by a probe and so Global Accessibility Cones (GACs) must be computed. GACs are a sub-set of LACs and consist of all the half-line probe abstractions that can access a feature without interfering with the workpiece. The computation of GACs is non-trivial and is achieved by an algorithm that determines a set of half-lines that do not intersect the global obstacles of a feature, which have previously been determined by computing the Minkowski sum of the $\text{LAC}(F)$ and the feature $F$. Finding the global obstacles of a feature significantly reduces the computation, as the algorithm would otherwise be required to determine intersections of the LAC with every surface of the workpiece.

Once GACs have been computed for every feature, it is necessary to determine a minimal set of probes that will suffice to inspect the features and this is achieved through clustering. It is shown that two features can be inspected by the same probe if, and only if, their corresponding GACs have a non-empty intersection (that is to say their GACs intersect). The features can then be clustered into sets with non-empty
intersections, and the GACs in each cluster can be intersected to obtain direction cones, which represent the available approach directions for that cluster. Unfortunately this approach does not generate an absolute solution, but good suboptimal solutions can be found.

System Critique. This is an important and highly relevant piece of research because it considers the critical problem of probe accessibility, first described by Kawabe, Kimura and Sata (section 4.2). This work describes how probe access directions can be found for components that can be described within the solid modelling domain and proposes that this information can be used to both determine component orientations and select probes.

The method described is capable of determining all possible probe access directions for every point on every feature, which gives the inspection planner maximum flexibility for selecting component orientations and probes and also for distributing probing points. However this approach has the inherent disadvantage that probe approach directions are computed using a half-line probe abstraction, which can only prove that surfaces are inaccessible and not the opposite, which requires further analysis. A further problem is presented when a surface cannot be inspected from a single approach direction, although the research team is considering an alternative method, which allows the feature to be segmented and re-analysed.

The computation of GACs is admitted to be much more expensive than the computation of LACs, but it is suggested that this type of approach could be provided as a tool by an inspection planner, which could be invoked whenever necessary. Viewed at this level, this is undoubtedly a powerful approach that could form the basis of an automatic inspection planning system.
4.15 Commerclally Available Inspection Planning Systems

The most sophisticated inspection planning system on the market is represented by the Valisys suite of software tools from the Valisys Corporation of Santa Clara, California, which is a subsidiary of the FMC Corporation. The software tools that are currently available include [125]:

V:Check verifies that dimensioning and tolerancing conforms to standard and is consistent with part geometry

V:Gauge creates a softgauge which is a software model of the worst case mating part derived from the dimensions and tolerances applied to the part

V:Tolerance verifies that mating parts will assemble under worst case tolerance conditions and automatically specifies fixed and floating fastener tolerances

V:Path creates inspection paths for Coordinate Measuring Machines or machine tool probes

V:Interface communicates inspection paths and measured data between CMMs or machine tools

V:Inspect executes Valisys inspection paths and collects measured data to allow the creation of as-built part comparison

V:Qualify compares measured data with softgauge, makes the pass/fail determination and provides a complete inspection analysis

V:Track uses inspection results for statistical process control analysis to provide real-time process monitoring

V:Control allows a series of machining and inspection steps to be executed by invoking Valisys Control Language commands
Each of these tools can be integrated with a commercial CAD system and Valisys has negotiated marketing agreements with a variety of CAD vendors including McDonnell Douglas [126] and IBM [127].

The Valisys tool that is most relevant to the work described in this thesis is V:Path which creates a generic inspection path that can be run on any inspection device. This achieved by analysing the softgauge generated by V:Gauge, which ensures that the inspection process incorporates checks of all important features. V:Path eliminates the need for on-line CMM programming as the inspection path is defined using the graphics capabilities of the 3D CAD host and can be previewed by the operator for modification.

This is clearly a very powerful suite of software tools which is bound to make a big impact in industry. However the product literature does not provide a satisfactory explanation of the actual capabilities of each tool and it is therefore difficult to assess the product critically. The softgauge concept is the strongest feature of the system, which should overcome the problems associated with interpreting the geometrical tolerancing standards through the system's ability to construct a model from the measured constraints and to assemble the model with the softgauge [127,128].

Apart from Valisys, there are very few other inspection planning systems on the market and none that offer the high product functionality of Valisys. However, two products which have found many applications in the UK include CIM CMM from CIMLINC [129] and Automeasure from Computervision [130]. Both products have a similar specification, which includes integration with a CAD system, probe libraries, graphical display of the probe path defined by the operator and output in a neutral format. The CIM CMM product generates DMIS output, whereas Automeasure generates its output in NDF (refer to section 15.3). Both products are highly capable sys-
tems that assist rather than automate the inspection planning task.
Chapter 5

The Research Objectives

5.1 Introduction

The purpose of this chapter is to state and explain the objectives of the research and to describe the issues that are involved in achieving them.

5.2 The Objectives of the Research

The objectives of the research are:

1. to research and implement a method for automating the entire Inspection Plan and Code Generation process of the Design to Manufacture Environment for a Coordinate Measuring Machine (CMM) with the support of a Product Modelling System.

2. to define and realise the role of an automated inspection planner within a highly-integrated Design to Manufacture Environment and to participate in the design of a Product Model Data Structure concentrating on those aspects related to the inspection activity.

3. to undertake experiments to establish the efficiency of a prototype Inspection Plan and Code Generator operating both individually and as part of the Design to Manufacture environment.

5.3 The Automation of Inspection Plan and Code Generation

Inspection plan and code generation is a process that determines exactly how a component should be inspected on a particular measurement device and, when required, generates the part programs to control it. In many respects it is broadly similar to the parallel activity of process planning and code generation for machining,
as many of the steps involved in the two processes are equivalent. However, in contrast to the machining process, relatively little research effort has been directed towards Inspection Plan and Code Generation, until recently when it has been observed that the number of research programmes concentrating in this area has increased dramatically. This has possibly been caused by a combination of factors related to the market forces operating on industry and the amount of progress that has been made in the parallel research area of process planning and code generation for machining. Consumer-led pressure for higher levels of quality and greater product variability have forced the manufacturer to increase the level and frequency of inspection and to reduce batch sizes, which both serve to increase the load on the inspection planning activity. Increasingly more automated planning systems for machining have also had the effect of reducing the lead-time for machining but as yet without a corresponding reduction in the lead-time for inspection. There are indeed many similarities between the problems that must be solved in machining and inspection, but as yet very little effort has been devoted to transferring this knowledge between the disciplines.

The objective of this research programme is to research and implement an Inspection Plan and Code Generator specifically for Coordinate Measuring Machines, which are currently the most numerous type of computer-aided inspection machine in the UK manufacturing industry [131]. This type of machine, if provided with a list of constraints to measure and a description of the part, would require the following types of information to be planned before a part program could be generated:

1. axis system and datum setting operations,
2. measuring and probing operations,
3. component and probe orientations,
4. fixturing methods,
5. probe configurations,
6. probing point coordinates,
7. safe rapid paths between probing points.

The issues involved in planning these types of information are as follows:

1. Planning the axis system and datum setting operations. Unlike the datum system on a machine tool, there are two distinct components to the datum system of a CMM; the axis system and the datum point. The axis system, or reference frame, is an orthogonal axis system which indicates the orientation of the component (figure 5.1) and can also be used to relate certain types of measurements. The datum point is used only to locate the position of the component and should not be confused with the datum faces that may be specified by certain types of geometric tolerance. It is, however, quite likely that datum faces, if specified, would be used to set up axis systems and datum points, as these items are set up by probing key surfaces on the component and by submitting the resultant surface representations to CMM-based controller operations, which construct the required axis systems and datum points. An automated planning system for this task, would therefore be required to select these key surfaces, determine how they should be probed and identify the appropriate controller operations to construct the axis systems and datum points.

2. Planning the measuring and probing operations. For any constraint to be measured on a CMM, the machine first needs to probe the constrained surfaces so that they can be represented within the CMM controller as mathematical entities
such as lines, circles, planes or cylinders. These Surface Representations can then be analysed or related to each other so that the required characteristic under constraint can be calculated and compared with the specified value. An example of this is illustrated in figure 5.2, where a simple linear constraint has been applied between a cylindrical and a planar surface (figure 5.2a). The probing operations under these circumstances might typically consist of a four points on a circle for the cylindrical surface and three points on a line for the planar surface (figure 5.2b) and the measuring operation would be a routine that calculated the perpendicular distance between the centre-point of a circle and a line when projected on to the Z plane of the component axis system (figure 5.2c). In conclusion, an automated inspection planner would therefore need to determine the surfaces that are under constraint and how they should be probed and represented. It would also need to select both the specific CMM-based routine required to analyse or relate the Surface Representations and the most appropriate method to output the result.

3. Planning component and probe orientations. Probe and component orientations are important because in essence they represent the relationship between individual surfaces and the probe, which has a profound effect on the quality of the Surface Representation obtained. The probe and component should, in theory, be oriented to allow the probe access to a maximum area of the surface so that the resultant mathematical entity represents the surface as accurately as possible. This is largely made possible by the fact that the majority of automated-CMMs are equipped with motorized probe heads (figure 5.3), which allow the probe to assume a wide variety of orientations under automatic control. A secondary objective of this activity is to minimize the number of component orientations
required to measure the part accurately, as few CMMs are presently equipped with rotary tables, which means that the part would need to be re-setup by the operator or loading mechanism, which is a time-consuming activity. An automated inspection planner would therefore need to analyse each surface to determine the probe approach direction which offers maximum access to the surface so that the relationship between the probe and the component could be ascertained.

4. Planning the fixturing methods. Once the required component orientations have been planned, the fixtures necessary to hold the part in these orientations should be designed. Fortunately the inspection process, by its very nature does not require heavy-duty fixtures that prevent component deformation under extreme cutting conditions, but it does require fixtures that support the part in the same position and orientation for the entire inspection cycle. The initial position and orientation are not critical, as these are determined automatically as a consequence of the axis system and datum setting operations that were planned earlier. As a result of these less stringent requirements, a simple fixture will usually suffice, which has the added benefit of providing good accessibility to the probe.

5. Planning the probe configurations. There are many different types and styles of probe styli that can be used on a CMM, but the spherically-tipped stylus is used for the majority of applications. However even this simple probe can be varied in length and radius and it is the objective of this activity to determine the most appropriate stylus configuration (figure 5.4). The choice of stylus length is governed by the extra access to a surface that a longer stylus might offer but limited by the loss of probe sensitivity as length increases. An alternative and
preferable option is to increase the reach of a probe by the use of extensions bars, which can be inserted between the probe head and the probe. This option has negligible effect on measurement accuracy. The choice of stylus radius is also controlled largely by surface access considerations, but the measurements of a very small stylus tip will tend to be adversely affected by micro surface deformations on certain types of surfaces. In conclusion, an automated inspection planner would be required to determine the most appropriate probe configuration for each surface, without compromising the measurement accuracy.

6. **Planning the probing point coordinates.** Although there are no strict guide-lines for distributing points on a surface, it is understood that the best mathematical representation of the surface can be obtained by distributing the points across the surface in a uniform manner so that neighbouring points are approximately equidistant (figure 5.5a). Those surfaces that are modelled by non-3d representations such as lines and circles, should also have uniform point distributions, but constrained to the shape of the representation (figure 5.5b).

7. **Planning collision-free probe paths.** Clearly the paths that the probe follows between the points must be as short as possible without causing the probe to collide with the component or for that matter with any other part of the environment such as the machine or fixtures.

Once this information has been planned, the next stage is to generate a part program so that the component can be inspected on the CMM. This is a technically straightforward process complicated by the fact that there are many CMMs available on the market and although they are largely similar in structure and design the same cannot be said of the programming languages. Therefore any research into Inspection Code Generation must be capable of accommodating the fact that there are many different
CMM programming languages in use and that there is the possibility that a company will require part programs to be generated in a variety of languages depending on the machines available.

5.4 The Integration of an Automated Inspection Planner within a Design to Manufacture Environment

A skeletal Design to Manufacture Environment has been defined by the Information Support Systems for Design and Manufacture research project, which consists of the following elements:

- a feature-based design environment,
- a machining plan and code generator,
- an inspection plan and code generator,
- a manufacturing data analysis facility.

The purpose of this environment for the project, is to experiment with the integration possibilities that exist between design and manufacturing activities in order to establish and realise the full potential of a highly-automated manufacturing cell.

With respect to the work described in this thesis, the Design to Manufacture Environment will be used to focus on the integration issues that are related to the inspection planning activity. These issues centre on the extent of the knowledge and data that the inspection planner requires and the type and content of inspection information that should be generated for the benefit of other design and manufacturing applications.

Integration is achieved through a Product Modelling System (PMS) which is a powerful data management tool that allows complex data structures to be represented and accessed by a variety of software applications. As its title suggests, the underlying
The purpose of the PMS is to represent Product Model Data Structures and one of the principal objectives of the research project is to define and implement a data structure that adequately describes all the geometrical and technological data associated with a product. The issues that impact on the inspection planner are the research and implementation of data structures that adequately describe the information generated by the planner and the integration of these structures with the Product Model Data Structure.

5.5 Practical Experiments Based on the Inspection Plan and Code Generator

The purpose of the practical experiments is to prove that the research ideas, when embodied in prototype software, are capable of processing real and synthesized components chosen purposely to test the capabilities of the system. These experiments should help to highlight the strengths and limitations of the approach and will allow comparisons to be made with other planning and code generation methods.

Additional experiments will also be carried out to investigate the rôle the Inspection Plan and Code Generator plays in the skeletal Design to Manufacture Environment. A component will be designed and planned for manufacture by the system before it is machined and inspected by a small manufacturing cell. The results of the inspection cycle will be fed back to the Product Model so that they can be analysed and any errors corrected.

5.6 The Research Environment

This work is funded as part of a large SERC/DTI funded research project between the Loughborough University of Technology and The University of Leeds (refer to Appendix A).

The project operates within a software environment that is based on the use of Ada [132] for all prototype software. This language is recognized for its highly
structured design and its excellent interfacing characteristics through the use of strong typing and separately compilable packages. This is countered by relatively poor performance in multi-tasking applications and a relatively heavy compilation requirement. However, on balance, the language is deemed highly suitable for the type of work undertaken by the research project.

The hardware used by the project at Loughborough consists of two Sun 3/60s and a Sun 3/75 supported by a Sun 3/160 file-server, which all run under version 3.5 of the Unix Operating System. The Loughborough network is connected to a Vax 8600 at Leeds University by the Joint Academic Network (JANET), which is also supported by a network of Sun 3/60s.

A Ferranti Merlin 750 CMM supported by a Hewlett-Packard 9000 controller is used for all the practical inspection research. This machine is equipped with a Renishaw PH9 motorized probe head fitted with a TP2 5-way touch trigger probe and a simple fixturing kit.

The machining experiments are carried out on a Wadkin V4/6 Machining Centre and a Dean, Smith and Grace lathe retro-fitted with a GE2000 NC controller (figure 5.6). An additional Wadkin V4/6 Machining Centre is used for practical research at Leeds.

5.7 The Constraints Imposed upon the Research

This chapter describes an ambitious set of objectives for a Ph.D. research project and this section qualifies these objectives by describing known limitations which were instigated to optimize the available resources of time and manpower without curtailing the ability to achieve the stated objectives.

The functionality of the prototype Inspection Plan and Code Generator is res-
stricted in the following respects:

- the target CMM is equipped with a motorized probe head (figure 5.3) and without a rotary table,

- simple fixturing methods (adhesives, Blu-tack etc.) are used whenever possible as the incorporation of a fixture design facility is beyond the resources of the research programme,

- a single probe configuration using a sphere type stylus is used throughout the inspection cycle as the CMM is not equipped with an automatic probe changing unit.
Chapter 6

Inspection Plan and Code Generation
within a Design to Manufacture Environment

6.1 Introduction

The Inspection Plan and Code Generation research has been undertaken as part of a larger project entitled the Information Support Systems for Design and Manufacture (ISS project) which is described in Appendix A. The principal objective of this project is to investigate the ways in which a highly-automated design to manufacture environment can be built and integrated, and the purpose of this chapter is to describe this research and the role that the Inspection Planning work played in the achievement of the objective.

6.2 The Objectives

Inspection Planning is a fundamental part of any design to manufacture environment because inspection is the means by which a manufacturing system maintains the quality of its output. Ideally the system will be configured to manufacture the product right first time [133,134] but the physical environment is not constant and consequently defect-free output cannot be guaranteed [135]. The inspection process needs to be planned for the same reasons as other manufacturing applications and its output should be used to detect and correct manufacturing errors.

The objective of this chapter is therefore to discuss the knowledge that has been gained from establishing Inspection Planning within a design to manufacture environment and to explain the means by which integration has been achieved between the related design and manufacture activities.
6.3 Research Elsewhere

Research projects of the scale required to investigate and implement design to manufacture systems are not commonplace in the academic environment, but nevertheless some worthwhile research has been undertaken and reported.

The most celebrated work in this area is represented by the so-called *Quick Turnaround Cell* (QTC) research of Chang, Anderson and Mitchell [75,105] which concentrates on the research and implementation of an integrated manufacturing cell that incorporates four tightly coupled modules: a feature-based design system, an automatic process planning and part programming system, a cell controller and a vision monitoring and inspection system. The principal objective of the cell is to produce parts quickly and correctly, which is argued to be especially applicable in a research and development environment where quick part turnaround results in a shorter development time. Data files are used as the method of integration between the functional parts of the system which allows each sub-system to be developed incrementally without having a drastic effect on the overall system structure. There are two major human interfaces within the system, which are associated with the design module and the cell controller as the rest of the QTC is designed to be fully automatic. The design system is based on a features approach, where each geometric feature is loosely related to machining operations such as a hole, slot or countersink. Dimensional and geometrical tolerances are associated with the size parameters of each feature and form tolerances can be associated with individual surfaces. The geometry of the component is maintained in the feature-based form and it can be converted on request into a boundary representation for visualisation or analysis purposes. The functionality of the process planning and part programming system consists of process selection, tool selection, process sequencing, fixturing method planning and NC cutter path genera-
Each of these activities achieves its task using a combination of feature-based information and expert system knowledge. The purpose of the cell controller is to coordinate the cell, schedule the jobs, maintain data-bases and to provide an interactive operator interface. The vision-based inspection system is designed to monitor the manufacturing process by checking that the tools, fixture and process are operating correctly and that the final part has been manufactured within the specified tolerances (refer to section 4.7). On balance, the QTC project has researched a comprehensive and detailed design to manufacture system that has been tested using a simple synthesized component (figure 6.1a), which was machined and inspected within one hour.

There are no other projects that have reported research into design and manufacturing environments on the same scale as either the QTC project or the ISS project, but several projects have considered the narrower but relevant field of inspection planning and data analysis.

Mullineux et al of Brunel University [121,122] are investigating the benefits of installing a bi-directional intelligent link between design and manufacture, and in particular between a CAD system and a Coordinate Measuring Machine (CMM). The approach adopted by the team is based on the concept of a blackboard, which is an idea that has arisen from work into Artificial Intelligence and is used to refer to a central data structure capable of communicating with a number of other systems. So far the blackboard has been used to integrate a CAD system with an inspection planning facility (refer to section 4.12) and to analyse the results of the inspection process. The analysis of measurement data is explained using an example based on a die-block that is essentially comprised of five holes in a block. The measured dimensions of the holes are returned by the CMM to the blackboard so that two global model spaces can be constructed; one to hold the CMM data the other for the original CAD representa-
tion. The system then attempts to align the holes of the two model spaces by moving the measurement model space until it fits within the prescribed tolerances of the designed model space. If this is achieved the component is accepted, otherwise further analysis is carried out to elicit the cause of the error. This process starts by considering the method of manufacture, which in the case of the example was by jig-boring. The system can then, using rules, identify which jigs were out of alignment by manipulating each of the holes independently until the two model spaces match. The example given for the analysis of inspection data is rather contrived but it does highlight the capabilities of the blackboard environment when dealing with this type of problem and it is clear that this approach could be extended to a wider variety of error-correction scenarios.

Finally, the research of Galm and Merat [116] is interesting in the first instance because it combines the use of fast low-accuracy inspection devices such as range imagers with higher-accuracy but relatively slow CMMs (refer to section 4.8). The system first employs the fast low-accuracy devices to inspect the component and to construct an object model based on the measurements which can be compared with the corresponding reference model constructed by the designer. The system checks each feature of the object model against the reference model and rejects any feature that falls outside the tolerances specified by the reference model. If there are any features with tolerances that are smaller than the accuracy associated with the object model they are inspected by a higher accuracy inspection device such as a CMM and the process is repeated. This represents an ingenious method for optimizing the inspection resources of a manufacturing cell but does not attempt to determine either the cause of an error or the appropriate correction.
6.4 Assumptions and Limitations

Depending on the particular industry, design to manufacturing environments can incorporate a very wide variety of tools and processes which would be impractical to accommodate in this experiment. For this reason, a skeletal Design to Manufacture Environment is researched by this project, which is restricted to the following processes:

- Design
- Machine Operations Planning and Cutter Path Generation for a 3-axis Machining Centre
- Inspection Plan and Code Generation for a Coordinate Measuring Machine
- Manufacturing Data Analysis

6.5 The Method

A skeletal Design to Manufacture Environment (DME) has been researched and implemented according to the architecture illustrated in figure 6.2, which shows the DME supported by the Product Modelling System (PMS).

6.5.1 The Product Modelling System

The PMS is a revolutionary concept in data management that has been researched at the University of Leeds (refer to section 2.4.3.2) and is used by this project to represent and maintain a Product Model that describes all the geometrical and technological data associated with a product. The PMS is based upon a software tool known as the Structure Editor that permits data structures to be described by a meta-structure and data to be represented within the structure as an instance. For example, if a Product Model Data Structure for commercial vehicles was constructed then that
would be a *meta-structure* and an *instance* could be represented by a Product Model of a Ford Transit van.

The Structure Editor is able to describe complex data structures because it is equipped with a set of data constructs which include lists (*lis*), collections (*col*), selections (*sel*) and pointers. An example of the use of a collection occurs when a car is described as a collection of body, engine, transmission and suspension. An example of the use of a selection occurs when a car body is described as a selection of either saloon, coupé, hatch-back, estate or convertible. The purpose of lists and pointers is self-evident. A facility known as lambda calculus is available which allows data structures to be parametrized, which is especially useful when defining features.

Research into Product Model Data Structures has been underway since mid-way through the project that preceded the ISS research programme, and as the research has evolved it has been observed that a simple pattern repeats itself throughout the Product Model Data Structure [136]. This pattern is known as the Framework (figure 6.3) and may be regarded as parametric where the name of the *entity* and the *entity description* are the parameters. This Framework is instantiated four times in the Product Model Data Structure for each product, assembly, component and feature.

Each *entity* has four types of data associated with it:

- *Data about entity data* may include a part number, version information and a list of references to pieces of third party software that contain data pertaining to a particular entity.

- The *entity specification* is the list of requirements that the designer specifies and which must be satisfied by the *actual entities*.

- Each *entity specification* may be satisfied in one or more ways: each of these will be described by a separate *entity definition*. 


Chapter 6 - IPCG within a DME

The data about actual entities such as measurements and historical data will be kept in the actual entities node.

Five pieces of data of data are associated with each entity definition:

- **Name** will act as an identifier.
- The **entity description** will depend upon the type of the entity.
- The **relationship graph** will be generated from the entity description and represents the dimensional and geometrical constraints that have been applied.
- **Planned processes** will be used by the manufacturing applications.
- Data which will be used by the Data Analysis and Error Correction facility will be described in the decision network.

Each actual entity has a reference to the definition and a measurement graph which is equivalent to the relationship graph but represents the measurements generated by the Inspection Plan and Code Generator.

Figures 6.4 to 6.7 illustrate the Product Model Data Structure and it can be observed that the Framework is repeated four times as mentioned previously. The jagged lines represent intermediate pieces of structure that have not yet been formalised but for the purposes of the experiment have been defined as:

- **product range**: lis (product)
- **product description**: sel (component, assembly)
- **assembly description**: lis (part : sel (component, assembly))
- **solid**: lis (feature)

The highest or *ROOT* level (figure 6.4) holds information that relates to the whole product or company and this includes the feature data-base and manufacturing information, which describes the plant, methods and rules. A *component* may be either a
composite or a solid made from a single material.

6.5.2 The Design to Manufacture Environment

The skeletal Design to Manufacture Environment (figure 6.2) of this project supports the following six activities:

1. Design
2. Process Planning
4. Inspection Plan and Code Generation
5. Machining and Inspection of the Prototype
6. Manufacturing Data Analysis

6.5.2.1 Design

Products are constructed by the designer within the Product Modelling System using a combination of features and CSG half-spaces (refer to section 2.2.4). The features are company-specific and represent geometric entities that are commonly used in the product range of the company. The feature definitions are held at the ROOT level of the Product Model (figure 6.4) and are constructed from parametrically-controlled CSG half-spaces. Manufacturing information can be associated with each feature, and in the current implementation of the DME this consists of knowledge and data for the purposes of the machining and data analysis applications.

Dimensional and geometrical constraints as defined in BS308 [19,20,21] and ANSI Y14.5M [22] are automatically applied to the geometry by the Relationship Graph Evaluator (RGE) [137] and are held under the relationship graph node of the appropriate entity. The relationship graph is structured so the designer can adjust the
size and shape of the geometry by editing the dimensions of the relationship graph.

The designer is supported by a selection of tools, which include linear tolerance analysis, a hidden-line wire-framer and a shaded-image ray-tracer. The wire-framer and the ray-tracer can be used for visualisation purposes and are based on the Spatially Divided Solid Modeller (refer to Appendix B).

6.5.2.2 Process Planning

Process Planning is included in this list because it is an essential part of any Design to Manufacture Environment. However, it is omitted from the skeletal DME of the project because the Alvey Design to Product demonstrator was also based at Leeds and Loughborough and was committed to researching the area of generative process planning. Therefore in order to avoid duplication of effort, the two projects agreed to keep a watching brief on these potentially transferable applications which also included the Inspection Plan and Code Generator and Manufacturing Data Analysis.

6.5.2.3 Machine Operations Planning and Cutter Path Generation

The principal objective of this activity [138,139] is to determine the component orientations, fixturing methods, cutting tools, cutting parameters and cutter paths required to machine a component. It assumes that the process planner has selected a 3-axis Machining Centre as a suitable processing method for manufacture and analyses the component using a combination of manufacturing information attached to features and geometric analysis. The feature-based manufacturing information includes fixturing strategies, operation types, tool types and a parametrized tool path plan [140]. The actual tools used and the cutting parameters (feeds, speeds etc.) are determined using a rule-base and manufacturing data held by the manufacturing information node at the ROOT level of the Product Model (figure 6.4).
Those parts of the component that are not constructed from features are planned using similar methods but based on geometric analysis techniques developed from the research of a previous Leeds University project [79]. This hybrid approach to Machine Operations Planning and Cutter Path Generation allows the system to benefit from the considerable advantages of both methods of analysis as the features approach is fast and computationally economical and the geometric analysis approach can accommodate a wider domain of geometry. However this interaction between the two activities is only achieved through the intervention of the Product Modelling System which allows the systems to communicate with each other through the planned processes data structure (figure 6.3).

6.5.2.4 Inspection Plan and Code Generation

Inspection Plan and Code Generation (IPCG) is the subject of this thesis and is therefore only briefly summarized in this section as it will be explored in detail during forthcoming chapters.

The Inspection Planner analyses the geometry and the constraints of the component in order to generate an Inspection Plan which describes the component orientations, probe configurations, probing points, probe paths and all the measuring, probing, axis system and datum setting operations required to inspect the component. This Inspection Plan can then be translated into a part program by the Inspection Code Generator and transferred to the Coordinate Measuring Machine (CMM) for execution. The measured results are transferred back to the host after the component has been inspected and are stored in the measurements graph (figure 6.3) for subsequent analysis by the Manufacturing Data Analysis application.

The IPCG operates using a combination of rules to convert dimensional and geometrical tolerances into measuring operations and geometrical analysis to determine
Chapter 6 - IPCG within a DME

probe approach directions, probing points and probe paths.

6.5.2.5 Machining and Inspection of the Prototype

NC and CMM part programs are transferred to the Machining Centre and the CMM respectively, using standard DNC technology developed by the project. This process was regarded as no more than an enabling activity for other research tasks but is nevertheless worthy of research in its own right.

6.5.2.6 Manufacturing Data Analysis

The principal objective of the Manufacturing Data Analysis (MDA) facility is to analyse the measurement results from inspection in order to analyse any errors detected by the CMM and to suggest probable causes [141]. The MDA facility uses a feature-based approach where decision networks (figure 6.7) are associated with each feature and operates by analysing each measured constraint in order to classify them into one of five states [142] which include:

- upper-fault,
- upper-warning,
- satisfactory,
- lower-warning,
- lower-warning.

The decision network can then be used to identify the fault codes using an influence diagram approach [142], which refer to a library of faults held in the manufacturing information node at the ROOT level of the Product Model (figure 6.4). Each fault code is associated with collection of information which includes fault type, cause, action and probability [143]. The user is then presented with an ordered list of probable
causes and suggested actions for each fault found on the component.

The MDA facility is currently being extended to accommodate errors that might occur on non-feature-based geometry and also to incorporate a facility for monitoring the manufacturing process.

6.6 The Results

An experiment based on a simple industrially-sourced component supplied by the Glacier Metals Company (figure 6.1a) was carried out as the final demonstration of the ISS research project to the ACME/SERC review panel on March 30, 1990. In a two hour live demonstration session, the component was designed using features, planned for machining and inspection, machined and inspected and the measured results analysed for errors. In order to perpetrate an error into the system, the members of the review panel were given the opportunity to exchange one of the cutting tools selected by the Machine Operations Planner, and this was eventually detected by the Manufacturing Data Analysis application after the part was inspected. Although the experiment was technically successful, some software errors were experienced during the demonstration as this was the first occasion on which the entire project software base had been integrated into one executable file. Fortunately these errors were not serious and were eradicated by the time the experiment was repeated in front of the project’s industrial collaborators on June 5, 1990.

6.7 Discussion

This chapter describes an ambitious programme of work that has successfully implemented a highly-automated and integrated Design to Manufacture (DME) Environment supported by a Product Modelling System (PMS). The DME is necessarily restricted in scope as it would otherwise be impractical to consider all the poten-
tial manufacturing processes, but it does represent a relevant and highly topical vertical slice of the design to manufacture process.

There are many current research issues under consideration in the design environment (refer to section 3.2) and as yet very few solutions. However, although the design environment researched and implemented by the project does not incorporate a conceptual design facility, it does allow specifications to be drawn up (figure 6.4) and assists the designer to create actual product definitions using a feature-based CSG solid modelling system. A suite of tools are available to further assist the designer in this process including a tolerance analyser and visualisation aids.

Process Planning (refer to section 3.3.1) and Cutter Path Generation (refer to section 3.3.2) are fields of research that have received a considerable amount of research attention over recent years but an original and possibly revolutionary approach is proposed by the ISS project. This approach combines the strengths of both a feature-based and a geometric-analysis-based method which results in a highly-automated system that both plans all the information required to machine a component and generates the cutter paths necessary to produce a part program. The system benefits from the speed and efficiency of the feature-based knowledge and the wider geometric domain of geometric analysis and achieves full integration between the two approaches using the PMS. This research is only beginning to demonstrate its full potential and will be the subject of a future research programme.

Inspection Plan and Code Generation is the subject of this thesis and will be investigated in full detail throughout the following chapters. Suffice it to say at this stage, that the approach is capable of generating an Inspection Plan and CMM part program for a component using a combination of rules and geometric analysis and will be shown to represent a level of automation and integration that has not yet been
Chapter 6 - IPCG within a DME

achieved elsewhere. Inspection Plan and Code Generation is a central and strategic part of the DME because it analyses the same geometry and constraints used by the Machine Operations Plan and Cutter Path Generator to determine an inspection method and part program. The CMM is then used to inspect the machined component so that the measured results can be returned to the Product Model for analysis by the Manufacturing Data Analysis facility.

Manufacturing Data Analysis is a critical function of the Design to Manufacture Environment as it effectively provides a feed-back loop between the design and manufacturing activities. This has been achieved in this project using a feature-based approach that is capable of suggesting the possible causes of errors detected by the inspection facility. Currently the user is expected to examine and correct the causes of error either with the Product Model or the manufacturing environment but an automatic solution to this problem based on artificial intelligence techniques is expected to be the subject of future research.

In comparison with the other research projects investigating the design to manufacture environment, the strongest element of the ISS approach lies in the use of the Product Modelling System (PMS) as a mechanism for integration. The PMS allows complex data structures to be built that provide sufficient detail for highly-automated manufacturing applications such as those of the DME to communicate with each other bi-directionally. A Product Model-Data Structure has been researched, principally by personnel at the University of Leeds, which represents all the geometrical and technological information required to design and manufacture a product. The PMS is not restricted to supporting the applications described here and can be extended to incorporate third-party and commercial applications using the CIM Applications Architecture depicted in figure 2.10. The interaction between design and
manufacturing applications and the PMS will be the subject of substantial future research.
Chapter 7

The Integration Aspects of Inspection Plan and Code Generation

7.1 Introduction

The Product Modelling System is a data management concept that has been established by the Information Support Systems for Design and Manufacture research project, which provides a structured and accessible environment for the data and knowledge requirements of manufacturing applications such as the Inspection Plan and Code Generator.

This chapter describes the integration that has been established between the Product Modelling System and the Inspection Plan and Code Generator and discusses the opportunities that it has realised for the Inspection Planning research.

7.2 The Objectives

Inspection Plan and Code Generation like any other automated manufacturing application requires a substantial amount of information from both the design process and the manufacturing environment. Previous research efforts in this area have elected to satisfy these data requirements by integrating directly with commercial CAD packages and data-bases or by relying on substantial user interaction. Unfortunately this type of approach is becoming less acceptable as manufacturing industry strives towards Computer Integrated Manufacture (CIM) and the substantial benefits that this philosophy confers. Therefore it is an essential prerequisite of research into advanced manufacturing applications that data requirements should be served by a central common data repository and that generated knowledge and information should be fully accessible to related applications.
Chapter 7 - The Integration Aspects of IPCG

7.3 Research Elsewhere

The majority of Inspection Planning research initiatives have concentrated on the inspection issues without seriously addressing the problems of integration with external manufacturing applications. The research of Atkins and Derby [39] is the exception as the major objective of their work was to investigate the advantages of Product Definition Data to the Inspection Planning process. Product Definition Data (PDD) was researched by the US Air Force sponsored Product Definition Data Interface (PDDI) project and was designed to represent both the product’s geometric and non-geometric information which includes the data necessary for machining and inspection. However although the PDD adequately represents the data required for these activities there appears to be no facility within PDD for representing the manufacturing plan generated by machining or inspection and this is also the case with the parallel PDES/STEP initiative [144]. The inspection planner of Atkins and Derby avoids this issue by generating its output in the DMIS [120] neutral format (refer to section 15.3), which can theoretically be post-processed into any CMM programming language but is represented in a part programming language text format which does not facilitate analysis for error detection. Atkins and Derby tested their Inspection Planner using an industrially-sourced case-study and commented that PDD was able to supply all the part information required by the inspection application but that certain data regarding secondary datums was not found until after the experiment was nearly complete. They argue that this was partly caused by the organisation of the PDD, which was unnecessarily complicated through the requirement to include a large number of different entity types. They also question the capability of the PDD to accommodate components from a wider variety of manufacturing industries, but they do recognize that this type of approach provides the best opportunity for progress with highly-
automated manufacturing applications.

Steger and Garbrecht [145] analyse the role of information flows within the Computer Aided Quality Assurance environment which they define to include quality or inspection planning, off-line CMM programming and the evaluation and feedback of measurement data. They argue that quality data is becoming an increasingly important component of the CIM environment and that a central data store for the technological and geometrical product information with controlled access capabilities is a precondition for functionality. They conclude that quality-controlled CIM environments require CAD/CAM systems with open and expandable data-bases.

Finally there have been several research initiatives devoted to the development of a Production Modelling Environment and these are reviewed in detail in section 2.4.

7.4 The Method

The Product Modelling System (PMS) established by the project is described in sections 2.4.3.2 and 6.5.1. The purpose of this section is therefore to describe only those aspects of the Product Modelling System that impact upon the Inspection Plan and Code Generator (IPCG), which include the retrieval and storage of data that is both required and generated during the inspection planning process.

7.4.1 Retrieval of Data

The Inspection Plan and Code Generator requires three classes of data in order to operate effectively, which include the geometry of the component, the constraints associated with the component and a description of the CMM.

The IPCG clearly requires a solid model representation of the component because its principal objective is to automate the inspection planning process and this
necessitates the analysis of a solid model in order to be able to determine Probe Approach Directions (refer to Chapter 10) and collision-free Probe Paths (refer to Chapter 14). The solid model representation of the component is provided by the component description (figure 6.7) within the PMS which is essentially a graph-type data structure that describes a CSG representation of the solid (refer to section 2.2.4). The component geometry is retrieved from the PMS by a standard tree-walking function that walks the CSG tree and returns an Ada data structure that can be understood by the IPCG. However although a CSG representation is well suited to the requirements of designers for constructing objects, it is not particularly conducive to analysis by manufacturing applications such as inspection planning. For this reason an alternative geometrical representation is used which is known as the Spatially Divided Solid Model (refer to Appendix B). A Spatially Divided Solid Model (SDSM) of the component is created by a software application that is also integrated with the PMS which decomposes the CSG tree retrieved by the IPCG into an Ada data structure that is comprised of a list of cubic cells that are classified as either full (within the solid), empty (outside the solid) or boundary (on the boundary of the solid). This form of geometrical representation is especially conducive to analysis by manufacturing applications and its use is described in more detail during Chapters 8,10,13 and 14 of this thesis.

The retrieval of the constraints associated with the component is an essential prerequisite of an Inspection Planning system as the measurement of these constraints is the principal goal of the inspection process. To satisfy the requirements of the Inspection Planner, the constraints must be represented by a data structure that describes the type of the constraint, the nominals, tolerances and pointers to the constrained surfaces within the CSG description of the component. This information is
represented within the PMS as the Relationship Graph (figure 6.7) which is a graph-type data structure that is comprised of a list of dimensioned entities [137]. Each dimensioned entity represents a constrained surface or centre-line and is described by both a list of constraints and pointers to the constrained surfaces in the CSG representation of the component. The constraints correspond to the dimensional and geometrical tolerances listed in BS308 [19,20,21] and are generally described in terms of the nominal value, tolerances and back-pointers to the dimensioned entities which consequently make the structure recursive. The Relationship Graph represents a complete description of the information required by the Inspection Planner and is therefore retrieved by calling a procedure that walks the Relationship Graph data structure within the PMS and builds an equivalent Ada data structure that can be analysed by the Inspection Planner. This Ada data structure is known as the Inspection Planner Relationship Graph (figures 8.9 and 8.10) and is tailored for the requirements of the Inspection Planner as it is configured as a list of constraints that point to CSG half-spaces rather than as a recursive data structure.

A description of the CMM is required by the IPCG to allow the detection of collisions between the component and the machine. For this purpose the IPCG requires the key dimensions of the probe which include the radius and length of the probe body, the length of the probe stylus and the radius of the stylus tip. The PMS is well suited to representing and storing this type of information which would appear under manufacturing information at the ROOT node of the Product Model (figure 6.4), but as yet this has not been implemented. Currently the IPCG retrieves this information from an internal procedure which requires editing, recompiling and relinking if the data needs to be modified but can be tolerated for this research because only one probe is used on the CMM. Future work in this area will need to consider the selec-
tion of more than one probe (refer to Chapter 12) and this will necessitate the implementation of a data structure describing the CMM and its available probes.

7.4.2 Storage of Data

The IPCG generates a substantial amount of information during the inspection planning process and this needs to be represented within the PMS for analysis by other applications. This information includes both a description of the method of inspection known as the Inspection Plan and the inspection results generated by the CMM.

The Inspection Plan is a complex data structure (figures 8.4 to 8.8) that describes every aspect of the process required to inspect the part. At the beginning of the inspection planning process it is empty and it is gradually populated with data as each planner of the IPCG performs its task. When it is completed it can be stored for later use or translated immediately into a part program by the Inspection Code Generator of the IPCG (refer to Chapter 15). Currently the Inspection Plan is represented as an Ada data structure, which forms part of the IPCG, but ideally it would be represented under the planned processes node of the Product Model (figure 6.7) to facilitate analysis by other applications such as Manufacturing Data Analysis (refer to section 6.5.2.6). The PMS could be configured to incorporate the Inspection Plan but as yet research resources have not been allocated to this task.

The results of the inspection planning process are generated when the part program is transferred to the CMM and executed, where an actual measurement is generated for each constraint specified in the Relationship Graph. The measurement data is transferred back to the host computer and is theoretically incorporated within a copy of the Relationship Graph, known as the Measurements Graph (figure 6.7), which is associated with the data structure that describes each manufactured component. This
facility has been discussed and agreed by the project research staff but has yet to be implemented.

7.5 The Results

The Users Guide (refer to Appendix D) was based on a case-study which used a version of the IPCG which is integrated with the PMS and is able to retrieve the component geometry and the SDSM representation of the component. This case-study showed that integration between the PMS and the IPCG was feasible and that the geometry of the component could be successfully exchanged between the two environments. A separate experiment which used a simple component based on a block with a slot showed that the IPCG is also capable of retrieving and analysing the Relationship Graph which represents the constraints associated with the component.

7.6 Discussion

This chapter describes the interaction between the Inspection Plan and Code Generator and the Product Modelling System which acts as a data and knowledge management system for the Product Model Data Structure defined by the research project. The data requirements of the IPCG include the geometry of the component, the constraints associated with the component and a description of the CMM, whereas the IPCG generates an Inspection Plan and a list of measurements. Experimental results show that it is possible to represent and retrieve the geometry and constraints of the component but as yet the other information flows have not been implemented. It is suggested, however, that the PMS has the capability to represent the other data structures required and generated by the IPCG and that these could therefore be successfully integrated with the Inspection Planner.
Integration with the PMS confers substantial benefits to the Inspection Planner which centre on the fact that the Product Model is a neutral and highly detailed data structure.

The neutral aspect of the Product Model allows any other design or manufacturing application to be integrated with the PMS as illustrated by the proposed CIM Applications Architecture (figure 2.10) established by this research project. This means, for example, that a potential user of the Inspection Planner is not restricted to a particular solid modelling package but can integrate any commercial or experimental solid modelling system by creating an interface between its internal data structure and the component geometry using tools provided by the PMS. Conversely it also means that a user can elect to integrate an alternative inspection planner or inspection code generator by creating the necessary interfaces to the Inspection Plan data structure. Whatever the situation, the user can configure the PMS and the IPCG to suit the conditions and requirements of the company at the time of its installation.

The fact that the Product Model is a highly detailed data structure permits true integration between manufacturing applications because there is sufficient detail of information to be able to determine exactly how a particular component was machined or inspected. Unlike other approaches to product modelling this information is not represented as NC or CMM part programs but as true data structures that are designed to be walked and analysed by other manufacturing applications. This capability is especially important in the Design to Manufacture Environment of this project as it potentially allows the Manufacturing Plan and Inspection Plan to be analysed by the data analysis and error correction facility known within the project as Manufacturing Data Analysis (refer to section 6.5.2.6).
Chapter 8

Inspection Planning

8.1 Introduction

The principal function of the Inspection Planning research is to identify and implement a method for automating the process of generating an Inspection Plan, which in the context of this research incorporates the following activities:

- Operation Type Planning (refer to Chapter 9),
- Probe Approach Direction Generation (refer to Chapter 10),
- Component Setup Planning (refer to Chapter 11),
- Probe Setup Planning (refer to Chapter 12),
- Probing Point Planning (refer to Chapter 13),
- Probe Path Planning (refer to Chapter 14).

As each one of these planning activities is considered individually and in detail further on in the thesis, this chapter is purposely restricted to the investigation and discussion of higher-level Inspection Planning considerations. These include the critical issues of the retrieval of inspection information from the component description and its subsequent representation for related applications.

8.2 The Objectives

The major objective of an Inspection Planning facility is to automate the process of generating an Inspection Plan and this inevitably requires a substantial understanding of both the form of the component and the constraints applied to it, as these are the two most important factors involved in the planning of inspection. It can be established from the literature that two distinct methods have emerged for retrieving
manufacturing knowledge from a component and these are either by using parametric macros attached to features or by analysing the geometry.

The features approach involves the definition of a set of simple geometric entities such as pockets, slots, steps, holes, tapped holes etc. to which manufacturing knowledge can be associated such as the method to be used for machining or inspection. There are several inspection planners (refer to the section 8.3) that have been designed using this principle and they are all able to take advantage of the fact that inspection information can be inherited from the features which substantially reduces the amount of analysis that is required.

The geometric analysis route is undoubtedly more complex as it relies purely on the use of algorithms to analyse both the constraints and the geometry to which they are applied in order to determine the method of inspection. This type of approach requires a complete understanding of both the principles of inspection and the effects of the surrounding geometry on an inspection operation but it does have the advantage that it is not restricted to the inspection of essentially simple geometric features.

The first objective of this research is therefore to investigate the type of approach that is most appropriate to this solution of the Inspection Planning problem and to identify the strengths of limitations that might be incurred by that approach.

The other high-level Inspection Planning consideration is to design a data structure that allows the entire inspection cycle to be described and stored so that it can be both analysed by other manufacturing applications and converted into a CMM part program when required. Although this approach is not strictly essential in an Inspection Planning environment it is seen as a desirable feature of this research which would be facilitated by the Ada programming environment (refer to section 5.6) and the Product Modelling System (refer to Chapter 6) that have been established by the
Chapter 8 • Inspection Planning

Therefore the second objective of the Inspection Planning research is to design a data structure that allows the entire inspection cycle to be represented for the purposes of analysis and translation into specific CMM programming languages.

8.3 Research Elsewhere

Chapter four surveyed all of the relevant research in the area of Inspection Plan and Code Generation and the purpose of this section is to highlight firstly those research programmes that use feature-based methods for Inspection Planning and secondly those that use geometric analysis. The section is concluded by a review of research into the use and representation of Inspection Plan data structures.

One of the foremost and earliest feature-based inspection planners is that of ElMaraghy and Gu [102] who have researched a highly-automated system which is specifically designed for rotational parts. The basis of operation of the planner is the automatic classification of the component to facilitate feature extraction so that an expert system can interrogate a rule-base which incorporates feature-based logic to determine how the part should be inspected. The part is geometrically defined using conventional modelling tools, and the features are extracted from the component using the method of Syntactic Pattern Recognition so allowing the inspection methods associated with the features to be inherited by the component. During the inspection planning process, features are used firstly to determine whether the part can be measured on a CMM, which is achieved through a series of if-then rules based on the type of feature and the dimensions and tolerances that have been applied to it. Surface accessibility and probe selection are determined by a similar rule-based method. This is a relatively successful application of the features approach to inspection planning, which uses the features to make reasoned judgements regarding the inspection of the part. However although the authors claim that this approach is equally applicable to
the inspection of prismatic parts it is difficult to imagine how, for example, surface accessibility could be determined using feature-based reasoning in a three-dimensional environment.

The vision inspection planning system of Park and Mitchell [106] is also feature-based and attempts to simulate the way in which humans manually inspect parts or features. The inspection planner uses the features to retrieve both the constraints that should be measured and the appropriate viewing directions that were previously determined by the process planning function. This information allows the component set-ups to be planned by computing the overall tightness of the tolerances in each viewing direction and as such represents the only inspection planner that can compute this information. This is a good example of how simple inspection-related information can be associated with features and used in the inspection planning environment. However the constraints are associated with the parameters of the features and this is bound to restrict the freedom of the designer to allocate tolerances according to function.

The final feature-based inspection planner reviewed in this section is that of Atkins and Derby [39] which uses features both to design the part and to assist with the generation of probe paths, which is achieved by either one of two methods. If the feature to be measured is sufficiently simple like a hole, then a parametrically controlled macro can be associated with the feature and called up by the operator when the probe path is being determined. Otherwise, in the case of a feature such as a plane, the operator is requested to specify the probe path manually, although this process is assisted by the automatic provision of additional coordinate systems relative to neighbouring features so that the operator is not restricted to moving the probe with respect to the master coordinate system. Once the probe path has been completely
defined, the operator is able to animate the motions of the probe so that a visual verification can be carried out to ensure that the probe path is collision-free. This approach highlights some of the difficulties faced through using a feature-based approach, as the user is forced to supply information that cannot otherwise be provided by the features. However the authors indicate that this part of the process could be automated if the features were represented as solids.

Compared to the number of feature-based inspection planners, there are relatively few systems that analyse the geometry in order to determine the inspection information. Kawabe, Kimura and Sata [96] researched the earliest inspection planner and they instigated a system that is part manual and part automatic. The user constructs a solid model representation of the geometry and then uses this to specify the measurements that should be carried out by selecting surfaces and probing points on an image of the component using the cursor. The automatic part of the inspection planner can then utilize this data to determine collision-free probing paths by constructing a solid model of the swept volume of the movement of the probe and determining whether this intersects with the model of the object. If this is the case the probe is retracted to a safe space beyond the bounding box of the component where no collisions are assumed to occur. This a simple but highly effective method of determining collision-free probing paths, which is arguably one of the most labour-intensive and time-consuming inspection planning activities. However the approach only searches for collisions between the probe stylus and the component and takes no account of the collisions that might occur between both the stylus and the fixture and the probe body and the component. Nevertheless it is possible to imagine how the approach could be extended to accommodate these eventualities.

Spyridi and Requicha [115,124] are in the process of researching a geometric-
analysis based inspection planner and so far they have made worthwhile progress in the area of surface accessibility analysis. Their approach determines all of the directions in which a probe can approach each point on the component by a two stage process. Firstly Local Accessibility Cones (LACs) are constructed which represent every direction in which a probe can approach a surface without considering the collisions that might occur with the rest of the component. The computation of Global Accessibility Cones (GACs) which are a sub-set of LACs, is declared as a non-trivial exercise but has been achieved by an algorithm that determines a set of half-lines that do not intersect the global obstacles of a feature, which have previously been determined by computing the Minkowski sum of the LACs and the component. The GACs represent all the directions in which a probe can approach the component and this wealth of information is reduced to the minimum set of directions that will allow the component to be probed using cluster analysis. This information can subsequently be used to determine the workpiece orientations and probe configurations that are required to measure every constraint. This is by far the most sophisticated approach that has been established for analysing the geometry within an inspection planning system and the information that it generates can make a substantial contribution to the overall automation of the inspection planning process. However the approach assumes that the probe can be modelled as a half-line or ray with an end-point at the probe’s tip and extending to infinity along the probe’s axis (figure 4.6a). This clearly simplifies the analysis but does mean that the technique can only guarantee the directions in which the probe cannot approach the surface and not the opposite which requires further analysis.

Turning to the issue of the representation of Inspection Plans as data structures, it is clear that as yet there are no inspection planning research programmes that have
investigated this area in any depth. However it is worth noting that the principal objective of the research of Atkins and Derby [39] is to integrate inspection systems into the CIM environment using Product Definition Data (PDD). PDD has been researched by the US Air Force sponsored Product Definition Data Interface (PDDI) project and has been designed to represent both the product's geometric and non-geometric information which includes the data necessary for machining and inspection. However although the PDD adequately represents the data required for these activities there appears to be no facility within PDD for representing the manufacturing plan generated by machining or inspection and this is also the case with the parallel PDES/STEP initiative [144]. The inspection planner avoids this issue by generating its output in the DMIS [120] neutral format for inspection devices (refer to section 15.3), which can theoretically be post-processed into any CMM programming language but is represented in a part programming language text format which does not facilitate analysis for error detection.

8.4 The Method used to retrieve inspection information from the component

When the Inspection Planning research began, it was decided that the most effective starting point would be to research and implement a reduced-functionality Inspection Planner that could at least generate part programs and act as a test-bed for more sophisticated solutions as and when they became available.

This first version of the Inspection Planner was therefore capable of generating part programs by planning Operation Types, Probing Points and Probe Paths from a feature-based representation of the component. This type of representation was chosen in the first instance because some success had been achieved through the use of features in a planning system for machining operations [140] and it was felt that there was sufficient similarity between the two approaches to merit its application in
The first test-component using this approach was necessarily simple and was based on a single pocket in a block as illustrated in figure 8.1a. The pocket was represented as a parametrically defined inspection macro that allowed the principal dimensions of the pocket to be measured (figure 8.1b). The inspection macro was essentially an ordered list of probing points and paths that were organised so that each planar surface of the pocket was probed once and the cylindrical surface three times so that simple mathematical algorithms could be called up to calculate the relevant dimensions. As a Coordinate Measuring Machine (CMM) was not available to the project at this time the inspection macro generated its output in the NC code format, which was run on a Wadkin V4/6 machining centre equipped with a spindle-mounted probe.

Encouraged by the success of this admittedly simple experimental Inspection Planner, a second case-study was commissioned which attempted to extend the technique to a more complex component incorporating a variety of features. This test-component was known as the Bolster Plate (figure 8.2) and was loosely based on a component used in the plastics industry for moulding tubs for food products. This component incorporated six features which included:

- pocket,
- through pocket,
- hole,
- boss,
- slot,
- step.
Each feature was represented as a list of dimensional parameters, positional and rotational offsets and an inspection macro that described the probing points, paths and the measuring algorithms. By this time, the project had arranged access to a LK Metre-4 CMM at the nearby the LK Tool Company and so the code generator was reconfigured to output a part program in LK's CMIS programming language [146].

Although the part was successfully inspected using this method, the fact that more than one feature had been used, highlighted the restrictions that are inevitably placed on the allocation of constraints. This is because the inspection macro necessarily specifies the constraints that can be measured on a particular feature and this makes it impossible for the designer to allocate a different tolerance configuration. One possible solution to this problem would be to create an inspection macro for each possible tolerance configuration that could be applied to a feature but this would still prevent constraints from being applied between features and would also require a substantial manpower investment to implement each feature.

The inspection macro approach was therefore dropped in favour of a system that marked the beginnings of the Inspection Planner that is presented in the subsequent chapters of this thesis. This approach retained the use of features in order to represent a more complete description of the component geometry and also to act as a repository for Probe Approach Direction information, which was being used by the Component and Probe Setup Planners described in Chapters 11 and 12 respectively. It was necessary to extend the geometric description of each feature because the Inspection Planner was now deprived of the inspection macro information and would instead need to analyse the geometry in order to plan such data as probing points. This was achieved by describing each feature as a list of surfaces and centre-lines, where the surfaces were represented by a type (planar, cylindrical etc.), a surface normal and a
set of parametrized bounds. When the designer used such a feature in the construction of a component it was parametrized to generate the actual bounds of each surface and positional and rotational offsets were applied accordingly. Using this approach, dimensional and geometrical constraints could be applied to any part of the component simply by specifying the surfaces or centre-lines that should be constrained.

This approach was implemented and successfully used to plan the inspection of a component provided by one of the industrial collaborators of the project and the resultant case-study is included in this thesis as Appendix C. A part program was generated for this component and run on a Ferranti Merlin CMM which was generously provided for project use by GEC(FAST). Although the feature-based approach used for this case-study successfully overcame the problems of the previous feature-based system, it can be noted from Appendix C that an additional set of problems were encountered. Firstly it was impossible to represent the entire component using features because a great deal of the geometry was simply too complex to parametrize and would therefore be easier to define using conventional CSG primitives. Secondly it was found that interactions between features could not be tolerated because both the Probe Approach Directions and the geometry inherited by the feature could be corrupted. This was graphically highlighted by the case-study (refer to section C.4.1.3) when the Probe Approach Direction attached to a boss was corrupted because the wall of a neighbouring pocket obstructed the movement of the probe body therefore causing a collision. However neither of these problems pointed to a total rejection of the feature-based approach as the basic concept of associating low-level inspection information to features such as Probe Approach Directions was still valid except that it was now becoming necessary to supplement the feature definitions with pure geometric representations and also to employ geometric analysis techniques so that feature
interactions could be detected.

The inspection planning research therefore turned its attention to the problem of geometric analysis and after some lengthy consultation with practitioners in this field a possible solution was presented in the form of the Spatially Divided Solid Modeller (refer to Appendix B). This technique was suggested principally because of its successful application to the problem of NC code generation for machining [79] and also because a substantial package of query routines were available and could easily be integrated with the existing Inspection Planning software.

After an initial familiarisation period and the implementation of a series of short experiments to test the suitability of the approach, it was decided to proceed and implement a full working version of the Inspection Planner using geometric analysis, and the findings of this research form the basis of the subsequent chapters in this thesis. The new Inspection Planner was tested in the first instance by re-implementing the Bolster Plate using a CSG representation for spatial decomposition and this was followed by similar studies of a structural bearing supplied by The Glacier Metals Company (figure 8.3a) and a robot wrist motor body supplied by GEC(FAST) (figure 8.3b). The case-study based on the robot wrist motor body is the most recent and the most complex and is the basis of the Users Guide presented in Appendix D.

The following chapters in this thesis will endeavour to prove the efficacy of the spatial decomposition approach for planning inspection tasks, which supercedes the original feature-based approach. However it has been observed from the research that has been carried out so far that a successful inspection planner will need to be both judicial and sparing in the application of geometric analysis as although it is indispensable to a fully-automated planning system it is also computationally intensive. In this context features have the advantage that they can reduce the amount of geometric
analysis that is required providing they are uncorrupted by other features and
geometry. Probe Approach Directions provide a case in point as they require substan-
tial geometric analysis to compute but when features are used it is possible to predict
which Probe Approach Directions are not valid so that they can be removed from the
analysis. This type of facility would be straightforward to implement and integrate
within an Inspection Planning system but as yet it would still be difficult to determine
whether or not the feature had been corrupted.

8.5 The Method used to design the Inspection Plan

The underlying principle of the Inspection Plan research is to design a data struc-
ture that is capable of describing the entire inspection cycle so that it can be used and
stored with the other manufacturing plans that are associated with a component. This
approach has the advantage that because the manufacturing plans are stored in a neu-
tral format they can be retrieved and translated into any programming language,
whether it is NC code or for a CMM, which significantly widens the choice of
machines that can be used. The manufacturing plans can also be used to represent
part of the component's life-cycle data and because they are highly detailed they pro-
vide an ideal medium for a manufacturing analysis package to determine, for example,
how a manufacturing error could have been caused.

Because there are so many different CMM programming languages, it was
decided that the most practical approach to this problem would be to design a data
structure that allowed every detail of the inspection cycle to be recorded, rather than
attempting to study the requirements of each CMM programming language to design a
plan that suited them all. This was achieved in a stepwise fashion where those parts of
the Inspection Plan that were required at a particular stage in the development of the
system were implemented and tested by populating the structure and generating a part
The Inspection Plan illustrated in figures 8.4 to 8.10 represents the final version of the plan and was used for the case-study which was the basis of the Users Guide described in Appendix D. This plan is not discussed in detail in this section because it is a complicated data structure that requires careful explanation which will instead be given during each of the relevant chapters. Chapter 15 in particular tackles the subject of Inspection Code Generation and discusses the feasibility of translating the Inspection Plan into a CMM part program.

8.6 The Results

Case-studies were carried out using versions of the Inspection Plan and Code Generator that incorporated both the feature-based approach and the geometric analysis approach. The Micro-Switch Cover case-study (refer to Appendix C) uses the feature-based approach and highlights the inability of this method to adequately verify Probe Approach Directions. The Wrist Motor Body case-study (refer to Appendix D) uses the current implementation of the Inspection Plan and Code Generator which uses the Spatially Divided Solid Modeller to analyse the geometry. A satisfactory Inspection Plan was successfully generated for the Wrist Motor Body and the resultant part programs were executed and tested on the CMM.

8.7 Discussion

One of the principal issues that must be addressed in Inspection Planning research is the type of method that should be used to retrieve inspection information from the component geometry. This chapter describes the experiences of this research programme, which initially experimented with a feature-based system using parametrized inspection macros but later progressed to a geometric-analysis approach
when it was realised that the former method would be inadequate for complete automation of the inspection planning process. This was largely because the *raison d'être* of the feature-based approach is the association of inspection information with a pre-defined segment of geometry and it was found that this information was easily corrupted by the surrounding geometry of the part. The literature survey shows that several inspection planners have been researched using this approach and varying degrees of success are reported. ElMaraghy and Gu appreciate the difficulties of associating inspection macros with geometry and instead use an approach which classifies features into groups that have common high-level inspection methods. Park and Mitchell have researched an inspection planner for vision which uses features to inherit view direction information, although it is not clear how the system accommodates a view direction that is obscured by surrounding geometry. The inspection planner of Atkins and Derby uses features to retrieve pre-defined probing point and traverse path information, which is subsequently checked by the operator in case collisions have occurred.

Clearly a geometric-analysis approach can potentially overcome all of these problems as it is capable of analysing the component geometry to determine when surrounding geometry is likely to corrupt the inspection process. The approach described in this thesis is based on the use of geometric analysis, and the successes and difficulties experienced with this method will be discussed and criticised during the relevant chapters. However it is evident that an inspection planner that uses geometric analysis is difficult to implement because it does require a complete understanding of both the inspection process and the possible interactions between the machine and the component. This is reflected by the inspection planners that have used geometric analysis which show that this approach does allow higher levels of
automation to be achieved for tasks such as the determination of workpiece orientations, probe configurations and collision-free probing paths but at the cost of a greater computational overhead. Even then these current methods are not entirely suitable for a production environment because they are not capable of determining all the possible collisions that might occur on an inspection machine. However, the method described in this thesis will endeavour to prove that it is possible to detect all collisions and it is expected that future inspection planners will follow this trend.

The other high-level inspection planning consideration described in this chapter is the description of a data structure that will allow the inspection method to be represented and stored so that it can be analysed by other activities and when required translated into a particular CMM programming language. This is undoubtedly a highly desirable inspection planning feature as it does permit higher levels of integration to be achieved within a CIM environment. As yet the only other inspection planning research that has investigated this area is that of Atkins and Derby and their approach uses the Product Definition Data format to retrieve component geometry and constraint data, which as yet does not have the facility to represent the generated Inspection Plan. The Inspection Plan data structure used by the inspection planner described in this thesis is presented in this chapter and described and critiqued in the chapters that utilize it.
Chapter 9

Operation Type Planning

9.1 Introduction

The first major task of the Inspection Planning process is to determine the types of operations that must be carried out in order to inspect all the constraints that require measurement. The planner must, for example, determine how the axis system of the component should be set up, how each surface should be probed and how surface characteristics or relationships should be derived in order to generate measurements.

9.2 The Objectives

The principal difficulty with Operation Type Planning is that there are many different ways of measuring a constraint in coordinate measurement and as yet no scientifically-based guidelines for the CMM programmer or Inspection Planning system. In conventional CMM programming the choice of inspection method rests with the inspector and is based on previous experience relating to a variety of factors including the tightness of the tolerance, the measuring machine and the manufacturing method. Clearly an automatic solution to this problem would need to use artificial intelligence and research in this area would require detailed experiments into the selection and effect of measurement methods which is beyond the scope of this project. The objective of this work is instead to research and implement a less ambitious Operation Type Planner that is capable of determining the type of operations required to set up and measure each constraint on the component but is restricted to a particular set of measurement methods that is appropriate for general inspection tasks. In this way the Operation Type Planner is able to avoid the complex and poorly
understood issues involved in both the selection of measurement algorithms and the determination of sample point densities and can instead concentrate on generating operation types suitable for use by down-stream planning activities.

9.3 Research Elsewhere

This section opens with a review of the Inspection Planning research initiatives that have made a contribution to automating the Operation Type Planning process and concludes with a discussion of the research that is currently being directed towards software for CMMs.

Hopp and Lau [97,98] use a hierarchical task decomposition method to plan operation types and this is achieved by decomposing the goal of Inspect Part into a series of sub-goals to measure each tolerance. These sub-goals are further decomposed to determine the component features and consequently the surfaces that require measurement. A similar approach is used to determine the operations required to set the axis system. The number of points and mathematical representation used for each surface is determined by the so-called Probing Points control level and it is implied that the choice of the number of points could be adjusted depending on the tolerance and machine inaccuracies. This is a rigorously defined and powerful approach that considers all aspects of the Operation Type Planning problem but the content and form of the rules used to decompose these goals are not fully described.

ElMaraghy and Gu [102] have researched a generative feature-based inspection planning system for CMMs which classifies parts of the component into feature groups that affect the outcome of rules for determining operation types. A rule-base is also discussed for determining the dimensionality of the mathematical representation of each surface and this is based on selecting the lowest possible dimensionality in order to reduce the total number of points. No detail is provided regarding the planning of
operations for axis system and datum setting but the planner does identify datum features if they are specified by the designer. This inspection planner is dedicated primarily to rotational parts and is the only system that considers the selection of the mathematical representation to be used for each surface. However very little detail is provided about the rules used to determine the method for measuring constraints.

The inspection planning system researched by Park and Mitchell [106] is vision-based and uses rules to determine measurement methods depending on the type of constraint and the affected geometry. An example is provided which is based on the choice of measurement algorithm depending on whether the constraint is dimensional or geometrical. The component axis system and datum are set using part recognition methods which assume that the part is not rigidly fixed and is placed on a surface that is orthogonal to the camera. The part recognition method used by the system is of the hypothesis and verification variety which seeks sets of mutually consistent image-feature to model-feature assignments. This is currently the most complete solution to Operation Type Planning for inspection researched elsewhere, as it determines operations for both measurement and axis system setting supported by a simple case-study.

Turning to the CMM software viewpoint, it appears that there are two issues that are being addressed by the literature with respect to the algorithms used for measurement. Firstly it has been determined by a number of researchers that there are significant discrepancies between the results obtained from different measurement algorithms that purport to fulfil the same function [24,25,147,148]. This is partly caused by ambiguities in the tolerancing standards as described by Edson and Parry [149] and partly because of differences between the characteristics of the measuring algorithms as described by Weill [24]. However a significant amount of research continues into the development of more effective algorithms for measurement
Chapter 9 - Operation Type Planning

[150,151,152] and for the derivation of axis systems [153]. The second issue that has relevance to the problems faced by Operation Type Planning is that of the determination of sample point density which in simple terms is the number of points that should be used to represent a surface. At a workshop into Research Opportunities in Mechanical Tolerancing it was agreed that the effects of sample point density on the results of coordinate measuring systems are poorly understood [25] and that research should be carried out to characterize the effect of point density with respect to the manufacturing process and into the application of confidence factors in relation to point density. Results are beginning to emerge from research into a technique for determining sampling errors [154] and it is anticipated that further work in this area could result in the determination of the optimum number and distribution of coordinate points required to satisfy pre-determined tolerance and accuracy specifications.

9.4 Assumptions and Limitations

The Operation Type Planning research is restricted to using a fixed number of Probing Points for each Surface Representation and two-dimensional Surface Representations for all dimensional tolerances. These measures enable the system to plan operations which are suitable for the general inspection tasks considered by this research.

9.5 The Method

The main function of the Operation Type Planner is to determine three types of operations that are required to inspect a component, which include Probing Operations, Measuring Operations and Axis System and Datum Setting Operations.
Chapter 9 - Operation Type Planning

9.5.1 The Probing Operation

Measuring Operations and Axis System and Datum Setting Operations both require surfaces of the component to be probed and this activity is controlled by Probing Operations. The most important information associated with the Probing Operation includes the points to be probed, the collision-free path to be followed between points and the type of Surface Representation to which the probed points are fitted (figure 8.8). At this stage in the Inspection Planning process only the type of Surface Representation, the Working Plane and the quantity of Probing Points are required as the other data are determined at a later stage in the planning process.

The Surface Representation is a fundamental part of coordinate measurement as it is the means by which a surface is represented as a geometric entity so that it can be used, for example, to calculate linear and geometrical constraints for Measuring Operations. The types of Surface Representations that can be used for each type of surface geometry are listed in Table 9.1 and each one of these Surface Representations is determined by a software procedure at the CMM controller that processes the probed points using a method such as least squares fitting.

<table>
<thead>
<tr>
<th>Surface Type</th>
<th>Surface Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>planar</td>
<td>point</td>
</tr>
<tr>
<td></td>
<td>line</td>
</tr>
<tr>
<td></td>
<td>plane</td>
</tr>
<tr>
<td>cylindrical</td>
<td>circle</td>
</tr>
<tr>
<td></td>
<td>cylinder</td>
</tr>
<tr>
<td>conic</td>
<td>cone</td>
</tr>
<tr>
<td>spherical</td>
<td>sphere</td>
</tr>
</tbody>
</table>

Table 9.1: Surface Representations

The type of Surface Representation is always specified by the Measuring Operation or Axis System and Datum Setting Operation that commissioned the Probing
Operation and this process is explained in sections 9.5.2, 9.5.3 and 9.5.4 respectively. However depending on the type of the Surface Representation the Operation Type Planner must determine the Working Plane and the number of Probing Points.

The Working Plane is only required for two-dimensional Surface Representations such as *lines* and *circles* and represents a plane on to which the probed points are projected before they are fitted to the Surface Representation. The Working Plane corresponds to one of the planes of the component Axis System and is critical to the result of the Surface Representation as illustrated by figure 9.1. In figure 9.1a the Working Plane is normal to the axis of the cylinder and the points can be projected and fitted to a *circle* as required, but in figure 9.1b the Working Plane is orthogonal and the CMM controller will fail to fit the points to a *circle*. The Operation Type Planner determines the Working Plane for a *line* Surface Representation by selecting the plane that is orthogonal to both the surface normal and the longest axis of the surface. This ensures that the *line* represents the greatest possible area of the surface so that the resultant measurement or Axis System is as accurate as possible.

The number of Probing Points is also dependent on the Surface Representation as CMM point fitting algorithms specify a range of point quantities that can be fitted to each Surface Representation. The minimum number of points depends on the mathematical properties of the geometric entity, for example a *plane* must be represented by at least three points, and the maximum number of points depends on a combination of the memory capacity of the CMM controller and the capability of the point fitting algorithm. As yet there are no scientifically-based guide-lines for selecting the optimum number of points [25,154] and so the Operation Type Planner currently specifies a fixed number of points depending on the Surface Representation as shown in Table 9.2.
Chapter 9 - Operation Type Planning

<table>
<thead>
<tr>
<th>Surface Representation</th>
<th>Number of Probing Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>point</td>
<td>4</td>
</tr>
<tr>
<td>line</td>
<td>3</td>
</tr>
<tr>
<td>plane</td>
<td>4</td>
</tr>
<tr>
<td>circle</td>
<td>5</td>
</tr>
<tr>
<td>cylinder</td>
<td>6</td>
</tr>
<tr>
<td>cone</td>
<td>6</td>
</tr>
<tr>
<td>sphere</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 9.2: Probing Point Quantities for each Surface Representation

The *point* Surface Representation is only used by this Operation Type Planner for representing datum planes for geometrical tolerances such as parallelism and therefore uses the same number of points as a *plane*. The other Surface Representations use point quantities that are within the ranges specified by the Ferranti CMM programming language.

9.5.2 The Measuring Operation

The Measuring Operation describes the method that should be used to measure each dimensional and geometrical constraint that requires inspection on the component. The choice of the Measurement Method depends on the type of the constraint and the types of the constrained surfaces, and it is represented within the Inspection Planner as a data structure that describes the Measurement Method, the Probing Operations, the Working Plane and a pointer to the parent Relationship (figures 8.6 and 8.7).

The most important decision that faces the Operation Type Planner in this area is the choice of the Measurement Method and this is achieved by use of a rule-base that is configured to specify either two or three-dimensional inspection methods for geometrical tolerances and two-dimensional inspection methods for dimensional tolerances. The choice of Measurement Method for each geometrical tolerance is tightly
constrained but it is possible to specify a variety of inspection methods for dimensional tolerances as illustrated by the example in figure 9.2a. The one-dimensional measurement method is the simplest approach that could be used for the measurement of this surface and would require a single point to be taken from each surface (figure 9.2b) so that the CMM controller could calculate the distance between the two points in the X axis of the component Axis System. This would determine the distance between the two faces at those points but would not be able to accommodate any errors of form or attitude that might be present on either surface. An alternative two-dimensional method would represent each surface as a line by taking two or more points on each surface as shown in figure 9.3a and by calculating the distance between the lines in the Z plane. This approach has the advantage that it at least considers the form and attitude of the surfaces in one dimension. The most sophisticated approach is the three-dimensional measurement method which represents each surface as a plane by taking at least three points on each surface as shown in figure 9.3b and by calculating the distance between the centre points of the Surface Representations in the Z plane. This approach considers the form and attitude of each surface in two dimensions but has the disadvantage that it requires the most points and therefore the longest cycle time.

The CMM manufacturers [155,146] and the standards institutions [19,20,21,22] provide no formalised guide-lines for the dimensionality of the Measurement Method that should be used, leaving the decision to either the individual inspector or company standards. The Operation Type Planner implemented for this research is restricted to the two-dimensional method which is taught at the Ferranti CMM training course and therefore approved by the company.

The Operation Type Planner selects the Measurement Methods by using a set of rules to analyse each Relationship of the Relationship Graph in order to retrieve the
Chapter 9 - Operation Type Planning

Relationship Type (figure 8.9) and the constrained surfaces as illustrated by the following examples.

- Figure 9.4a shows a planar surface constrained by the geometrical tolerance of flatness which is described in the Relationship Graph (figure 8.10) by a tolerance and a pointer to a planar surface in the CSG tree of the component. There is only one Measurement Method for flatness (figure 8.6) and so the Operation Type Planner instantiates a Measurement Method of plane_flatness which is described by the type 3d_form. This type only requires a single Probing Operation which uses the plane Surface Representation.

- Figure 9.4b shows two cylindrical surfaces constrained by the geometrical tolerance of squareness which is described in the Relationship Graph (figure 8.10) by a datum surface, a test surface and a tolerance. The Operation Type Planner finds that there are two Measurement Methods for squareness; axes_squareness and planes_squareness (figure 8.6) but selects axes_squareness because the tolerance is constraining cylindrical surfaces. The Measurement Method of axes_squareness is described by the type attitude which requires a Datum Probing Operation and a Test Probing Operation so the Operation Type Planner instantiates two Probing Operations with cylinder Surface Representations and provides pointers to the appropriate datum and test surfaces.

- The final example (figure 9.4c) shows a linear dimension between a cylindrical surface and a planar surface which is described in the Relationship Graph (figure 8.10) by the dimension axis, a nominal, plus and minus tolerances and pointers to the constrained surfaces. The Operation Type Planner finds that there are four Measurement Methods for the Relationship type of linear_dimension (figure 8.6) but selects circle_line or line_circle depending on the order of the surface pointers.
under \textit{dimension} (figure 8.10). Both of these Measurement Methods have the type \textit{linear dimension} which requires a Working Plane and two Probing Operations. The Probing Operations are instantiated with \textit{line} and \textit{circle} Surface Representations and the respective surface pointers and the Working Plane is made the same as the Working Plane of the \textit{line} Probing Operation.

9.5.3 The Axis System Setting Operations

An Axis System is constructed for a part so that one and two dimensional constraints can be measured and also to identify the true 3D orientation of the component.

A part has a major Axis System which represents the general orientation of the part and may well have a number of minor Axis Systems for constraints that cannot be measured against the major Axis System. Each Relationship, therefore belongs to a Reference Frame (figure 8.9) and is measured with respect to the equivalent Axis System which is set up on the CMM.

An Axis System is commonly established by the following sequence of events. A surface is probed and represented by a 3D Surface Representation, such as a \textit{plane}, \textit{cylinder} or \textit{cone}, and the primary axis of the Axis System is aligned along the operating axis of the Surface Representation. A secondary axis, orthogonal to the primary axis, is constructed by probing one or more surfaces lying on an orthogonal plane and by aligning the secondary axis along a 2D representation of the probed surfaces. The orientation of the third and final axis is constructed automatically as it is mutually orthogonal to the primary and secondary axes.

The most difficult decision to be made when planning this activity is the choice of surfaces to be probed to construct the Axis System. This decision is critical as the
actual orientations of the surfaces used to construct the Axis System must be representative of the orientation of the component as otherwise measurements would be inspected against a non-representative Axis System and would therefore be susceptible to error. This problem does not have an absolute solution as it is only possible to identify the ideal surfaces for constructing an Axis System once the entire part has been measured. Therefore the most practical approach is to predict the surfaces that are likely to be most representative.

In certain circumstances the decision will have been made by the designer if any geometrical tolerances of attitude, such as parallelism, squareness or angularity have been specified. By definition these constraints incorporate a datum surface which has true attitude, and therefore these surfaces are ideal candidates for setting up an Axis System. However geometrical tolerances of attitude are not always specified and some other guide-lines for surface acceptability must be followed. One such set of guide-lines might be based on the tolerances attached to linear, radial and angular dimensions, as these also constrain, by implication, the attitude of the surface. One hypothetical approach might use what is termed the Mean Tolerance Ratio, shown below, which provides an indication of the expected positional accuracy of the surface as it calculates the average tolerance range of all the dimensions that constrain it.

\[
MTR = \frac{1}{n} \sum_{i=1}^{n} \frac{(U_i - L_i)}{N_i}
\]

where,

- \(MTR\) is the Mean Tolerance Ratio
- \(U_i\) is the upper tolerance value for constraint \(i\)
- \(L_i\) is the lower tolerance value for constraint \(i\)
\( N_i \) is the nominal value for constraint \( i \)

\( n \) is the number of constraints

The tolerance range is represented as a ratio against its nominal as this allows angular dimensions as well as linear and radial dimensions to be incorporated in the analysis and it also takes account of the fact that the greater the nominal, the more difficult it is to maintain a close tolerance. Therefore if the Mean Tolerance Ratio is calculated for every surface, the surface with the lowest value should represent a good candidate for setting up an Axis System.

This approach not been tested as the Operation Type Planner used for the most recent case study (the basis of the Users Guide described in Appendix D) uses a simpler method which counts the number of tolerances attached to each surface and selects the ones with the greater number of tolerances. However it is interesting to note that Park and Mitchell [106] use a similar approach to the Mean Tolerance Ratio for determining the constraints that should be measured in each component orientation.

9.5.4 The Datum Setting Operations

The Datum in the context of this Inspection Planner is the origin of the component Axis System and its sole function is to locate the actual position of the part on the CMM table. The Datum is commonly established by probing mutually orthogonal surfaces and by setting the Datum Point at the intersection of their operating axes. This Operation Type Planner allows two approaches to be used depending on the surfaces that are probed. If the principal surface is cylindrical then an orthogonal planar surface is used to set the remaining axis otherwise the datum is set using three orthogonal planar surfaces.
9.6 The Results

The Operation Type Planner has been tested by both the Micro-Switch Cover case-study (appendix C) and the Wrist Motor Body case-study (appendix D) with respect to the generation of a range of constraints including linear dimensions, radial dimensions, concentricity and true position. The capability of the Operation Type Planner to generate Axis System and Datum Setting Operations was tested by the Wrist Motor Body case-study. Another case-study based on a structural bearing and supplied by the Glacier Metal Company (figure 8.3a) was successfully used to test the capability to generate Measuring Operations for the flatness, parallelism and squareness geometrical constraints.

9.7 Discussion

The results show that the approach adopted by the Operation Type Planner is effective and can accommodate a wide range of dimensional and geometrical tolerances. The Planner has been configured to generate operations for all dimensional and geometrical tolerances described in BS308 [19,20,21] excluding the constraint of run-out which cannot be measured on a Coordinate Measuring Machine using a Touch Trigger Probe. Rules to generate operations for linear dimensions that constrain intersections of surfaces such as vertices or edges have not been implemented but this could be achieved using an extension of the method used for conventional linear dimensions.

Axis System and Datum Setting Operations have been planned successfully for the Wrist Motor Body and the method for selecting the principal surface for the Axis System and Datum was proved to operate effectively as it chose the surfaces that an inspector might have chosen intuitively. It is recognised, however, that this area could benefit from further research to determine the criteria for selecting surfaces for Axis
System and Datum Setting Operations.

The research elsewhere is well established in this field with several Inspection Planners using similar rule-based approaches to determine Measuring and Probing Operations although little detail is provided regarding the actual rules used and the effectiveness of the approach. Park and Mitchell are alone in considering the problem of part orientation for vision systems which is equivalent to Axis System setting for CMMs. Other researchers also agree that the effect of sample point densities and measurement methods are not well understood, which vindicates the decision to restrict this Operation Type Planning research to fixed point densities and two-dimensional measurement methods. Future work in this area will need to address these issues and it appears that an artificial intelligence approach might have some benefit as the solution is dependent on many interrelated factors including the type of constraint, the tolerance, the manufacturing method and the accuracy and repeatability of the measuring device. It is worth noting that the Product Modelling System researched by this project (refer to section 6.5.1) would prove an ideal vehicle for this type of approach as it is purposely designed to store and supply this type of detailed information to manufacturing applications.
Chapter 10

Probe Approach Direction Generation

10.1 Introduction

The greatest breakthrough that has been made in the research described in this thesis is the automatic generation of Probe Approach Direction information, which facilitates the automation of several down-stream inspection planning activities. This chapter describes the evolution of the Probe Approach Direction Generator and comments on the benefits that it confers upon the Inspection Planning process.

10.2 The Objective

One of the principal objectives of the research described in this thesis is to automate the entire Inspection Plan and Code Generation process (refer to section 5.1) and in an effort to satisfy this objective for down-stream planning activities such as Component Setup Planning (refer to Chapter 11) it is realised that Probe Approach Direction information must be generated automatically.

Each surface on a component can be probed from a number of approach directions but not all of these directions will allow the probe to obtain the most accurate representation of the surface. This can only be achieved if the probe has access to a maximum area of the surface and this may not be possible from every approach direction.

The objective of this research is therefore to determine the approach directions that will permit the probe to obtain the most accurate representation of each surface without colliding with the component.
10.3 Research Elsewhere

Very little research has been carried out in this area as its importance to the inspection planning process is only just beginning to be recognised. However, the work of Kawabe, Kimura and Sata and Spyridi and Requicha are both relevant to PAD generation and this section highlights their input to this field, which, incidentally, is described in more detail in sections 4.2 and 4.14 respectively.

Ironically the work of Kawabe, Kimura and Sata [96] was the earliest documented research in inspection planning (1980) and yet they were sufficiently perceptive to understand the value of a solid model representation of the component to the inspection planning process. They were not able to determine the principal Probe Approach Directions (known as Probe Access Directions in their work) but they did use a similar technique to the first method described later in this chapter to verify that Probing Paths were collision-free. This technique consisted of the construction of a volume representing the movement of the stylus of the probe and then verifying whether this solid intersects with the component. If it was found to intersect then the Probing Path was adjusted and the intersection analysis repeated. From the observations of the research described in this chapter this approach must have required a relatively lengthy run-time, although, in fairness, it does pre-date the competitive research of Corrigall and Bell [156,157] and Spyridi and Requicha [115,124] by some nine years. An additional short-coming of this approach is that it assumes that the probe stylus is the only part of the probe that is likely to collide with the object, although it is recognised that this assumption has been made to reduce the computational load on the analysis.

The recent work of Spyridi and Requicha [124] tackles exactly the same problem as the research described in this chapter albeit from a different angle. Their strategy allows all possible probe approach directions (known as accessible directions in their
work) to be computed for every point of every surface and as such it represents a complete and very thorough solution to this problem. However the authors admit that this approach suffers from two quite serious shortcomings inasmuch as the accessible directions are computed using a half-line abstraction of the probe and a surface is deemed unprobeable if it cannot be measured from a single accessible direction. The first problem is important because although a half-line abstraction greatly simplifies the computation of the Local and Global Accessibility Cones used in their work, it can only prove that a surface is unprobeable and not the opposite, which is the requirement of the exercise. The second problem is serious because it could result in many surfaces on a typical part being rejected, but it can supposedly be remedied by segmenting the surface and re-analysing, although it has not been explained how this should be done.

10.4 The Underlying Principle

The underlying principle of the approach described in this chapter is that the movements of the probe can be modelled geometrically in order to determine the surface coverage that a particular Probe Approach Direction confers. However this is only possible because the movements of a probe when inspecting a surface are simple and consistent and can therefore be modelled relatively easily.

When probing a given point on a surface the probe traverses towards what is termed the Offset Point, which is offset a short distance from the Probing Point on the surface normal (figure 10.1). The probe then traverses slowly towards the Probing Point along the surface normal until it triggers against the surface, whereupon the probe retracts to the Offset Point and exits along the approach direction (figure 10.2).†

† Note that if the probe is not triggered for any reason the sequence remains the same as the probe continues along the surface normal for a short distance (normally set as a default value on the controller) and then returns an error signal before continuing with the exit procedure.
Because this sequence of movements is simple and consistent it can be modelled economically and this model is known as the Probe Movement Envelope (PME) as shown in figure 10.3. The objective of the research, however, is to identify the Probe Approach Directions that allow access to the greatest part of the surface and therefore the notion of a PME must be extended to enclose the entire surface and this model is known as the Total Probe Movement Envelope (TPME) as shown in figure 10.4. The theory is, therefore, that if the TPME is not found to intersect with the component then the Probe Approach Direction is acceptable.

10.5 The Assumptions and Limitations

1. The underlying principle of this approach is based on the assumption that the component has been modelled using Constructive Solid Geometry (refer to section 2.2.4), which through its limited domain can only be used to represent 90% of engineering components (refer to section 2.2.5), but does facilitate the process of generating PADs. It is also assumed that the technique of spatial decomposition (refer to Appendix B) is used, which is in any case highly appropriate to the PAD generation problem, and is well supported by software tools and technical expertise on the research project.

2. On a more fundamental level, it is assumed that any surface can be probed in at least one of six approach directions (figure 10.5a), which are based on the component's axis system (figure 10.5b). This assumption is made out of necessity, because it would be impractical to analyse every possible PAD using the underlying principle described in the previous section as this would result in a run-time stretching to infinity. However it is not an unreasonable assumption as there are few surface features that could not be accessed by a PAD which was coaxial with one of the component axes. This assumption also implies that each
surface should only be probed from one direction which is normal inspection practice although it is technically possible to probe each point on a surface from different directions. This assumption simplifies the analysis but is recognized as an area of further research.

3. To obviate the need to model the motorized probe head and ultimately the CMM spindle, the underlying principle assumes that it is only necessary to model the movements of the stylus and probe body. This greatly simplifies the computation of the Probe Movement Envelope and can be tolerated by the planner because it is a simple procedure to extend the probe body if there is any risk of the motorized head colliding with the component.

4. Finally the use of a standard sphere type stylus is assumed in compliance with the limitations described in section 5.7.

10.6 The First Method

The underlying principle of this approach as described earlier in this chapter is that the movements of the probe can be modelled geometrically because they are simple and consistent. This initial attempt at satisfying the objective does exactly that, by constructing a simple CSG solid of the Total Probe Movement Envelope (TPME) so that it can be analysed for intersection with the object using the spatial decomposition method (refer to Appendix B).

Intersection analysis is achieved by decomposing both the object and the TPME and by compiling a list of boundary cells from the TPME. The centre-point of each boundary cell is determined and the equivalent cell at that centre-point in the object is determined. If that object cell is found to be either full or a boundary cell the TPME is reasoned to intersect with the object and is rejected.
Naturally the decomposition level will exert a significant influence on the accuracy and efficiency of this intersection analysis as the larger the cell size, the greater the chance of an erroneous intersection. However, it can be deduced that the optimum cell size should be equivalent to the size of the stylus ball, which is the smallest geometrical feature of the probe. If the cell size is larger than this, the TPME model would be larger than necessary, which would increase the risk of erroneous intersections with the object and would subsequently reduce the number of surfaces that could be probed. If the cell size was smaller than the stylus ball the TPME would be modelled from a greater number of cells without a corresponding increase in accuracy and would therefore require a significantly greater planning time as the SDSM's computational requirements increase approximately as a cube of the decomposition level. Inevitably there will be geometrical details on some components that are smaller than the stylus ball and these will also be over-modelled. However there is no advantage in modelling these details more accurately as they could not be probed by that size of stylus ball in any case.

The CSG representations of the TPMEs are constructed from a range of parametric models that are selected and parametrized according to the combination of surface type and PAD. Examples of the TPME models required for planar surface/PAD combinations are shown in figure 10.6 and those for cylindrical surface/PAD combinations are shown in figure 10.7. These examples represent the TPMEs that were actually implemented for experimentation and as such were simpler than the ideal and less approximate TPME geometry shown in figure 10.4. This is not seen as a serious problem as the computational overhead incurred in achieving this higher level of accuracy would be minimal.

More serious problems can be found, however, when the surface under analysis
does not have a regular boundary such as the planar surface shown in figure 10.8a. The parametric model designed for this surface/PAD combination (figure 10.6a) assumes that the surface under examination is a pure rectangle, because it would otherwise be difficult to construct a parametric model that was suitable for all possible surface boundaries. The solution implemented to circumvent this problem operates by determining the coordinates of a rectangle that would completely enclose the surface (figure 10.8b), so that this new extended regular surface could be analysed instead. However this solution has the inevitable disadvantage that it is forced to analyse a larger area, which might cause an erroneous intersection to be found.

This approach could theoretically be extended to accommodate the other types of surfaces represented by Constructive Solid Geometry (cones, spheres and tori) but the problem of irregular boundaries would be far more prevalent and difficult to solve as these types of surfaces are rarely employed in their original state.

10.7 Discussion of the First Method

This initial attempt to satisfy the objective was a logical step from the underlying principle, which operated by literally modelling the TPME and determining whether there was any intersection between this solid and the component geometry. The theory is sound, but in practice the system was found too inflexible to accommodate the complex surfaces generated by intersections and far too slow in operation to be of any real practical value. More importantly, it was also fundamentally incapable of quantifying the percentage of a surface that could be accessed when intersections were discovered as satisfactory algorithms for analysing the effect of these intersections could not be derived.

However it was the first attempt at solving the problem and it did at least determine a set of PADs for a component albeit only the ones with 100% surface access.
The slow operation of the system was largely caused by the fact that five computationally intensive spatial decompositions were required for each planar surface (two for cylindrical surfaces) and attempts were made to minimize this requirement by undertaking some simple checks on a surface before spatial decomposition took place.

The first check was carried out for planar surfaces with coplanar PADs and cylindrical surfaces with axial PADs and operated by comparing the length of the probe against the depth of the surface (figure 10.9). Obviously if the stylus was found to be shorter than the surface depth then the probe body would collide with the surface and the PAD would be rejected.

The other check tested for the common occurrence of an intersecting surface that obscures the approach direction as shown in figure 10.10a/b. In these circumstances a column of cells outside the surface and parallel to the PAD would be checked to verify that they were empty (figure 10.10c). If a cell was found to be either full or a boundary cell the PAD could be rejected.

It is worth noting that both of these checks are only appropriate because the system is incapable of quantifying surface coverage. A more effective approach would determine the proportion of the surface the probe could access either with the shorter probe or by avoiding the intersecting surface respectively.

10.8 The Second Method

The major problems with the initial method were the inability to quantify surface coverage and the extended run-time caused in large part by the number of costly spatial decompositions that were required. These unacceptable short-comings could not be resolved by further refinement of the approach and so a radically different solution was researched.
The second method was still based on the underlying principle of modelling the TPME but rejected the idea of using a CSG representation. This time the surface of the TPME was modelled by a matrix of imaginary cells equivalent in size and organisation to the decomposed boundary cells of the object. This was achieved in a series of stages. Firstly a list of all the boundary cells on the surface under analysis was compiled using a standard SDSM query routine. A PME was then built for each boundary cell in this list, which consisted of a list (known as the PME cell list) of all the centre points of the imaginary cells used to build the PME. As each PME was built a TPME cell list was compiled, by checking each cell in the PME cell list against the cells in the TPME cell list. If the PME cell did not exist in the TPME cell list then a new TPME cell was created and a pointer added from the TPME cell to the parent surface cell that caused the PME. If the PME cell was found to exist in the TPME cell list a new cell was not created but a pointer was added from the equivalent TPME cell to the parent surface cell that caused the PME. The result of this process, which is complicated to describe, but simple to implement, was a model of a TPME that consisted of a list of imaginary cells with pointers to parent surface cells. Each cell in this list could then be checked against the equivalent cell in the object using the method of intersection analysis described in the first method, and if intersection was detected, the parent surface cells of the TPME cells could be labelled as unprobeable. The final stage in the process was to calculate the percentage of probeable cells on the surface so that the accessibility of the surface could be quantified.

This method was implemented successfully, but only because an efficient method for generating the imaginary PME cells was determined. Going back to first principles, it can be seen that a volume similar to the one shown in figure 10.11b is required to represent the PME of the probe movements shown in figure 10.11a. For the purposes
Chapter 10 - Probe Approach Direction Generation

of the approach described in this section it is sufficient to represent this volume as a collection of surfaces (figure 10.12a), each one of which is represented as a matrix of cell centre-points (figure 10.12b). This is acceptable because each point coordinate represents the centre-point of a cubic cell equivalent to the decomposed cells of the object. Therefore if the imaginary cells are built into a PME (figure 10.12c), it can be seen that the resultant volume is similar to that of the required PME shown in figure 10.11b. Originally a rather crude method was used to generate the point matrices of each of the seven PME surfaces, which used trigonometry to calculate the coordinates of each surface vertex with respect to the object. This proved inefficient and difficult to code. A far more effective method based on vector theory was subsequently implemented as shown in the following example.

The objective of the exercise is to generate the point matrices required to describe the Probe Movement Envelope that the probe would occupy when probing a Probing Point \( A \) on a surface with normal vector \( N \) (where \( N = A\bar{B} \)) at a Probing Distance \( d \). The probe has a stylus length \( l_1 \), body length \( l_2 \) and body radius \( r \) and approaches the Probing Point from a PAD \( P \) as shown in figure 10.13a.

To avoid the generation of unnecessary points the stylus length \( l_1 \) and body length \( l_2 \) are calculated as follows. The stylus length is originally set to the length of the stylus fitted to the machine and the probe body is assumed to be sufficiently long to probe any surface on the part. However these values can be optimized by considering the relationship between the PME and the part bounding box supplied by the SDSM, which represents the outer limits of the part. A simple algorithm allows the point \( X \) to be calculated, which lies on the intersection between \( P \) and the bounding box of the object. Therefore:

\[
\text{if } XA \leq l_1 \text{ then } l_1 = XA \text{ and } l_2 = 0
\]
else $l_2 = XA - l_1$

From this information it is now possible to calculate every surface vertex of the PME (figure 10.14a) as shown below:

\[ B = A + dN \]
\[ C = B - l_1 P \]
\[ D = A - l_1 P \]

An additional unit vector $\hat{u}$ and points $C_1$ and $D_1$ are constructed as follows (figure 10.14b):

\[ C_1 = C + rN \]
\[ D_1 = D - rN \]
\[ \hat{u} = N \times P \]

which allows the remaining points to be calculated:

\[ E = D_1 + r\hat{u} \]
\[ F = D_1 - r\hat{u} \]
\[ G = F - l_2 P \]
\[ H = E - l_2 P \]
\[ I = C_1 + r\hat{u} \]
\[ J = C_1 - r\hat{u} \]
\[ K = J - l_2 P \]
\[ L = I - l_2 P \]

Finally it is a straight-forward affair to calculate, again using vector theory, the coordinates of the points that lie on each surface, so that they are separated by no more than a cell width equivalent to that of the object cells.

This approach is suitable for all surface types and PADs but can be simplified for the special case where $N = -P$ as shown in figure 10.15a/b.

10.9 Discussion of the Second Method

This second attempt to satisfy the objective of PAD generation is capable of determining the percentage surface access open to a particular PAD and is therefore
capable of determining the principal PAD for a surface and has subsequently satisfied the objective. However, like the approach of the first method, it also suffers from a long run-time, which is mainly caused by the unexpectedly large size of the TPME cell list. This list often runs to tens of thousands of elements for a typical surface and as each cell of each PME cell list has to be checked against each cell of the TPME cell list the time requirement for this operation can become excessive. The approach therefore received a final set of modifications to achieve a satisfactory solution.

10.10 Modifications to the Second Method

As the run-time problems had been traced to the size of the TPME cell list the first line of enquiry investigated the possibility of whether the activity could be simplified or circumvented. The latter proved more profitable and a system was implemented that did not require a TPME cell list as each cell was checked directly against the object as soon as it was generated. This must result in some points being checked more than once as the PMEs will inevitably overlap on any surface, but this penalty is more than compensated by savings made elsewhere. As well as obviating the need to manipulate the TPME cell list, the system takes advantage from the fact that it can halt the generation of PME points for a particular surface cell as soon as a PME point is found to intersect with the object. This allows situations such as the ones illustrated in figure 10.10 to be detected very quickly so that analysis can be redirected to one of the other PADs.

Continuing the quest to make the PAD generator as economical as possible, another set of changes were made, this time to take advantage of particular conditions that exist with the planar surface/coplanar PAD and the cylindrical surface/axial PAD combinations as shown in figure 10.16. In these special cases it is not necessary to generate a complete PME for every surface cell because the PMEs overlap in such a
way that it is possible to generate smaller and more efficient incremental PMEs as described by the following example. If a planar surface is probed from the -Z approach direction as shown in figure 10.17a, the more efficient analysis strategy would operate by analysing the surface cells in columns aligned with the PAD. The top cell in each column would then require the generation of a complete PME (figure 10.17b) but the subsequent cells in the column would only require an incremental PME (figure 10.17c) consisting of a single layer of cells representing the movement of the stylus ball and a single layer of cells representing the bottom of the probe body. Incremental PMEs would be generated for each subsequent cell until an unprobeable surface cell was detected. Obviously there is no need to investigate the column any further as any lower cells would also be unprobeable. Extending this line of thought even further it is also possible to deduce that if each of the cells in the top row are found to be unprobeable then it can be argued that all of the surface is unprobeable from that PAD so obviating the requirement to analyse the incremental PMEs. This strategy was implemented and again made further savings.

However as a completely speculative experiment, it was decided to implement the opposite strategy, known as the all good strategy (the first strategy was retrospectively termed all bad), to investigate what difference this might make. This time all the bottom cells were analysed with complete PMEs on the basis that if all of them were found to be probeable then the entire surface would also be probeable. If this was not the case then each column with an unprobeable cell at the bottom could be verified from the top down using incremental PMEs until an unprobeable cell was found.

Remarkably this unlikely strategy was again faster than the all bad method, and the only logical explanation could be the with respect to the numbers of particular surface intersections found on the component under analysis. If the component has
more 0% PADs than 100% PADs on planar or cylindrical surfaces with coplanar or axial PADs then the all bad strategy will be better, but if the opposite is true then the all good strategy will succeed.

Clearly whatever strategy is used, the general approach substantially reduces the number of cells that need to be generated and checked but it does have a slight additional overhead as the surface cells must be organised into ordered columns before analysis can commence.

10.11 The Results

The Wrist Motor Body case-study which forms the basis of the Users Guide described in Appendix D was carried out using the final version of the Inspection Plan and Code Generator which is based on the method described in section 10.10. The generation of the 122 PADs necessary to inspect this typical industrially-sourced component took exactly 20 minutes it was observed that all percentage surface coverages were calculated accurately (refer to sections D.4.1.3 and D.5.1.3). From this information, the PAD Generator was able to determine the optimum Probe Approach Directions for each surface, which is the objective of the exercise.

10.12 Final Discussion

These modifications to the second method represent the pinnacle of development of this technique and as such this section serves to discuss the strengths and weaknesses of the overall approach and to compare it with its competitors.

The accessibility cone method of Spyridi and Requicha aims to satisfy a similar objective to the one described in this chapter in that it computes the accessible directions of a probe to each surface on the part. In one crucial aspect, it is superior to the method described here in that it is capable of computing every possible accessible
direction for every point on the surface, which is clearly better than the restricted information available from the PME method. However this advantage is immediately lost because the method is only feasible if the simplest possible representation of a probe is used, which in this case is a half-line with its origin at the surface point that extends to infinity along the axis of the probe. The implication of this is that it is impossible to guarantee a collision-free access direction because any part of the ball, stylus or probe body that does not lie on the half-line can potentially collide with the object. In fact the approach can only guarantee those access directions, which will collide with the object. It is nevertheless a powerful approach that will undoubtedly be developed further with eventual success.

Returning to the PME method, it is clear that this approach can achieve its objective within acceptable performance criteria, which is established both through the case-study results and the benefits it confers on down-stream inspection planning activities.

However it is worth discussing the limitations that might be imposed by the assumptions stated towards the beginning of this chapter. The first assumption was in regard of the geometric environment within which the PAD generator would operate, which consisted of a primary CSG representation supported by an auxiliary spatial decomposition representation. These representations undoubtedly confer a less than complete geometrical domain, but the PAD generator is reliant on a solid representation, which CSG certainly provides. The geometry that CSG cannot describe consists largely of sculptured surfaces [13], which are in any case inspected by methods [99,101] that are quite different to the conventional Coordinate Measurement approach described here. As an aside, it is worth noting that research is currently being directed at Leeds University towards the possibility of spatially decomposing
sculptured surfaces [158]. If the outcome of this work is successful, then further research in this area would be warranted to extend the PME approach to include these more complex surfaces.

The second assumption was in regard of the sufficiency of only six possible Probe Approach Directions based on the component axis system of the object. The PAD generator was implemented on this basis for simplicity although it is appreciated that a certain class of geometrical configuration could prove unprobeable such as the one illustrated in figure 10.18. This class is typified by an internal geometrical feature that can only be probed along a single axis, such as a hole, but which is located on an additional feature that has been rotated with respect to the component axis system. The solution, although unimplemented and therefore untested, is quite straightforward in that the feature in question is analysed with respect to its local axis system rather than that of the component. This would require very little modification to the code of the PAD Generator but down-stream activities such as the Component Setup Planner (refer to Chapter 11) would need to be adapted to take this change into consideration.

The implication of the second assumption that surfaces should only be probed from one direction means that each point probed on the surface must have the same PAD, which is conventional practice on CMMs. However it is technically possible to probe every point from a different PAD on a CMM and it is theoretically possible to incorporate this extended functionality within the PAD Generator in the following manner. As a by-product of calculating the percentage surface coverage for each Probe Approach Direction, the PAD Generator also constructs a map of the probeable areas of the surface, principally for the purposes of the Probing Point Planner (refer to Chapter 13). If the principal PAD for a surface is less than 100% these maps could be used to identify additional PADs whose maps do cover the areas that are not covered
by the principal PAD. This would entail very little additional analysis and would result in a greater proportion of 100% surface coverages, albeit using combined PADs.

The third assumption was concerned with the fact that the PAD generator should only need to model the stylus and the probe body in order to verify each Probe Approach Direction. This greatly simplifies the PAD generation process and can be tolerated because the probe body can be extended if there is any risk of the motorized head colliding with the component. This is achieved by the method described in section 10.8, which calculates the probe body length \( l_2 \) by calculating the distance between the Probing Point and the bounding box of the component and subtracting the stylus length \( l_1 \). The resulting value \( l_2 \) is subsequently passed to the Probe Setup Planner described in section 12.5, which selects an appropriate probe body extension bar.

The final limitation of the approach was the assumption that a sphere type stylus should be used for all Probe Approach Direction analysis as the CMM used for all practical experimentation was not equipped with an automatic probe changing unit. However it would be possible to extend the approach described in this chapter to accommodate all the various stylus types, which include disc, star, cylinder and point (refer to section 12.5.1) simply by changing the shape of the Probe Movement Envelope.
Chapter 11

Component Setup Planning

11.1 Introduction

Component Setup Planning is one aspect of Inspection Planning that has received very little attention from the research and commercial communities, as most planning systems assume that the inspector will be able to determine the optimum set of component orientations and their fixturing requirements intuitively. Conversely, this chapter proposes a method that automates the Component Setup Planning process, which is principally achieved through the use of information generated by the Probe Approach Direction Generator (refer to Chapter 10).

11.2 The Objectives

One of the principal goals of the research described in this thesis is to automate the entire Inspection Plan and Code Generation process (refer to section 5.1) and one key aspect of this process is the planning of Component Setups.

The primary objective of Component Setup Planning is to determine a minimum set of component orientations that will permit every surface on the component that requires measurement to be inspected from its principal Probe Approach Direction. Therefore each Component Setup must specify the orientation, the fixturing method and the list of associated Measuring Operations. The Component Setup would also normally include probe configurations and orientations but for reasons of simplicity the planning of this information is considered separately in Chapter 12.

The Component Setup Planner places emphasis on the minimum number of component orientations as the time taken to reorient a component is considerable in
comparison with the typical inspection cycle time. This is because the component will need to be unloaded from the machine, re-fixtured, re-aligned and re-datumed by the CMM part program. Rotary tables that obviate the need to reorient the part manually are finding an increasing number of applications on CMMs, but as yet they have not become a part of the standard CMM configuration, like for example, motorized probe heads. For this reason, a Component Setup Planner should not assume the use of a rotary table but it should be possible for the theory to be extended to accommodate their use.

A secondary objective and consideration for a Component Setup Planner should be the ease with which each component orientation can be fixture. Clearly there is no value in planning the component orientations only to find that they are difficult to fixture and so it is important to give this factor consideration during the planning stage.

The problem therefore resolves itself into one of establishing a compromise between the primary objective of selecting the minimum number of Setups and the secondary objective of ensuring that a satisfactory fixture can be designed easily for each component orientation.

11.3 Research Elsewhere

As can be seen from the review of inspection planning literature in chapter 4, relatively few research programmes are investigating the problem of component orientation and fixture design in the inspection planning environment.

The work of Park and Mitchell [106] arguably represents the most comprehensive effort, which is based on the planning of view directions for a vision system. These view directions are determined by an upstream process planning system that
analyses a B-rep representation of the workpiece to determine tool approach directions, feed directions and view directions in support of both the machining and inspection functions. For each view direction the inspection planner calculates the value $A_k$, which is given by the following equation:

$$A_k = \sum_{i \in F_k} \sum_{j \in T_i} \frac{1}{\alpha_i \beta_j \text{tolerance}_j}$$

where,

- $F_k$ is the index of features in orientation $k$
- $T_i$ is the index of tolerances in feature $i$
- $\alpha_i$ is the weighting factor for each design feature
- $\beta_j$ is the weighting factor for each tolerance

This value represents the overall tightness of the tolerances that can be measured from a specified view direction and is used to indicate the relative importance of that view direction to the inspection cycle. The view directions can then be sequenced so that the direction with the greatest $A_k$ is carried out first and so on until all tolerance measurements have been allocated to a view direction.

Although this approach has been implemented for a vision-based inspection system, the problem is similar to the one faced by an inspection planner for CMMs. However, the solution described here does not extend as far as attempting to determine the absolute minimum number of component orientations required to inspect the part but reasons that a near minimum number of component orientations can be determined if constraints are allocated to a ranked list of view directions. This approach does not consider the fixturing requirements of each view direction, although it could be argued that this is not a serious omission as the fixturing requirements for
vision are even less stringent than they are for CMM inspection.

The method of accessibility analysis described by Spyridi and Requicha [115,124] and reviewed in section 4.14 is reasoned to be highly appropriate for the planning of component orientations or Setups but as yet this work has not been published.

The research of Tang and Davies [123] has been directed towards identifying the types of knowledge representations that are best suited to inspection planning tasks and they have demonstrated their results by implementing a rule-based planning system for determining how turned components should be held during the inspection cycle. The rules that they have implemented are strictly limited to the case of turned parts and a new rule-base would need to be employed if a wider range of parts were to be considered. However even for this limited domain of parts it is possible to see the power of this type of approach in spite of the necessarily trivial examples cited by the publication.

In comparison with the lack of research into Component Setup Planning for inspection, there are many research programmes considering the similar problem in machining, as shown by the following brief examples.

Bell and Young [138] are part of the same research project as the research described in this thesis, and they describe an approach for Setup planning in machining, which uses a selection of rules to select the most appropriate of three fixturing methods, including side clamping, through clamping and positive surface clamping.

Englert and Wright [159] describe an expert system for part Setup and workholding in a small batch environment that attempts to combine quantitative physics-based analyses with decisions based qualitatively on experience, as they observe that this is the approach used by skilled craftsmen in industry.
Chan and Voelcker [72] have been researching an approach based on a high-level language known as MPL, which supports commands for positioning, clamping and unclamping a workpiece. The authors admit that this is only an initial study of the problem but discuss the possibility of supporting this environment with analysis tools such as a Setup Validator that computes whether the position and orientation of the workpiece are completely fixed by the specified Setup operations.

Boerma and Kals [76] are investigating the related problems of automatic Setup generation and fixture design and propose a computer-aided system, known as FIXES, which automatically determines how the part should be positioned and clamped using various rules and algorithms.

Finally Lim and Knight [77,160] describe an Intelligent Knowledge Based System called HOLDEX, which makes rational decisions similar to those of an experienced tool engineer to enable the automatic design of fixtures.

11.4 The Assumptions and Limitations

The first of the assumptions that are made by this approach is that any component can be oriented and satisfactorily inspected in any one of six possible orientations based on the component axis system as illustrated in figure 11.1. This assumption has the beneficial effect that it reduces the number of possible orientations from infinity to six and therefore brings the analysis within more manageable proportions. The only possible advantage that one component orientation could have over another is the increased accessibility that might be on offer to the probe. However as the Probe Setup Planner, described in chapter 12, assumes the use of a motorized head (figure 5.3) this advantage is of negligible benefit because the probe can adopt almost any orientation below 75° to the upper vertical.
The second assumption of this approach is the use of simple fixturing methods, which allow the part to be secured on only the lower surfaces of the part that cannot be accessed by the probe (figure 11.2). This has the substantial advantage that the Component Setup Planner can effectively ignore any fixturing considerations such as how the fixture should be designed and whether it will obscure any surfaces that require measurement. This assumption is listed as one of the constraints imposed upon the project in section 5.7 and is discussed in more detail in section 11.6 of this chapter.

11.5 The Method

The simplicity of this approach is due in no small part to the Probe Approach Direction (PAD) information that has already been determined for each surface that requires measurement. The PAD generator, as described in chapter 10, computed for each of six possible approach directions the percentage area of a surface that could be accessed by the probe without collision. From this information the principal PADs can be selected for each surface on the basis that the more access the probe has to a surface the more representative the measurement will be. The availability of this information therefore reduces the problem faced by the Component Setup Planner to one of determining the minimum set of component orientations that will provide the probe access to each surface from its principal Probe Approach Direction, as described by the following sequence of events:

1. For each component orientation, determine whether the axis system and datum setting operations can be carried out satisfactorily. The axis system and datum setting operations have to be carried out in each component Setup in order to construct the component axis system and the datum point, which are prerequisite to the inspection process. These operations are carried out by probing a number
of surfaces determined during the Operation Type Planning phase (refer to sections 9.5.3 and 9.5.4) and these surfaces can only be probed if their principal Probe Approach Directions are not in the +Z direction with respect to the CMM. If any one of these surfaces has a +Z principal PAD, the probe will not be able to gain access due to the limitations of the motorized head and the component orientation must be invalidated and removed from the list of possible component orientations.

2. For each valid component orientation, count the number of constraints that can be measured satisfactorily. Each constraint that requires measurement on the part is analysed to determine whether it can be measured from the current component orientation. If any one of the constrained surface has a +Z principal PAD the probe will not be able to gain access due to the limitations of the motorized head and the constraint cannot be measured in that component orientation.

3. Select the component orientation which allows the most constraints to be measured. This activity identifies the component orientation that allows the most constraints to be measured and assigns this orientation as a component Setup (refer to figure 8.4). The constraints that can be measured in that orientation are then allocated to that component Setup and are removed from the list of constraints to be allocated. A Safe Rotate Height (refer to figure 8.4) is calculated for every Setup, which represents the height at which the motorised probe head can reorient itself without risk of colliding with the component. This height is calculated with respect to the component datum and is derived by combining the height of the component in the Setup orientation to the length of the stylus, probe and extension bars specified by the Probe Setup Planner (refer to Chapter 12).
4. If there are any remaining constraints to be allocated, return to the third activity. It is highly probable that more than one Setup will be required to inspect all the constraints on a component and so this activity permits additional Setups to be assigned. The third activity will need to re-count the constraints that can be measured by each component orientation because it is quite likely that the constraints that have already been allocated were probeable in more than one component orientation and will now have been deleted from the list of constraints that need to be allocated.

The result of this process is a near minimum set of component orientations, each with a list of constraints that have been validated for measurement in that orientation.

This process was implemented and used in a number of experiments during the project, including the case studies described in appendices C and D. During these experiments it was occasionally found that the third activity in the Component Setup Planning process would determine that there was more than one component orientation which allowed the most constraints to be measured. Initially it was decided to let the planner make an arbitrary choice, but after some consideration it was found that there might be some advantage in making the planner select the component orientation that was most easily fixtured. Clearly such a quality is difficult to quantify without access to a fixture design knowledge base and so the planner was redesigned to compute the relative stability of the component, which is considered to be an important factor in the design of simple fixtures.

The relative stability is computed by calculating the centre of gravity of the component and determining its vertical height above the fixturing plane. This is easily computed from the spatially decomposed model using the following equation:
where,

\[ C_a = \frac{\sum_{i=1}^{n} (V_i d_{(i,a)})}{\sum_{i=1}^{n} V_i} \]

\( C_a \) is the centre of gravity in axis \( a \)
\( V_i \) is the volume of full cell \( i \)
\( d_{(i,a)} \) is the distance of cell \( i \) from the origin in axis \( a \)
\( n \) is the total number of the full cells

This generates an approximate value for the centre of gravity, which becomes more accurate as the decomposition level increases, but for the purposes of this activity the accuracy at the decomposition level required for inspection planning is more than sufficient.

A more thorough analysis of relative stability might calculate the minimum force required to topple the object, but the approach described above assumes that simple fixturing methods have been used to support the object under its lower surfaces. Figure 11.3a shows a component in a relatively stable Setup orientation with respect to the vertical height of the centre of gravity, but it would of course topple over if unsupported (figure 11.3b). Figure 11.3c shows the use of a simple mounting block to maintain the stability of the component in the desired orientation.

### 11.6 Fixture Design

The method of Component Setup Planning described in the previous section assumes that simple fixturing methods will be employed that both support and secure the component in the specified orientation using its lower surfaces and without interfering with probe accessibility to other surfaces of the part. This greatly simplifies
the Component Setup Planning process because it obviates the need both to design a fixture during the inspection planning process and to verify that surfaces remain probeable after the component has been assembled with the fixture. Fortunately there are many inspection fixturing methods that satisfy the above requirements and therefore substantiate the assumption. These include:

1. quick-setting adhesives that allow the component to be snapped off once the inspection cycle is completed

2. blutak or plasticine to mount components that will not be disturbed by the impact of the probe (7 to 8 grammes) but need deformable mounting pads to avoid seesawing movements

3. switchable magnetic beds for iron-based materials

4. suction beds for air-tight components

5. threaded pedestals for components with threaded features on their lower surfaces

6. internal clamps for components with pockets or holes on their lower surfaces

Each one of these fixturing methods is cheap, flexible, easy to use and satisfies all of the requirements of an inspection fixture, which are:

1. to prevent any movement of the component during the inspection cycle

2. to provide good accessibility to the component

3. not to deform or damage the component (some polymer-based materials and special surface finishes may not be compatible with adhesives)

However in certain circumstances, none of the fixturing methods described above will be appropriate and more complex methods will need to be used. These include:

1. top clamping
Chapter 11 - Component Setup Planning

2. side clamping

3. through-hole clamping

These methods corrupt the assumptions made by the Component Setup Planning activity as they obscure top and/or side surfaces and will therefore corrupt the PADs that have been previously defined. However it is theoretically possible to circumvent these problems by designing the fixture during the Component Setup Planning process in the following manner. Once a component orientation has been selected for a Setup an appropriate fixture could be manually designed by choosing and assembling modular fixturing elements from a CSG data-base taking care not to interfere with any surfaces of the part that require measurement. Once the component and fixture have been assembled the principal PAD of each surface that requires inspection in that component orientation could be re-analysed to ensure that the PAD is still valid. Inevitably there will be some surfaces that have reduced access through the addition of the fixture and in most circumstances this could be accepted providing the reduction was not too dramatic. However if a surface was completely, or mostly, obscured then it would have to be reallocated to another more suitable component Setup.

11.7 The Results

The Wrist Motor Body case-study which was the basis of the Users Guide described in Appendix D was carried out using the final version of the Inspection Plan and Code Generator which incorporates a Component Setup Planner based on the ideas presented in this chapter.

The centre of gravity was accurately computed for the component using the method described in section 11.5 and this information was used to assist the selection of component orientations for two Setups. Simple fixturing methods were successfully
employed and the Measuring Operations associated with each Setup were observed to operate correctly.

11.8 Discussion

This chapter describes a simple method based on the analysis of principal Probe Approach Directions that allows a near minimum set of component Setups to be determined for the total inspection of a part. Each Setup is comprised of a component orientation and a validated list of constraints to be measured. In its present form this approach is not capable of designing a suitable fixture as this is argued to be beyond the scope of the project resources (refer to section 5.7). However the planner assumes the use of simple fixturing methods listed in section 11.6 and this is argued to be a reasonable assumption in the majority of inspection applications. In those situations where simple fixturing methods are not feasible, it is theoretically possible to manually design a fixture using modular elements from a CSG data-base, which would permit the design of the fixture to be validated by analysing its interaction with the principal PADs.

The overall approach described in this chapter is similar in many respects to the method adopted by Chang, Anderson and Mitchell, which also determines a near minimum set of component orientations albeit for the vision inspection application. Their approach is slightly different in that it chooses the best component orientation by computing the overall tightness of the tolerances that can be inspected from each orientation, whereas the method proposed here chooses the orientation that allows the most constraints to be inspected. Both approaches have the same effect of enabling a near minimum set of component orientations to be determined but it is not clear whether either method has any advantage over the other.
Whilst on the subject of the near minimum set, it is quite probable that there are statistical methods that allow the computation of the absolute minimum number of component orientations, but as yet none have been discovered that are totally satisfactory. However, for the purposes of this exercise, the determination of the near minimum set is sufficient, although it is expected that this problem will receive some attention from the statistics community as soon as it becomes more widely recognized.

The secondary objective of identifying and selecting those component orientations that are most easily fixtured has only been partially achieved by the approach described in this chapter. The planner determines the fixturability of a Setup if more than one component orientation has been found to have the most constraints, when the fixturability criterion can be used to choose between them.

Ideally a Component Setup Planner would be able to combine the objectives and determine the minimum set of component orientations that could most easily be fixtured, but this is a more complicated problem than it first appears. Assuming that the relative stability method was an acceptable measure of fixturability, the planner would still have the difficulty of selecting the most appropriate criterion to plan by, as fixturability and the minimum set of orientations are not complementary objectives.

One possible solution could be to determine the likelihood of fixturability problems by classifying the component as either block-like, plate-like or rod-like as shown in figure 11.4. This could be done by computing the centre of gravity and its vertical height in each orientation. If one of the vertical heights was, for example, 20% greater than the average then the component is rod-like, whereas if one of the vertical heights was 20% less than the average then the component is plate-like and finally if there is no real difference between the three heights the component is block-like. Clearly the plate-like and rod-like components are going to be less stable and hence more difficult
to fixture in certain axes than in others and if this situation could be detected the planner could operate according to the fixturability criterion. This criterion would force the planner to analyse each constraint and allocate it to the most stable component orientation in which it could be measured. This would ensure that the most easily fixtured component orientations were selected but would not necessarily guarantee a minimum set.

Organising entities into sets using multiple and conflicting criteria is not easily achieved in the conventional data processing environment adopted by this research project and it is suggested that there would be a greater chance of success if artificial intelligence techniques were employed. Unfortunately, this was not a viable proposition under the constraints of this project and instead an effective but non-optimum solution has been researched.
Chapter 12

Probe Setup Planning

12.1 Introduction

When CMMs first appeared in industry they were equipped with a single rigid probe [161,162] but as the technology evolved so did the range of both probes and probe associated equipment. The purpose of Probe Setup Planning is therefore to determine the types of probes that should be used and the orientation that the probe should assume to inspect each surface.

12.2 The Objectives

In parallel with the adoption of automatic tool changers on NC machine tools, automatic probe changers [117,163] are beginning to be used on Coordinate Measuring Machines (CMMs), which typically allow up to eight probes to be held on a rack and automatically exchanged with the operating probe during the inspection program. However, unlike the cutting tools used on NC machining centres, all touch trigger probes on CMMs are capable of performing the same basic function and are principally changed for reasons of surface access. The probe can be configured to suit a particular surface by adding extension bars, using different length styli or ball radii, or by using a different type of stylus tip.

An alternative means of attaining increased surface access is to employ the use of the motorized probe head that allows the probe to assume almost any orientation within an envelope the shape of a truncated sphere. This device allows the probe to approach the surface from the most advantageous angle and also has the effect of reducing the number of Setups required.
Chapter 12 - Probe Setup Planning

The objectives of the Probe Setup Planner are therefore to identify the most appropriate probe configuration and orientation to measure every surface that requires inspection. This involves the consideration of the probes and auxiliary equipment that are available on the machine and the selection of the most appropriate probe assembly on the basis of surface accessibility. The probe orientation should be selected so that it affords the maximum access to the surface as this will ensure that the representation constructed from the points retrieved from the surface will be as accurate as possible.

12.3 Research Elsewhere

Like Component Setup Planning (refer to section 11.3) this is one of the more peripheral but nonetheless important areas of inspection planning where very little research has so far been carried out.

ElMaraghy and Gu [102] consider probe orientation and selection planning to be an integral part of their experimental inspection planner, which uses a feature-based geometrical representation and is currently restricted to turned parts. The probe selection part, like the rest of the inspection planning process is based on a set of if-then rules implemented in PROLOG, which list a number of conditions that must be satisfied before a particular probe can be used. These conditions might include the type of feature to be inspected, the size of the probe in relation to the size of the feature and the type of constraint to be inspected. Probe orientation is determined by computing the accessibility of each feature on the part and, although this detail is not explained, it is assumed that a different probe orientation is adopted if the feature is found to be inaccessible.

This is the only published research in this field and therefore represents the first attempt to validate the choice and orientation of the probe used to inspect a surface. However, this approach is restricted to the analysis of turned parts and although it is
claimed to be equally effective for prismatic parts, this must be doubtful without detailed evidence. This is because the rotational problem is essentially two-dimensional and therefore represents a substantially less complex environment for solving, for example, the feature accessibility problem. In a rotational environment, accessibility can be determined without excessive difficulty because the majority of inspection tasks are performed with the probe parallel to the centre-line axis, which consequently reduces the problem to checking the order of the features along the axis and their maximum radii. With prismatic parts, it is difficult to imagine how a similar approach could be adopted with the same degree of effectiveness.

As with Component Setup Planning, Spyridi and Requicha [115,124] also list Probe Setup Planning as a task that can be solved using their accessibility analysis approach. However no work has yet been published in this area.

By looking to the parallel research area of operation planning for machining, a growing number of research programmes can be found that are investigating the problem of tool selection. Clearly the cutting tools on machining and turning centres cannot be reoriented, unless five-axis machining centres are considered, but the issues involved in tool and probe selection have some similarity.

Bell and Young [138], who contribute to the same research project as the author, albeit in the area of machining, propose a tool selection methodology that determines the requirements and constraints of the cutting conditions of each operation so that they can be matched to a suitable tool from the manufacturing data-base.

Hannam and Plummer [164,165] employ a feature-based approach to integrate design and manufacture for the turned parts domain. Tool type information is associated with every feature in the so-called Shapes data-base so that when a feature is used in a design, the tooling information is automatically inherited.
Hannam continued this line of investigation with Lawlor-Wright [5], where a much more detailed analysis of tooling requirements was carried out, by establishing a method whose objective was to maximize the use of existing resources such as machines and tooling. They argue that tooling is a significant resource, because the tooling that is available is representative of the machines that are available, and if designers can be encouraged to use resources that are on-hand, then lead-times can be minimized.

Bard and Feo [74] have researched a cutter path generator for machining that optimizes tool utilization using the *Langrangian relaxation scheme* and a *greedy set covering heuristic*. The authors claim that this approach enables optimal solutions to be readily determined without having to resort to branch and bound techniques.

12.4 The Assumptions and Limitations

Although the research has been based on the assumption that the CMM is equipped with a motorized probe head (refer to section 5.7), this does not necessarily limit the approach to only those machines that are equipped with such a device. This is because probes can be fitted with an adjustable universal joint, which allows the probe to be reoriented manually before the inspection cycle commences. This clearly has the disadvantage that automatic reorientation is not available, but it does at least maintain the spirit of the assumption, which is to reduce the number of component orientations through probe reorientation.

The objective of determining the best probe configuration is confounded by another limitation of the research project described in section 5.7, which states that a single probe configuration should be used throughout the inspection cycle as the CMM used for all practical experimentation is not equipped with an automatic probe changing unit. For this reason, the probe configuration part of the Probe Setup Planner will
not be implemented, but a theoretical solution will be discussed as it is a small but crucial part of the inspection planning process.

It is also assumed that every probe configuration specified by the Probe Setup Planner is calibrated by the operator before the inspection cycle and that the points taken from each surface are adequately compensated for probe and stylus errors by the CMM controller using the calibration data.

12.5 The Method used for Probe Configuration Determination

The purpose of this activity is to determine the probe configuration that is best suited to probe each surface that requires measurement. The principal elements of the CMM probe that can be varied by this activity include:

- the stylus tip,
- the probe stylus,
- the probe body.

12.5.1 The Stylus Tip

The stylus tip is the only part of the probe that should touch the component and its design has a critical effect on the accuracy of the measurement. A range of different types are available, which include sphere, disc, star, cylinder and point, (figure 12.1) and each type is designed for a specific range of probing applications [166].

The sphere type is suitable for the vast majority of CMM probing applications as it can probe planar and curved surfaces in virtually any orientation. It is the default stylus tip on all CMMs and is available in a range of diameters typically from 0.3mm to 8.0mm.
The disc and star type styli find a similar range of applications, which include deep bores, undercuts and grooves (figure 12.2). They offer superior performance over the sphere type stylus when probing deep bores because the depth that the latter can probe is limited to the length of the stylus (figure 12.3a). An additional advantage can be found when probing holes that are not aligned with the axis of the probe, typically through a machining error. Under these circumstances the sphere type stylus may snag the surface with the stylus instead of the sphere and a false measurement will result (figure 12.3b). The disc and star type styli are also the only styli types that enable the probe to inspect a surface in the +Z direction, which is required when grooves and undercuts are measured.

Cylinder styli are used to measure threads and holes in sheet metal (figures 12.4a) and point styli are used to measure points, scribed lines and thread depths (figure 12.4b).

The selection of these stylus types depends on the surface geometry and the type of constraint to be measured and can therefore be determined by a simple if-then rule-base. An additional rule-base would be used to determine the dimensions of the various stylus types, which are also dependent on the geometry of the surfaces to be measured. Taking the case of the most commonly used sphere type stylus as an example, it is generally acknowledged that the largest radius sphere should be used whenever possible as this minimizes the effects of micro surface defects. Smaller radius spheres could be specified by the rule-base if the standard probe could not inspect a surface, such as a small hole.

In chronological terms, this stylus selection rule-base would operate before the PAD generator (refer to chapter 10) as this activity requires the stylus type and dimensions to determine the size and shape of the Probe Movement Envelope.
12.5.2 The Probe Stylus

The probe stylus typically used on CMMs is light and rigid and manufactured from non-magnetic stainless-steel. Its sole purpose is to transmit the force of the deflection from the stylus tip through to the probe mechanism within the probe body. Except under exceptional circumstances it is recommended that the shortest possible probe is used whenever possible, because the sensitivity of the probe is reduced as stylus length increases and this effectively reduces the accuracy of measurement [167]. Stylus length can typically vary between 10mm and 20mm without having a significant effect on probe accuracy, and so a 20mm stylus is generally specified as standard to allow a wider range of surfaces to be measured. The stylus selection rule-base described in the previous section would consider the length of stylus when measuring a surface and if the stylus was found to be too short (figure 12.3a) it would specify the disc or star stylus types instead.

12.5.3 The Probe Body

The probe body houses the probe mechanism and the electronics necessary to transmit the signal back to the controller. The most commonly used type of probe is a five-way device, so called because it can be deflected in five directions. A similarly sized six way device and a high accuracy probe for clean-room use are also available.

The principal variable characteristic of the probe body is its length, which can be extended if there is any risk that the motorized probe head or the CMM spindle will collide with the component. Extension bars that elongate the probe body length by up to 200mm can be specified (figure 12.5) using information generated by the PAD Generator as described in section 10.8. The PAD Generator determines the length of the probe body that it requires by calculating the distance between the point to be probed and the bounding box of the component along the axis of the Probe Approach
Direction. When the length of the stylus and the standard probe body have been subtracted from this figure, the required extension bar length is given, which can be used by the Probe Setup Planner to specify the appropriate probe configuration.

12.6 The Method used for Probe Orientation Determination

If a motorized probe head is fitted to the measuring machine, then the probe can be orientated to almost any attitude with respect to two rotational axes. The horizontal axis (A) rotates from 0° to 105° in 7.5° increments and the vertical axis (B) rotates from +180° to -180° also in 7.5° increments. The motorized probe head can be controlled automatically by a command in a part program, which asks it to assume an attitude previously calibrated by the operator at the beginning of the shift or inspection cycle depending on company practice.

If the machine is not fitted with such a device then the probe can be set in a particular orientation before the inspection cycle commences using an adjustable universal joint. This clearly has the disadvantage that automatic reorientation is not available and the machine would need to be stopped for manual readjustment of the joint if a new orientation was required.

Regardless of the method of probe orientation available on the machine, the purpose of the Probe Orientation Determination activity is to determine the orientations that the probe should assume in order to achieve the maximum access to the component.

The principal consideration involved in planning the orientation of the probe is to achieve the maximum access to the component so that the Surface Representation constructed from the points is as accurate as possible. Along with Component Setup Planning this is another example of the value of the PAD Generator to the inspection
planning process as this activity has already determined and validated the Probe Approach Directions that will offer the greatest access to each surface. The Probe Setup Planner therefore uses this information to group all the Probing Operations that have similar principal Probe Approach Directions into Operation Groups (refer to figures 8.4 and 8.8), so that the number of motorized probe head reorientations can be minimized during the inspection cycle. When the part program is generated (refer to section 15.5) the Probing Operations of each Operation Group are run sequentially and at the beginning of each Operation Group a command is generated to automatically reorient the motorized probe head to the desired orientation.

12.7 The Results

A similar version of the Probe Setup Planner was tested by both the Micro-Switch Cover case-study (refer to Appendix C) and the more recent Wrist Motor Body case-study (the basis of the Users Guide described in Appendix D).

Both case-studies successfully planned the probe orientations by grouping Probing Operations into Operation Groups with the same principal Probe Approach Direction. Probing Operations which were attached to Axis System and Datum Setting Operations were grouped separately from those attached to Measuring Operations as the Axis System and Datum Setting Operations have to be executed at the beginning of the cycle by definition. This tended to result in rather more Operation Groups than was strictly necessary (refer to Table D.7) which could be possibly be reduced by further analysis.

12.8 Discussion

The Probe Setup Planner described in this chapter is concerned with planning the probe configurations and orientations that are best suited to inspecting every surface on the component that requires measurement. It determines this information through
a combination of data provided by the PAD Generator and a simple if-then rule-base for stylus selection.

This approach is not currently matched by any other inspection planner under research, as apart from the work of ElMaraghy and Gu, which is limited to the trivial domain of turned components, there are no other research programmes investigating this area. It is, however, expected that the research team of Spyridi and Requicha at the University of Southern California will venture into this field as their method of accessibility analysis is highly appropriate to the solution of the problems faced in this area.

In conclusion, it can be said that this research has achieved its objectives, subject to the limitation of the use of single probe configuration during the inspection cycle mentioned in section 12.4. This limitation was discussed in section 5.7 and was set because the CMM used for practical experimentation was not equipped with an automatic probe changer. This precluded the testing of any probe configuration planning software and so this part of the Probe Setup Planner was not implemented. The machine was however equipped with a motorized probe head and therefore the probe orientation planning part of the Probe Setup Planner was implemented and tested in full.
Chapter 13

Probing Point Planning

13.1 Introduction

The generation of Probing Points is a central activity of the Inspection Plan and Code Generator as the points that are probed on the component have a critical effect on the accuracy of the Surface Representations and hence the quality of the results.

This chapter investigates both an ideal and a practical solution to this problem and discusses the benefits and shortfalls of both approaches.

13.2 The Objectives

The primary objective of this work is to research a Probing Point Planner that is capable of distributing a given number of points determined by the Operation Type Planner (refer to section 9.5.1) over the probeable parts of a surface that have been determined by the PAD Generator (refer to chapter 10).

Ideally the points should be distributed so that neighbouring points are approximately equidistant, which is understood to generate the most accurate representation of the surface as each region of the surface is given equal precedence. However the automatic determination of this ideal distribution is not necessarily a practical proposition as a system that is capable of distributing points in this manner over a surface with a possibly irregular boundary would clearly require a substantial amount of analysis and would represent a major research task.

Therefore in order to achieve a solution that would be practical to operate in the Inspection Planning environment, the requirement to determine an ideal point distribution is relaxed so that distributions that are close to the ideal are deemed acceptable.
This compromise is justified because the effects of the point distribution on measurement accuracy are not well understood by the research community and there is no guarantee that in searching for an ideal solution that the measurement accuracy will be increased.

Nevertheless the determination of the ideal point distribution is discussed in detail in section 13.6 of this chapter.

The secondary objective of this research is to determine the coordinates of Offset Points, which are offset from the Probing Point by the Probing Distance on the surface normal. These are the points, which the probe first approaches at rapid traverse before touching the surface, and hence the Probing Point, at a slower speed.

13.3 Research Elsewhere

The automatic planning of probing points is not well represented in the research that has been carried out elsewhere, which indicates the complexity of this problem and the difficulties that are faced in its solution.

The inspection planner of Kawabe, Kimura and Sata [96] employs a manual approach, which operates by asking the user to indicate the approximate positions to be measured on an orthogonal image of the component using the cursor. This is one of the few non-automatic processes in an otherwise highly-automated inspection planner, but the user at least benefits from the ability to select the points using the cursor. This approach also has the advantage that a human operator can intuitively position points in an approximately uniform distribution.

Duffie et al [99] have developed an inspection planner specifically for sculptured surfaces, which generates matrices of points for bicubic parametric surface patches. The inspection process is initiated by the user selecting the patch or group of patches
that require measurement so that the inspection planner can automatically determine the coordinates of the points to be measured. This is done by spreading the points uniformly over each patch in a grid-like formation dependent on the size and proportions of the patch (figure 4.3a). For each surface point specified by the system another two points are determined which lie on the surface normal and are equidistant from the surface; one behind and one in front (figure 4.3b). During the measurement cycle the probe traverses rapidly to the point in front of the surface and then traverses slowly to the point behind so that it touches the surface somewhere between the two points. Although this approach has been designed for the inspection of sculptured surfaces, the problems faced in this area are similar to those of conventional Coordinate Measurement techniques. Pairs of points are uniformly distributed over surface patches, which are suitably conducive to this operation through their near-rectangular shape.

Atkins and Derby [39] have combined the activities of Probing Point Planning and Probe Path Planning in a feature-based inspection planner that allows this information to be generated in either one of two ways. If the feature to be measured is sufficiently simple like a hole, then a parametrically controlled macro can be associated with the feature and called up by the operator when the probing points and probe path are being determined. Otherwise, in the case of a feature such as a plane, the operator is requested to specify the probing points and probe path manually, although this process is assisted by the automatic provision of additional coordinate systems relative to neighbouring features so that the operator is not restricted to moving the probe with respect to the master coordinate system. This approach highlights the difficulties faced through using a feature-based approach, as the user is forced to supply information that cannot otherwise be provided by the features. The planner is similarly restricted
by the inherent inability of the feature-based approach to vary the number of points that can be inspected.

No research has so far been directed towards determining the effects of point distribution on measurement accuracy but Coy of the University of Warwick [154] has researched the application of the autocorrelation function for determining the error incurred by not using all the possible data points when measuring a surface. It is claimed that research in this area will eventually lead to a solution for the inverse problem of the specification of the optimum number and distribution of coordinate points. Although this has not yet been achieved it is apparent from the research carried out so far that the distribution of points may well depend on the manufacturing method.

Finally, it is worth noting that the CMM manufacturers give little indication of the types of point distribution that are best suited to their Surface Representation algorithms. Ferranti [155] suggest that at least three points should be circumferentially distributed at both ends of the cone and cylinder Surface Representations, whereas the LK Tool Company [146] provide no advice at all. However all illustrations in both manuals show point distributions with approximately equidistant neighbouring points.

13.4 The Assumptions and Limitations

The only assumption made by this approach is that a point distribution with approximately equidistant neighbouring points will provide an accurate representation of the measured surface.

13.5 The Method

The idea behind this approach sprang from the realisation that a reasonable proportion of surfaces on components have regular boundaries as typified by the surfaces
on the Wrist Motor Body case-study (figure 13.1). It is possible to take advantage of this fact by establishing a set of point distribution templates for each type of regularly-bounded surface. The design of each template depends on the type of Surface Representation and the number of points to be measured, \( n \), as illustrated by the following examples.

1. **The Point Surface Representation.** Surfaces are represented by points both for simple dimensional measurements where only one point is used, and for geometrical tolerances such as parallelism where at least three points are used to represent datum planes. The single point distribution is simply a point in the middle of the surface (figure 13.2a), whereas the multi-point distributions are identical to the Plane Surface Representation and are therefore considered in that section.

2. **The Line Surface Representation.** The Line Surface Representation is constructed from at least two points, and the template is designed so that the desired number of points are equi-spaced along a line running through the centre of the surface on an axis predefined by the Operation Type Planner (figure 13.2b).

3. **The Plane Surface Representation.** The Plane Surface Representation is more complicated than the Line because the shape of the distribution varies with the number of points to be measured and to a certain extent with the length to width ratio of the surface. However it is possible to make an approximation to the ideal distribution, using odd and even point number templates as shown in figure 13.3. Unfortunately the distribution given by this approach can become less accurate as the number of points increases, where a greater number of horizontal rows would be required if the length to width ratio was low. However a template based on the examples shown in the figure was implemented as it was known that the
rule-base used by the Operation Type Planner (refer to section 9.5.1) was restricted to specifying low point numbers.

4. **The Circle Surface Representation.** The design of the Circle template is broadly similar to that of the Line in that a given number of points are equi-spaced along a circle whose centre-point lies at the centre point of the cylindrical surface under inspection (figure 13.4a). If the surface is open as shown in figure 13.4b then the first point is positioned at $\theta / 2$ (where $\theta$ is the point separation) and the remaining points are distributed as before.

5. **The Cylinder Surface Representation.** The design of the Cylinder template is similar to that of the Plane in that it is constructed from two Circle templates with the points shared between the two as shown in figure 13.5. If the number of points is odd then the circle with the larger circumference† receives the greater number of points. Again, the distribution given by this approach can become less accurate as the number of points increases, but this is tolerated for the same reasons as given in item 3.

6. **The Cone and Sphere Surface Representations.** These templates are identical in every respect to the Cylinder template.

The Probing Point Planner can only use these templates if regular boundaries are generated for each surface and this is achieved by retrieving a map of the probeable areas of each surface, which was generated as a by-product of the PAD Generator (refer to chapter 10). This map is constructed from cubic SDSM-type cells, which can be interrogated using standard SDSM query functions to determine the most extreme cells in each axis so that the dimensions and location of a regular surface boundary can

---

† The circumference at various sections of the cylinder may vary if the surface is intersected by other surfaces.
be retrieved (figure 13.6). The templates are then used to generate the points and each one is checked against the map to ensure that it lies within a probeable area. If it does not, the planner chooses the nearest unoccupied probeable cell and positions the point at its centre.

An Offset Point \( O \) is generated for each Probing Point \( P \) as shown below.

\[
O = P + dN
\]

where,

\( d \) is the Probing Distance, and

\( N \) is the local surface normal.

The final operation of the Probing Point Planner is to attach all the points to the corresponding Probing Operations in the Inspection Plan as illustrated in figure 8.8.

13.6 A Theoretical Method for Determining an Ideal Point Distribution

The problems that must be solved to determine an ideal point distribution are complex because the distribution of the points effectively depends on the boundary of the probeable part of the surface which is infinitely variable. Because of this complexity and the limited time resources available to the research project, a satisfactory solution has not been found, despite of the fact that some considerable amount of thought has been devoted to the problem.

The most productive line of enquiry is best visualised through the use of an analogy based on squishy balls, which is demonstrated by the following example (figure 13.7a). An imaginary squishy ball is obtained for each of the points that require probing so that the combined cross sectional area of all the balls is equivalent to the probeable area of the surface. The probeable area of the surface is then bounded by imaginary walls and the balls are placed inside the area one at a time (figure 13.7b). Once
the last ball has been placed, each ball will instantaneously stabilize the internal and external forces acting on it by distorting its shape. Once all the forces have been stabilized the centroids of the balls will approximately represent a uniform distribution of points within the probeable area of the surface as each point will be approximately equidistant from its neighbours and, when applicable, the boundary of the surface (figure 13.7c).

The feasibility of constructing a computer-model based on this type of behaviour was investigated and it was found that an iterative approach based on the cubic cell map of the probeable areas of the surface could have some benefit. This would operate by dividing the number of cells by the number of points in order to determine the number of cells that should represent each point. The model would then allocate groups of cells for each probing point until all of the probing points were represented by a fixed number of neighbouring cells. Inevitably these groups would have widely varying and unstable shapes and the next stage of the process would attempt to equilibrate the shapes by looping through each cell group identifying the cell exchanges that could be made. Cell exchanges could be made between neighbouring groups that resulted in a mutual improvement of the shape, where the ideal shape is represented by the minimum sum of the separation of each cell from the centroid of the group. However there was insufficient time to embark on what would have been a substantial piece of research, and this approach remains unimplemented.

13.7 The Results

The parametric template method of point distribution was tested by both the Micro-Switch Cover case-study (refer to Appendix C) and the more recent Wrist Motor Body case-study (the basis of the Users Guide described in Appendix D) although the Probing Point Planner implemented for the first case-study did not
incorporate the ability to check that the points were lying on probeable areas of the surface. Fortunately this did not affect the experiment as all of the probed surfaces were both cylindrical and regular and therefore matched the templates exactly. However the second case-study was based on a more complex component and many of the measured surfaces were intersected and irregular. The Probing Point Planner repositioned points that did not lie on probeable areas of the surface (refer to section D.5.1.6) and all surfaces were therefore probed satisfactorily. The point distributions generated in both experiments were observed to be acceptable despite the fact that points had been repositioned in the second experiment.

13.8 Discussion

This chapter discusses the problem of Probing Point Planning, which requires that a given number of Probing Points should be distributed over the probeable parts of each surface that requires measurement.

Ideally the points should be distributed so that neighbouring points are approximately equidistant but the automatic generation of such a distribution has been shown to be impractical as it would require a substantial amount of computational analysis to accommodate surfaces with irregular boundaries.

Very little relevant research has been carried out elsewhere in this area with the most significant work coming from Duffie et al who have investigated the related problem of point generation for sculptured surfaces. This approach benefits from the fact that the sculptured surfaces are defined in terms of bicubic parametric surface patches which are approximately rectangular in shape and are therefore conducive to uniform point distribution.

The solution presented in this chapter is similar in some respects to the research
of Duffie et al in that points are spread over regular surfaces using parametric point templates. Naturally it is impossible to guarantee that all surfaces will have regular boundaries on a component defined by CSG primitives and therefore each point is analysed to determine whether it lies on a probeable part of the surface. If it does not, the Probing Point Planner repositions the point to the closest unoccupied location which does lie on a probeable part of the surface. This approach was tested by two case-study experiments and was observed to operate satisfactorily in all respects.

The design of the point templates has a critical effect on the suitability of the resultant point distribution and it is evident that the templates designed for the plane, cylinder, cone and sphere Surface Representations perform less well as the number of points increases and also as the relationships between the dimensions of the surfaces varies. It is proposed that better templates could be researched, which would generate more accurate distributions under these conditions, but this was not considered necessary for this work as the Operation Type Planning activity (refer to section 9.5.1) is restricted to specifying low numbers of points.

The task of the Probing Point Planner would be greatly simplified if it could be supplied with a point density rather than a fixed number of points, as illustrated by the following example based on a complex planar surface (figure 13.8a). The inspection planner would operate by calculating the dimensions of the bounding box of the surface (figure 13.8) by simple analysis of the map of the probeable areas of the surface. If the length of the bounding box is found to be 36.0mm, the width is 215mm and the required point density supplied by the Operation Type Planner is 0.04 points/mm², then a point grid could be constructed and superimposed over the bounding box (figure 13.9a). Each point would need to be checked against the map to ensure that it is probeable, but it would not need to be relocated as it is no longer necessary to
maintain a fixed number of points. The resultant point distribution would be uniform and neighbouring points would be approximately equidistant (figure 13.9b). This method is simple, effective and would be easy to implement, but does rely on a relatively high point density. If the point density was only two or three times lower than the area of the surface then there would be a high probability of many of the points being rejected for a surface such as the one shown in the example (figure 13.8a). A high point density would certainly improve the accuracy of the Surface Representation but would also increase the inspection cycle time. Further research should be devoted to point quantity and distribution analysis before significant progress can be made in this area.
Chapter 14

Probe Path Planning

14.1 Introduction

One of the fundamental research issues in inspection planning is the automatic generation of collision-free probe paths, because it is currently a labour-intensive and time-consuming activity that can only be reliably achieved by manually teaching a path on a Coordinate Measuring Machine (CMM).

Clearly the key to a successful Probe Path Planner lies in its ability to analyse the geometric environment efficiently and this chapter describes just such an approach.

14.2 The Objectives

The principal objective of Probe Path Planning, as indicated by the introduction, is to generate a short, collision-free probe path between each pair of sequential points that require measurement. Ideally the process should be automatic and to be practical it should not need to rely on a substantial degree of computational analysis, which is clearly a hazard considering the type of problem to be solved.

The objective does not stress a requirement for the shortest path as one might expect, because this is impractical to achieve in a production environment as there are an infinite number of paths between each pair of points and, to determine the shortest, each one would need to be analysed. However it is important to identify a short path as this helps to minimize the inspection cycle time.

The requirement for a collision-free path is self-explanatory, and for several inspection planners this factor alone can justify the use of a solid modelling system [39,97]. This is because this type of representation provides a unique and unambigu-
uous environment, which is conducive to the analysis of interaction between solid objects (refer to section 2.2.4). For the path to be truly collision-free the Probe Path Planner will need to detect any interaction between the probe, the component, the machine and the fixture.

14.3 Research Elsewhere

Probe Path Planning is a key activity in the inspection planning process, which is reflected by a relatively high level of attention from the research community as illustrated by the following examples.

The method used by Kawabe, Kimura and Sata [96] is based on the creation of swept volumes to verify the path of the probe. The user selects points on the object using the cursor (figure 4.1a) and the planner checks the path between these points by generating a geometric model of the volume occupied by the probe (figure 4.1b). This model is then compared against the model of the object and if intersection is detected, the probe path has been found to collide with the object (figure 4.1c). However collision can be circumvented by retracting the probe to a safe space beyond the bounding box of the component (figure 4.1d), where no collisions are assumed to occur. This is by far the earliest attempt (1980) to verify safe probe paths in inspection and is also relatively effective. The method does not attempt to find the shortest or quickest path between the points and indeed the path could be quite circuitous depending on the geometry of the object and the position of the bounding box. The model of the movement of the probe is also somewhat approximate as no account is taken of the probe body, which could easily collide with the object. However this remains a simple and effective system, which should form the basis of more advanced and effective approaches.
Hopp and Lau [97,98] also worked on one of the earlier inspection planning research projects (1983) but they only propose a number of suggestions for how Probe Path Planning can theoretically be achieved. The simplest approach is to allow the operator to prescribe an explicit path between each pair of points, but this suffers from the disadvantages that it is time-consuming and also that it limits the flexibility of the system to alter the inspection sequence based on higher level considerations. Another method is to store a set of safe motion volumes for the component on that the basis that it should be easier to define and use these volumes rather than the object itself. Unless this could be done automatically, this would also be a very time-consuming process that might not be any easier then using the object itself. Certainly the probe movements between points could take advantage of simple safe motion volumes, but the analysis of surface touching movements would require more detailed volumes that would defeat the advantages of the approach. The final suggestion is to use the part geometry to generate safe probe paths and the authors suggest that this could be achieved through the use of a generate-and-test algorithm. The purpose of the exercise is to check if a linear probe path intersects with the object, which is, in the first instance, carried out by simple geometric analysis. If intersection is detected, a point is added to the path just off the edge of the surface that caused the collision, so that the path can now be considered as two shorter paths and consequently two simpler problems. The process is repeated until a safe path is found as illustrated by the algorithm depicted in figure 4.2. Although this approach is original and theoretically effective it may well prove uneconomic as it would require a great number of intersection checks.

The feature-based approach of Atkins and Derby [39] is combined with a Probing Point Planner and was therefore reviewed in section 13.3. As a brief summary, the
approach is based on the use of probe movements described in the form of macros and attached to features. For those features that are too complex for this approach the user is requested to supply suitable probe paths, which can be modified by viewing an animated sequence of the probe on an orthogonal image of the component. This approach highlights the difficulties involved when using features in an inspection environment as it is difficult to predict how a particular surface on a feature will be probed. This is because the inspection sequence depends on the type of constraint to be measured and the surrounding geometry which cannot always be known within a feature. However the operator is at least able to modify the probe path by viewing an animated sequence, but the authors admit that it is often difficult to visualize probe paths on a wire-frame representation of the object.

14.4 The Assumptions and Limitations

The approach described in this chapter assumes that simple fixturing methods as listed in section 11.6 have been used to hold the component. This obviates the need to ensure that the probe paths do not collide with the fixture as one of the requirements of simple fixturing is that it should only touch the component on its lower surfaces, which cannot be probed anyway.

The method also assumes that the probe can move safely when it is outside the bounding box of the component.

14.5 The Method

The Probe Path Planner described in this chapter was researched in two stages largely as a result of the varying requirements of the inspection planner as it developed during the project. In the early stages, a quick and reliable system was required to plug into an embryonic inspection planner that needed to generate part programs in
order to test the various elements of the process. However, as the project matured, a more sophisticated approach was researched both to maximise the potential of the underlying strategy and also to satisfy the objectives laid down for the work.

Therefore in the first instance a very simple non-optimal approach was adopted that took advantage of the fact that essentially there are two types of path that the probe follows; that is between an Offset Point and Probing Point when probing a surface (figure 14.1a) and also between an Offset Point and the next Offset Point when traversing between touches (figure 14.1b). This differentiation is important because it reveals the fact that the Probing/Offset Point path has already been verified by the PAD Generator (refer to chapter 10) and can therefore be used without any further analysis. The Offset/Offset Point paths can also benefit from the previous PAD analysis because these manoeuvres can be restricted to the safe zone defined by the Probe Movement Envelope that has been validated by the PAD Generator. This is achieved by retracting the probe to the region outside the bounding box of the component, whenever a movement is required that is not parallel to the Probe Approach Direction as illustrated in figure 14.2a. Implementation of this approach is straightforward as it is simply a case of generating a list of points that describe each probe path. Each point in the list is constructed by projecting the appropriate Offset Point on to the bounding box in the opposite direction to the Probe Approach Direction. Although this method is simple to implement and cheap to run it often generates inefficient probe paths as shown by the examples illustrated in figures 14.2b and 14.2c.

This method was implemented and successfully used for a number of early case-studies based on simple synthesized parts. However, during the research into PAD Generation it became clear that the Probe Movement Envelope (PME) approach could be used to greater effect during the Probe Path Planning activity. The underlying
theory of the PME approach is that an envelope that completely encloses the movement of the probe can be constructed from a number of simple planar surfaces described by spatially decomposed cubic cells. This is illustrated by the following example based on the linear path of a probe between two points (figure 14.3a). The PME (figure 14.3b) is a necessarily simplified geometric volume that is constructed according to the dimensions of the probe and the start and end points of the path that is under analysis. For the purposes of the approach described in this section it is sufficient to represent this volume as a collection of surfaces (figure 14.4a), each one of which is represented as a matrix of cell centre-points (figure 14.4b). This is acceptable because each point coordinate represents the centre-point of a cubic cell equivalent to the decomposed cells of the object. Therefore if the imaginary cells are built into a PME (figure 14.4c), it can be seen that the resultant volume is similar to that of the required PME shown in figure 14.3b. The generation of the matrix of cell centre-points is based on vector theory and is presented in section 10.8.

When this approach was first implemented, the planner checked the direct path between each pair of sequential points and if intersection was detected, a path was generated that retracted the probe to the bounding box of the component. This approach successfully circumvented the generation of inefficient probe paths as shown in figures 14.1b and 14.1c, but, it was noticed that the analysis could be simplified if certain conditions could be met. To assist the explanation of this simplification, the following elements are constructed according to figures 14.5a and 14.5b.

\[ A \] is the coordinate position of the start point,

\[ B \] is the coordinate position of the end point,

\[ N_A \] is the local surface normal at point \( A \).
\( \mathbf{N}_B \) is the local surface normal at point \( B \).

\( \theta_A \) is the angle between \( AB \) and \( \mathbf{N}_A \) when projected on to the plane of the Probe Approach Direction.

\( \theta_B \) is the angle between \( BA \) and \( \mathbf{N}_B \) when projected on to the plane of the Probe Approach Direction.

Therefore, if \( \theta_A > \pi/2 \) and \( \theta_B > \pi/2 \) then the surfaces are facing away from each other (figures 14.5c and 14.5d) and it would be impossible to traverse directly between the points and so points must be added to retract the probe to the component bounding box. Alternatively, if \( \theta_A < \pi/2 \) and \( \theta_B < \pi/2 \), the surfaces are facing towards each other (figures 14.6a and 14.6b) and it is worth using the PME method to verify the direct path between the two points, as it is possible that other surfaces can cause an obstruction. Finally, if \( (\theta_A > \pi/2 \) and \( \theta_B < \pi/2 \)) or \( (\theta_A < \pi/2 \) and \( \theta_B > \pi/2 \)) the surfaces are facing in approximately in the same direction (figures 14.5a and 14.5b) and it is possible that an indirect path could be found between the points. There is no point in checking the direct path as it is bound to pass through one of the surfaces, but the indirect path can be found by adding an intermediate point above the point where \( \theta_B > \pi/2 \). If, for example, \( \theta_A > \pi/2 \) then the intermediate point \( P \) would be determined by projecting \( A \) along the Probe Approach Direction to the plane which includes \( B \) (figures 14.6c and 14.6d). The path between \( P \) and \( B \) could then be analysed in the normal way and if no intersections are found \( P \) is kept as a Probe Path point. However if intersections are found (figures 14.6e and 14.6f), \( P \) is rejected and points are generated to retract the probe to the bounding box.

This approach exploits the strength of the PME method for detecting collisions.

\[ \dagger \] The full version of this conditional statement should be \( \pi/2 < \theta_A < -\pi/2 \), but it is shown in this shortened form for simplicity.
between the probe and other objects, but is intelligent enough to restrict its use to only those situations where a path is possible. It should also be noted that although this method has been explained by reference to planar surfaces it is in fact applicable to all types of CSG half-spaces.

Finally, with regard to the representation of this information within the Inspection Plan, Probe Paths are held as a list of Safe Rapid Points (refer to figure 8.8) which are attached to each Sub-Operation. Probe Paths are also determined between the Safe Rotate Position specified under Setup (refer to figure 8.4) and the first and last Offset Points of each Operation Group so that the probe can traverse without collision before and after re-orientation. These paths are also held as Safe Rapid Points under Operation Group (refer to figure 8.8).

14.6 An Alternative Heuristic Method

The method described in the previous section determines a short collision-free path between each pair of sequential points and therefore satisfies the objective of the exercise. However there are established techniques for finding near-optimum solutions to this type of problem, and this section discusses the potential of one of them which is known as the heuristic method.

As described in section 14.2, the determination of the shortest between two points is possible but not practical because there are an infinite number of paths between two points on an object and each one would need to be computed for the shortest collision-free path to be chosen. However it is exactly this type of problem that can benefit from a heuristic approach where the system determines a path to the goal by repeatedly moving to the neighbouring item that is nearest the goal.

This technique is easily applied to the spatially divided solid modelling environ-
ment used by this inspection planner and is best visualised by reference to the two
dimensional example illustrated in figure 14.7a. The approach operates by analysing
each of the twenty-six neighbouring cells (for a three-dimensional solution) to determine whether it is empty and if so its distance from the goal point. The neighbouring
cell that is closest to the goal point is then analysed for probeability using the PME
method established in section 10.8 and if it is not probeable, the system analyses the
next closest cell and so on. Once a neighbouring cell has been found that is probeable
and closer to the goal point, control is passed to that cell, and the process is repeated
until the goal point is reached (figure 14.7b).

Unfortunately this type of approach tends to generate a path that hugs the near-
side profile of the surface (figure 14.8a) and this path can be shortened using a simple
technique that analyses each point by projecting a line to the next point but one. If this
line is found to be probeable using the PME method then the mid-point can be deleted from the path and another line projected from the same start point to the next
point but one for the analysis to be repeated. The process halts temporarily once a
mid-point is found that cannot be deleted and control passes to that point for the
analysis to be re-started. The end result of this process is a substantially shorter path
(figure 14.8b).

14.7 The Results

The Micro-Switch Cover case-study (refer to Appendix C) was undertaken midway through the project and therefore used the original version of the Probe Path Planner which retracted that probe to the bounding box of the component between Offset Points. This generated perfectly adequate if non-optimum Probe Paths and did not require any geometric analysis.
The Wrist Motor Body case-study (the basis of the Users Guide described in Appendix D) used the current version of the Probe Path Planner, which demonstrated its advantage over the original approach by generating relatively short paths between every pair of Offset Points.

14.8 Discussion

The principal method described in this chapter attempts to find the shortest path between each pair of sequential points by analysing the direct route using the PME method described in section 10.8. Although this is a proven method for analysing the intersection between the probe and other objects it is also computationally intensive and for this reason the system carries out some simple analysis of the relative orientations of the probed surfaces to determine whether a direct or an indirect path is at least possible. If neither option is available there is no advantage in applying the PME method and the probe is retracted to the region outside the bounding box of the component which is assumed to be collision-free.

This approach only generates the shortest possible path when the direct route is acceptable and for the remaining cases only a relatively short path is generated. However there are mathematical methods for generating near-optimum solutions to this type of problem and an application of the heuristic approach is discussed with respect to Probe Path Planning in section 14.6. This alternative approach was not implemented but it is expected that it would generate marginally shorter paths than the PME method, although with a substantially greater computational requirement. This prediction can only be properly substantiated by a series of experiments, but it is argued that only a marginal improvement would be gained because the heuristic approach will always tend to generate more points in the path than the PME method. This has a negative effect on cycle time because most CMMs are designed to
incorporate acceleration and deceleration cycles when traversing and on short paths the CMM may not reach its terminal velocity before it begins to decelerate. Clearly on larger parts this effect will be minimized, but this inspection planner is better suited to small and medium-sized parts as geometric analysis becomes more computationally intensive as the number of SDSM cells increases. Therefore although the PME method does not generate the shortest possible path it does provide a path that does not adversely affect the inspection cycle time.

As far as the ability of the Probe Path Planner to generate a collision-free path is concerned there are three criteria that must be considered, which include whether the probe can collide with the object, the fixture or the machine. The PME method has already been established by the PAD Generator (refer to chapter 10) as an effective if computationally intensive method of detecting any interaction between the probe and the object and that quality can be applied equally well in this instance. Clearly the method will only be able to detect collisions with the fixture if it has been represented as a solid model, but this is not necessary as one of the constraints applied to this research (refer to section 5.7) is that simple fixturing methods should be used at all times. This obviates the need to ensure that the probe paths do not collide with the fixture as one of the requirements of simple fixturing is that it should only touch the component on its lower surfaces, which cannot be probed anyway. Collisions between the probe and the machine normally occur if the probe runs into the table in the \(-Z\) direction, which is potentially expensive as this type of collision often causes serious damage to the motorized probe head. The approach described in this chapter does not attempt to detect probe/machine collisions because the probe is always traversing linearly either between an Offset Point and a Probing Point or between Offset Points and as these points have been validated as lying on probeable areas of the component.
by the Probing Point Planner it would be impossible for the probe to collide with the
table. However if research was to be continued in this field it would be quite feasible
to model the table as a simple planar half-space on which a model of the fixture and
the component could be positioned and this would provide a complete model of the
environment, which would allow a truly collision-free path to be generated.

Finally it is worth noting that the inspection planner of Kawabe, Kimura and Sata
[96] uses a similar approach to the PME method described in this chapter, but their
system was only configured to model the probe stylus and would therefore ignore
many potential collisions. They also took advantage of the safe space beyond the
bounding box of the component but they were not able to plan indirect paths as
described in this chapter.
Chapter 15

Inspection Code Generation

15.1 Introduction

The output from the Inspection Planning process described in the previous seven chapters takes the form of an Inspection Plan that describes the entire inspection process for a particular component. The purpose of the Inspection Code Generator (ICG) is to translate this plan into a set of part programs that can be run on a particular inspection machine.

15.2 The Objectives

The principal objective of the Inspection Code Generator for this research project is to design and implement a system that will translate the Inspection Plan described in section 8.5 into a part program for the Ferranti CMM part programming language. This particular language is specified because the project currently undertakes all practical experimentation on a Ferranti Merlin 750 Coordinate Measuring Machine (CMM) (refer to section 5.6) which is a typical and popular inspection device. However there are many CMMs available on the market and although they are largely similar in structure and design the same cannot be said of the programming languages, which vary significantly between machines. Therefore any research into Inspection Code Generation must be capable of accommodating the fact that there are many different CMM programming languages in use and that there is the possibility that a company will require part programs to be generated in a variety of languages depending on the machines available.
15.3 Research Elsewhere

Most of the research programmes described in chapter 4 concentrate on the determination of the inspection method and leave the reader to assume that a corresponding part program can be generated automatically. This is not an unreasonable approach as the planning part of the process is technically more interesting than code generation and in many respects the role of an inspection code generator is simply one of translation. However if this new generation of inspection planners is to make an impact on industry it must provide a mechanism for generating output in a variety of languages. Atkins and Derby [39] lead the way in this direction with the adoption of the Dimensional Measuring Interface Specification (DMIS) [120], which is used as a neutral interface between the inspection planning system and any CMM that has a DMIS post-processor.

DMIS is an APT-like, human readable and writable language that is funded by the US CAM-I organisation and was originally developed by the Illinois Institute of Technology Research Institute in February 1985 [168]. It has since been approved by ANSI as an American National Standard on February 26, 1990 and it is expected that all CAD and CMM vendors will eventually supply the appropriate DMIS interfaces to remain competitive. Its principal objective is to provide a neutral bi-directional interface between CAD systems and a broad class of inspection devices including but not limited to CMMs, vision systems, optical comparators, robotic measuring devices, theodolites, photogrammators and laser-based measuring devices [169]. The language incorporates two basic types of statements, which are represented by geometric definition and process-oriented commands. The process-oriented commands control motion, machine parameters and other functions unique to the inspection process, whereas the geometric definition commands describe the geometry, constraints, coor-
As the DMIS vocabulary is similar in syntax and style to the APT NC programming language a DMIS part program can easily be written and the corresponding inspection device output understood by an inspector without needing to resort to automated equipment [170]. However DMIS will normally find applications in an automated environment such as the stylized example illustrated in figure 15.1 where a number of dissimilar CAD systems (A, B and C) can be used to design components and plan their inspection for a number of equally dissimilar dimensional measuring devices (1, 2, 3 and 4) [120]. Naturally, DMIS is the neutral interfacing mechanism that allows this fluency of communication and this can also be used to feed the inspection results back to either a Quality Information System (QIS) for use in statistical quality control and data archiving packages or to a Manual Interface System (MIS) for manual data analysis.

Although DMIS is the principal neutral interface language for dimensional measuring devices an alternative approach is provided by the Neutral Data File (NDF) specification [171] which is sponsored by the Coordinate Measuring Machine Manufacturers Association (CMMA). NDF, as its sponsor's title suggests, is restricted to representing part programs for Coordinate Measuring Machines and has not yet achieved the same levels of industrial acceptance as DMIS, most probably because of its limitation to CMMs and its reliance on a numerical rather than a textual language format.

15.4 The Assumptions and Limitations

The output from the Inspection Code Generator will be restricted to the Ferranti CMM part programming language owing to the absence of alternative programming environments to verify the ICG output.
15.5 The Method

From the start, the underlying and guiding principal of this research was that the Inspection Code Generator should not be required either to make any decisions or to analyse either the component or the inspection environment, therefore placing the onus on the Inspection Planner to provide all the information required to inspect the part. This leaves the ICG with the simple task of literally translating the Inspection Plan into a part programming language, which has the additional advantage that is theoretically easier to implement ICGs for other CMM languages.

The Inspection Code Generator was designed to operate in a series of logical stages as illustrated by the following sections. Frequent italicised references are made to data items in the Inspection Plan which is illustrated in figures 8.4 to 8.10.

1. **Select the first Component Setup.** The Component Setup Planner described in chapter 11 has determined a minimum set of component orientations that will allow every constraint that requires measurement to be inspected. As it may be necessary to carry out each Setup separately, a part program is generated for each component orientation.

2. **Create a text file to receive the part program for this Setup.** The ICG needs to open a text file in the *programs* sub-directory so that the ICG output can be fed into it as it is generated. The *Part Program Name* is provided by the Inspection Plan as shown in figure 8.4.

3. **Generate code to Initialize the CMM.** The information required to initialize the CMM will vary from machine to machine but it is likely to be broadly similar to the information required by the Ferranti programming language. The first lines of the part program state the name of the part program and the date and time of its generation for information purposes. The next statements are obligatory and
declare both a list of variables for internal use by the CMM and another set of
user-definable variables which include the positioning tolerance and the touch and
move speeds of the probe. These variables could be adjusted to achieve special
measurement characteristics but in this case the Inspection Planner relies on the
default values which are satisfactory for the conventional inspection methods
specified by the planner. The machine is set to measure in metric or imperial
which is provided by the Inspection Planner's Relationship Graph (figure 8.9) and
the first motorized probe head orientation is assumed before a command is
issued to ask the user to position the probe with respect to the measurement
datum which is provide by Datum Point under Datum Setting Operations in figure
8.5. The CMM's axis system is rotated according to the orientation of the com-
ponent provided by Transform under Setup in figure 8.4 as it is easier to rotate the
machine's axis system rather than the coordinates of the probing points. Finally
a temporary datum is set at the probe's current position and is shifted to the
component datum as all the probing points have been calculated with respect to
this point.

4. Select the first axis system. No code is generated at this stage but the ICG sets
a pointer to the first Axis System as shown in figure 8.4.

5. Generate code to set up the axis system. The operations that must be carried
out are provided by Axis System Setting Operations under Axis System in figures 8.4
and 8.5. These operations essentially consist of a Probing Operation to set the
principal axis and either one or two Probing Operations to set the secondary and
tertiary axes. The sequence of commands required to undertake each Probing
Operation is described under item 7 and after these commands have been issued
the appropriate Surface Representations are retrieved and the Axis System is con-
structured.

6. **Generate code to set up the datum.** The process for setting the datum point is similar to that used for setting the axis system in that it is essentially a list of Probing Operations as shown by *Datum Setting Operations* in figure 8.5. Therefore the Probing Operation commands are issued and the Surface Representations are retrieved so that the datum in each axis can be set. Finally a command is issued to shift the datum back to the component datum as all the Probing Points are calculated with respect to this point.

7. **Generate code to probe the component.** The Probing Operations are organised into Operation Groups which have a common Probe Approach Direction and so the sequence starts by setting a pointer to the appropriate *Operation Group* (figure 8.8) and by selecting the appropriate motorized probe head orientation given by *Probe Approach Direction*. Commands are issued to move the probe to the first Probing Operation using the *Start Safe Rapid Points* under *Operation Group* and each Probing Operation is carried out in turn. Providing the Probing Operation is a *Master Probing Operation* (figure 8.8) each of its Sub-Operations are carried out by issuing commands to move to the Offset Point, touch the *Probing Point* and to retract to the *Offset Point*. The probe is then moved to each of the *Safe Rapid Points* in order to traverse to the next *Offset Point* without colliding with the object. Once a Probing Operation is completed a command is issued to set the *Working Plane* and to fit the points to a *Surface Representation*. If the Probing Operation is a *Master Surface Representation* the dependent *Surface Representations* are also fitted. After all the Probing Operations have been carried the Operation Group is terminated by issuing commands to retract the probe from the object using the *Finish Safe Rapid Points* list under *Operation Group* (figure 8.8).
8. Select the next axis system. No code is generated at this stage but the ICG sets a pointer to the next Axis System as shown in figure 8.4 and returns to item 5. If the current Axis System is the last in the list, control is passed to item 9.

9. Generate code to measure the constraints. The actual commands required to measure a constraint depend on the Measurement Method of the Measuring Operation as shown in figure 8.6 but they all follow a similar pattern. Firstly the Surface Representations (figure 8.8) are retrieved from the Probing Operations attached to each Measurement Method as shown in figure 8.7. The Working Plane, if it is required, is set and a series of measurement commands appropriate to the Measurement Method of the Measuring Operation is issued. This command may for example call a CMM-based routine that calculates the flatness of a plane, the distance between two lines or the radius of a circle. Finally the results of this operation are formatted and sent to the printer along with a number assigned to the Measuring Operation for reference purposes.

10. Conclude part program and close the text file. The final statements depend on the programming language but for the Ferranti CMM, a command is issued to park the machine in its home position along with a set of subroutines used to store Surface Representations. The text file containing the part program is closed and is now ready to be used to inspect the part.

11. Select the next Component Setup. No code is generated at this stage but the ICG sets a pointer to the next Setup as shown in figure 8.4 and returns to item 2. If the current Setup is the last in the list, the ICG has finished its task and control returns to the IPCG menu.
15.6 The Results

Part programs were generated using this method for both the Micro-Switch Cover case study (Appendix C) and the Wrist Motor Body case study (the basis of the Users Guide described in Appendix D) and they were observed to inspect the parts according to the respective Inspection Plans.

15.7 Discussion

The case-study results prove that the ICG presented in this chapter is capable of generating a Ferranti part program that accurately reflects the Inspection Plan for two industrially sourced and relatively complicated components. It can therefore be argued that the principal objective of the Inspection Code Generation research has been satisfied. These successful results also imply that the Inspection Plan presented in section 8.5 is sufficiently detailed to describe the entire inspection process as the ICG is not required to compute any additional information above that provided by the Inspection Plan. However the case-studies have not tested every possible permutation of the Inspection Plan and it must be assumed until further tests are carried out that suitable code can be generated for all the other untested Measuring Operation and Axis System and Datum Setting Operation Types. This is not an unreasonable assumption because the method for handling each of these operation types follows a similar pattern which has previously been verified by the case studies.

The Inspection Code Generator was originally implemented for the LK CMES programming language [146] as the early practical experiments were carried out on a machine provided by this company. Part programs were generated successfully in this language although at this time the Inspection Plan was in the early stages of development and did not for example include such detail as Axis System and Datum Setting Operations. The ICG therefore generated a standard block of code that asked the user
to teach the machine how to set the axis system and datum point.

The ICG was subsequently reconfigured when the Ferranti CMM became available and it was found that the majority of modifications were concerned with a slightly different language syntax and the replacement of each LK procedure call for the Ferranti equivalent. This is easily achieved in the Ada environment used by the project as it only requires the modification and recompilation of package bodies as the functionality of each package described by the specification remains the same [132]. The major problem encountered with the Ferranti language was the fact that limitations are placed on the number of Surface Representations that can be in use at any one time and this contradicts the approach of the ICG which carries out all the Probing Operations before the constraints are measured. Clearly even a component of moderate complexity is going to utilize the six line Surface Representations that are available early in the inspection cycle and so a set of additional CMM subroutines were implemented that allow the ICG to specify the exact number of Surface Representations that it requires. The alternative solution of repeatedly utilizing the limited number of Surface Representations and carrying out the corresponding Measuring Operations was rejected because it would almost certainly result in the multiple measurement of surfaces and would be more difficult to implement.

When research was started into Inspection Code Generation, DMIS and NDF specifications had not been published and the intended solution to the generation of part programs in a variety of CMM programming languages was based on the use of the device-independent Inspection Plan and a set of simple ICGs. Clearly it has been proved that it is possible to implement this approach for the Ferranti programming language and in a restricted sense for the LK CMES language and it is suggested that the same approach could be adapted for every other CMM programming language.
However the emergence of DMIS as an ANSI standard and its likely acceptance on an international scale makes this approach redundant as a single ICG could be implemented for the DMIS language which would be capable of serving all CMMs through a post-processor. This is an ideal candidate for future work in this area once affordable DMIS post-processors appear on the market-place, which has yet to happen.
Chapter 16

The Final Discussion

16.1 Introduction

The research carried out to satisfy the objectives listed in section 5.2 has been described in detail in the preceding ten chapters. Each chapter was concerned with a discrete element of the Inspection Plan and Code Generation process and described:

- the objectives of the element under scrutiny,
- the competitive work that has been done elsewhere,
- the assumptions and limitations,
- the method used to satisfy the objectives,
- the results of relevant practical experiments.

Each chapter was concluded with a discussion of the research achievements and a comparison with the work done elsewhere.

The purpose of this chapter is therefore to summarize the Inspection Plan and Code Generation research by discussing the extent to which the objectives listed in section 5.2 have been satisfied.

16.2 The Automation of Inspection Plan and Code Generation

The research objective associated with this subject is:

*to research and implement a method for automating the entire Inspection Plan and Code Generation process of the Design to Manufacture Environment for a Coordinate Measuring Machine (CMM) with the support of a Product Modelling System.*
This objective was addressed by researching and implementing an experimental Inspection Plan and Code Generator that was comprised of the activities listed below.

- Inspection Planning (Chapter 8)
- Operation Type Planning (Chapter 9)
- Probe Approach Direction Generation (Chapter 10)
- Component Setup Planning (Chapter 11)
- Probe Setup Planning (Chapter 12)
- Probing Point Planning (Chapter 13)
- Probe Path Planning (Chapter 14)
- Inspection Code Generation (Chapter 15)

These activities represented every part of the Inspection Plan and Code Generation process and when implemented enabled a part program to be generated without any decision-based operator intervention.

The greatest single achievement that facilitated the complete automation of the process was concerned with the automatic generation of Probe Approach Directions (PADs). This information indicated the extent to which each surface could be accessed by the probe along a given approach direction and was used to plan Component and Probe Setups. The information could also be used to determine probe configurations and an extension of the technique was used to plan collision-free Probe Paths.

No other inspection planning research programmes have achieved the level of automation which is made possible by the PAD Generator, but Spyridi and Requicha [115,124] have researched a competitive approach for analysing surface accessibility. Although their approach considers every possible direction in which the probe can
access a surface it is not able to model the complete probe and therefore can only indicate the directions which are invalid.

16.3 The Integration of an Automated Inspection Planner within a Design to Manufacture Environment

The research objective associated with this subject is:

*to define and realise the role of an automated inspection planner within a highly-integrated Design to Manufacture Environment and to participate in the design of a Product Model Data Structure concentrating on those aspects related to the inspection activity.*

This objective was addressed in chapters six and seven of this thesis which discussed both the role of the Inspection Planner within a Design to Manufacture Environment and the method used to integrate the Inspection Planner with the Product Modelling System.

Depending on the particular industry, design to manufacturing environments can incorporate a very wide variety of tools and processes which would be impractical to accommodate in this experiment. For this reason, a skeletal Design to Manufacture Environment was researched by the project, which was restricted to the following processes:

- Feature-based Design
- Machine Operations Planning and Cutter Path Generation for a 3-axis Machining Centre
- Inspection Plan and Code Generation for a Coordinate Measuring Machine
- Manufacturing Data Analysis
Inspection Planning is a fundamental part of any design to manufacture environment because inspection is the means by which a manufacturing system maintains the quality of its output. In the Design to Manufacture Environment of the project, the Inspection Plan and Code Generator was used in parallel with the Machine Operations Planner to generate part programs for a manufacturing cell using the same component geometry and constraints retrieved from the Product Modelling System. After the part had been machined and inspected, the measurement results could be returned to the Product Modelling System for analysis by the Manufacturing Data Analysis facility to determine the existence and cause of any errors. If errors were detected, the Product Modelling System could be modified to ensure that subsequent components would be manufactured correctly.

Clearly the Product Modelling System plays a critical role in the integration of the various activities of the Design to Manufacture Environment and the integration of the Inspection Plan and Code Generator was no exception. Component geometry and constraints were retrieved by the Inspection Plan and Code Generator from the Product Modelling System so that the inspection of the component could be planned using exactly the same definition as the other manufacturing applications. Additionally, a data structure was researched and implemented to represent the output of the Inspection Planner which consisted of a complete description of how the component should be inspected. This data structure is known as the Inspection Plan and was used by the Inspection Code Generator to generate a part program in a specific CMM programming language. Clearly Inspection Code Generators could be implemented for any CMM programming language, but the emergence of DMIS as an international standard for CMM programming languages obviates this approach as a single Inspection Code Generator could be implemented to generate output in the DMIS language.
16.4 Practical Experiments Based on the Inspection Plan and Code Generator

The research objective associated with this subject is:

*to undertake experiments to establish the efficiency of a prototype Inspection Plan and Code Generator operating both individually and as part of the Design to Manufacture environment.*

A variety of case-studies were undertaken throughout the span of the research project based on both synthesized and industrially-sourced components.

The synthesized components (figures 8.1 and 8.2) were used to test early versions of the Inspection Plan and Code Generator as this approach had the advantage that the components could be designed to test specific aspects of the performance of the experimental software.

The Micro-Switch Cover (refer to Appendix C) was the first case-study to be based on an industrially-sourced component and was intended to test the capabilities of the original feature-based Inspection Plan and Code Generator.

The Design to Manufacture Environment which incorporates the Inspection Plan and Code Generator was tested using a case-study based on a structural road bearing supplied by the Glacier Metals Company (refer to section 6.6).

The final and most complex case-study was based on a Wrist Motor Body (the basis of the Users Guide described in Appendix D) supplied by GEC(FAST) which was used to test the performance of the final version of the Inspection Plan and Code Generator described in this thesis.
Chapter 17

Conclusions from the Research

The purpose of this chapter is to summarize the conclusions that were drawn from all aspects of the research described in this thesis.

1. It has been shown that it is possible to incorporate an Inspection Plan and Code Generator (refer to Chapter 6) within a highly-automated Design to Manufacture Environment (DME) supported by a Product Modelling System (PMS). A project-wide experiment was carried out which demonstrated the capabilities of the system to i) design a component using features, ii) plan the machining and inspection, iii) machine and inspect the component, and iv) to analyse the measured results and suggest the possible causes of error.

2. It has been established that an automatic Inspection Plan and Code Generator plays a central and strategic role within a highly-automated Design to Manufacture Environment as it analyses the same geometry and constraints as the applications that plan the manufacture of the product and generates the component measurements used by the data analysis facility (refer to Chapter 6).

3. An automatic Inspection Plan and Code Generator can be integrated with a Product Modelling System in order to facilitate the retrieval and storage of data from a central and neutral repository (refer to Chapter 7). The IPCG retrieves both the geometry of the component and its dimensional and geometrical tolerances, and stores the resultant Inspection Plan and the measurements from the CMM.

4. An additional benefit of the use of the Product Modelling System is the fact that the IPCG can be integrated with any third-party experimental or commercial CAD system that can be interfaced with the neutral Product Model Data
Chapter 17 - Conclusions from the Research

Structure supported by the PMS.

5. An Inspection Planning system can plan automatically all the information required to inspect a component on a CMM using a combination of if-then rules and geometric-analysis (refer to Chapter 8). Originally a feature-based approach was investigated but this was later rejected through its inability to adequately validate probe movements. However it is suggested that a combined approach would profit from the association of a restricted set of inspection information with each feature in order to reduce the amount of geometric analysis.

6. An Inspection Plan data structure is presented which allows all the the information required to inspect a component to be held within a Product Modelling System (refer to Chapter 8). This Inspection Plan is populated by the Inspection Planner and is translated by the Inspection Code Generator into a CMM part program. It is important because it allows other manufacturing applications to analyse and utilize the output of the Inspection Planner.

7. Measuring and Probing Operations can be planned automatically for the dimensional and geometrical tolerances defined in BS 308 and ANSI Y14.5M using an if-then rule-base that analyses both the constraints and the constrained geometry (refer to Chapter 9). This is a valuable first-step in the planning of operation types but it is appreciated that further research is required to determine sample point densities for each surface as the effect that this has on measurement accuracy is generally recognized [25,154] to be poorly understood by the research and industrial communities.

8. Axis System and Datum Setting Operations can be planned automatically for Coordinate Measuring Machines using a rule-base similar to the one used for planning Measuring and Probing Operations (refer to Chapter 9). If the
component definition does not specify datum surfaces for setting Axis Systems and Datums, the Inspection Planner can automatically identify suitable surfaces by analysing the tolerances associated with each surface.

9. It has been demonstrated that it is possible to calculate automatically the area of a surface that a probe can access from a particular Probe Approach Direction using a technique which analyses the swept volumes of the movement of the probe (refer to Chapter 10). It is therefore possible to determine the principal Probe Approach Direction which is the direction that allows the probe to access the greatest area of the surface as this information is essential to the automation of down-stream Inspection Planning activities such as Component Setup Planning and Probe Setup Planning.

10. The component orientations required to inspect every constraint applied to the component can be planned automatically by analysing the principal Probe Approach Directions of each surface (refer to Chapter 11). Candidate component orientations can be rationalised by determining the ability to fixture the component in that orientation which is reasoned to be indicated by the relative height of the the centre of gravity of the component.

11. The probe orientations required to measure each surface can be determined automatically using the Probe Approach Direction information (refer to Chapter 12). It is also possible to minimize the inspection cycle time by grouping Probing Operations into Operation Groups with the same principal Probe Approach Direction.

12. It has been argued that it is possible to determine automatically the probe configurations necessary to inspect every surface using an if-then rule-base and information derived from the Probe Approach Direction Generator (refer to
Chapter 17 • Conclusions from the Research

Chapter 12). This research was not implemented because the CMM used for practical experimentation was not equipped with an automatic probe-changer (refer to section 5.7).

13. The coordinates of Probing Points can be determined automatically using point templates for each type of surface although this method generates less acceptable point distributions as the number of points increases and the further the surface departs from a regular shape (refer to Chapter 13). However, it is suggested that distributing points over irregular surfaces is impractical for an automatic Inspection Planner and that future research should consider the possibility of specifying point densities, which would be significantly easier to process.

14. Short collision-free Probe Paths (refer to Chapter 14) can be generated automatically to allow the probe to move safely around the component during the inspection cycle. This information is derived using an extension of the method used to generate Probe Approach Directions which is based on the Spatially Divided Solid Modeller.

15. The Inspection Plan can be automatically translated by an Inspection Code Generator into a CMM programming language so that part programs can be generated and transferred to the CMM for execution (refer to Chapter 15). This capability has been demonstrated by translating the Inspection Plans into the Ferranti CMM programming language for a range of components. It is argued that Inspection Code Generators can be implemented for any CMM programming language, which would effectively permit the Inspection Plan and Code Generator to control a range of different CMMs.

16. Three experiments based on industrially-sourced components were successfully carried out to establish the viability of the Inspection Plan and Code Generator
when used both individually and when integrated with the Design to Manufacture
Environment of the project (refer to Appendices C and D and section 6.6).
Chapter 18

Recommendations for Future Research

The purpose of this chapter is to indicate the recommendations for future research in Inspection Plan and Code Generation.

- The possibility of probing a surface from more than one Probe Approach Direction should be investigated as this would increase the proportion of each surface that could be accessed without incurring a dramatic increase in computational analysis.

- The method of probe configuration determination described in Chapter 12 should be researched further and implemented as the application of probe changing devices will increase as the pressure to automate the inspection cycle becomes more intense.

- Research should be devoted to determine the effect of sample point density on measurement accuracy as this is a recognized research opportunity [25] that needs to solved before further progress can be made with the Operation Type Planning activity of the Inspection Planner.

- An Inspection Code Generator for the DMIS language format [120] should be researched and implemented as this would allow the Inspection Plan and Code Generator to control a wider range of CMMs.

- A computer-aided fixture design facility should be researched and implemented which allows the designer to construct a fixture from modular fixturing elements held in a CSG data-base. This would allow the Probe Approach Direction Generator to detect collisions between the probe and the fixture when generating PADs and collision-free Probe Paths.
Chapter 18 - Recommendations for Future Research

- The proposed *Mean Tolerance Ratio* method (refer to section 9.5.3) for identifying the surfaces to be used to set up Axis Systems and Datums would benefit from further investigation as it should identify the key surfaces on a component more accurately than the method currently implemented.

- If it is possible to extend the Spatially Divided Solid Modeller to decompose sculptured surfaces, the Inspection Plan and Code Generator should be reviewed in its totality to establish the viability of incorporating this wider geometrical domain.
References


References


References


References


[77] Lim, B. S. and J. A. G. Knight, A foundation for a knowledge-base computer integrated manufacturing system, Artificial Intelligence, Vol. 2, pp. 11-22,
1987.


References


[107] Teng, Sheng-Hsien and Bernard C. Jiang, Application of Vision System in Manufacturing Parts Alignment, Proc. of the IXth ICPR,


References


[122] Butler, Dr Clive and Dr Glen Mullineux, CMM’s and CAD - the missing link, *Quality Today*, pp. 31-32, June 1989.


Automeasure, 1988, Computervision Corporation Product Literature.

Peggs, Dr Graham N., BS6808 - Achieving traceability for co-ordinate measurements, Quality Today, pp. S1-S3, January 1989.


Shingo, Shigeo, Zero Quality Control, 1985, Productivity Press.


Low, Huang Boon, Data Derivation from CMMs, March 1990, Loughborough University of Technology B.Sc Thesis.

References


References


[166] CMM Styli, 1988, Renishaw Metrology Ltd Brochure Ref. H/1000/3065/01/B.

[167] Probing for Productivity on Co-ordinate Measuring Machines, 1988, Renishaw Metrology Ltd Brochure Ref. H/1000/5050/02/A.


Appendix A

The ISS Research Project

A.1 Introduction

The research described in this thesis formed one part of a larger research project entitled Information Support Systems for Design and Manufacture, which is funded by the ACME Directorate of the Science and Engineering Research Council and the Department of Trade and Industry.

The purpose of this appendix is to provide a summary of the objectives and achievements of this work.

A.2 The Objectives

- to build and experiment with an information support environment (the Product Description System) and to concentrate on machining and inspection applications.
- to demonstrate the value of the work to our collaborators by a skeletal Design to Prototype experiment, and by working with them on case studies.
- to understand by 1990 the problems and opportunities of linking both 1990 and mature CAE products together effectively. This understanding will be manifested in prototype software and an integrated experiment.
- to disseminate this understanding through research papers and seminars.

A.3 The Achievements

These objectives were achieved through the combined effort of ten research groups as shown below:

1. Product Description System
2. Classification and Form Features

3. Geometric Representation and Evaluation

4. Dimensions and Tolerances


6. Automated Workpiece Inspection

7. Manufacturing Data Analysis

8. Integration

The achievements of each one of these research groups is shown in the form of bullet statements for both clarity and brevity.

A.3.1 Product Description System

- A reference model for electro-mechanical products has been established. It includes an overall framework and data modules which allow the description of multi-dimensional geometry, features, component manufacture, dimensions and tolerances, assembly processes and feedback data.

- This Product Data Model has been strengthened by the inclusion of repeating patterns. The repeating patterns are captured and implemented using Parametric Data Models. A paper describing Parametric Data Models was presented at the IGES/PDES meeting in January 1990.

A.3.2 Classification and Form Features

- A data model which allows, but does not force, geometric, manufacturing and feedback data to be associated with features has been defined. Features may be of any or mixed dimensionality because they are a part of the multi-dimensional Geometry Graph.
The set of features that occurs in the Sponsors' case studies has been implemented as parametrically defined instances of this data model. Both explicit and implicit representations of each feature are maintained.

Four taxonomy structures for features have been designed and implemented. Features may be classified according to manufacturing strategy, topology, patterns that occur or the part family groups in which they are used. A single feature definition can appear in any number of taxonomies.

**A.3.3 Geometric Representation and Evaluation**

- The **Geometry Graph**, a graph-based data structure for multi-dimensional geometric representation, has been implemented and used for the homogeneous modelling of components with solid, surface and curve elements. The Parametric Data Model facility allows for a single consistent treatment of the geometry structures independent of dimension.

- Novel geometric algorithms based on recursive subdivision have extended the geometric domain to include objects with sculptured surfaces e.g. telephone handsets. In particular, these embody the ISOS (Inner Set Outer Set) approximation technique.

- A **2D Constraint Definition System (CDS)** capable of evaluating geometric parameters generated by a system of implicit geometric constraints has been implemented, based on a novel application of symbolic algebra techniques.

- The use of symbolic rather than numerical techniques for constraint evaluation allows the CDS to handle the problems of constraint sufficiency and determination of the desired solution.
A.3.4 Dimensions and Tolerances

- A data model, the Relationship Graph, has been specified and completed to satisfy the requirements for dimensions, dimensional tolerances, geometrical tolerances and surface texture.

- The applicability of the Relationship Graph has been demonstrated by the implementation of a tolerance analysis package and an interface to the Automatic Workpiece Inspection software.

- An evaluator has been written which enables the Relationship Graph to be created from a single homogeneous description which incorporates the dimensions and tolerances within the geometric description.

A.3.5 Machine Operations Planning and Cutter Path Generation

- An activity model has been produced describing the relationships between the features and general machining approaches to part program generation.

- A compatible data model has been established to integrate the two approaches. The definition of the model has been strengthened through interactions at the ISO PDES/STEP meetings.

- A method of analysing setup geometry has been defined to derive regions for machining and their spatial proximity.

- A method of analysing region geometry to identify tool size constraints and tool accessibility for rough machining has been defined.

- Machining plans and part programs can be produced using machine capability descriptions and by analysing product model data.
A.3.6 Automated Workpiece Inspection

- CMM part programs can be automatically generated from an SDSM evaluation of the Geometry Graph and the Relationship Graph without recourse to feature-based macros.
- A device independent Inspection Plan to be used during the automatic generation of CMM part programs describing every aspect of the CMM inspection process for the component has been designed and implemented.

A.3.7 Manufacturing Data Analysis

- A decision support aid based on influence diagrams has been employed for the experimental feedback software which explicitly identifies error(s) as a result of deviations in expected data from a featured-based workpiece representation.
- A structured library of user-modifiable fault clusters has been implemented which allows the user to perform the analysis during the manufacturing process.
- The experimental feedback software has been partially tested to establish the value of the technique using comprehensive feedback data.

A.3.8 Integration

- The role of a product data model in engineering information systems is demonstrated by (i) an integrated system consisting of closely coupled applications which support the phases from definition to inspection and (ii) an embryonic Engineering Data Management system where the emphasis has been placed upon a data model to support a company having software from more than one vendor. Applications use common data i.e. data that is held in a single product model.
- A unique environment for prototyping engineering data models demonstrates the
power of flexible, parametrised abstraction within the constraints of a specific data model.

- Substantial demonstration that the approach taken provides an appropriate route to integration, set in the context of the resources available.
- The experiment has made it possible to show how product data such as geometry and the dimensions and tolerances, can be represented and used as inputs in the generation of manufacturing code.
- For the first time evidence is shown of the potential value of feedback in what might in future be termed a closed loop concurrent design environment.
- The product model concept is shown as a key element in the integration of third party software.

A.4 Project Personnel and Collaborators

The following sections list the personnel that were attached to the project at both sites and the industrial collaborators.

A.4.1 University of Leeds

Principal Investigators

Professor A de Pennington  Dept. Mechanical Engineering
Dr M S Bloor  Dept. Mechanical Engineering
Professor P M Dew  School of Computer Studies
Dr D Holdsworth  University Computing Service
Mr L P Wickens  Dept. Mechanical Engineering

Research Staff

Mr P K Bell  Dept. Mechanical Engineering
Mr P G Dawson  Dept. Mechanical Engineering
Mr D R Dunnington  Dept. Mechanical Engineering
Ms A McKay  Dept. Mechanical Engineering

Associated Academic Staff
A.4.2 Loughborough University of Technology

Principal Investigators

Professor R Bell
Dr K Case
Dr N N Z Gindy

Research Staff

Mr M J Corrigall
Mr J S Eckersley
Ms L Lee
Dr X Gao

Associated Academic Staff

Mr R I M Young

A.4.3 Industrial Collaborators

BT (Design Technology Division)
BT (Product Design Group "B")
Renishaw Metrology
AT&T Istel

Glacier Metal Co Ltd
Pafec Ltd
GEC (FAST) Ltd
Lucas Engineering & Systems
Appendix B:

The Spatially Divided Solid Modeller

Summary

The Spatially Divided Solid Modeller (SDSM) \[12,79\] is a geometric representation which describes an object as a list of cubic cells that are classified into three types which include:

- \textit{full} for cells which are completely within the solid,
- \textit{empty} for cells which are completely outside the solid,
- \textit{boundary} for represents cells which are on the boundary of the solid and are therefore partially in the solid and partially in free-space.

Each cell can be analysed to determine its position with respect to the origin, its volume, its bounds and the addresses of its neighbouring cells. Query routines are also provided to generate lists of either the boundary cells, the full cells or the empty cells and another set of routines are available to determine the address of a cell at a point, the half-spaces that pass through a cell and the cells that lie on a half-space. All of these query routines are used extensively by the Inspection Planner.

Because there are often many thousands of cells in an object, it would be impractical to define a Spatially Divided Solid Model from first principles. Therefore a procedure is available which decomposes a CSG representation (refer to section 2.2.5) into an SDSM, as the characteristics of the CSG representation lend themselves to construction by the user. The decomposition of the SDSM is achieved in the first instance by completely enclosing the CSG representation of the component within a cubic cell known as the universe cell. The controlling application specifies a decomposition level and a termination criterion and the procedure begins the cyclic
decomposition process, which involves the segmentation of the universe cell into eight smaller cells. Each segmented cell is classified as either full, empty or boundary and the process is repeated until the decomposition level is reached. Applications can control the way in which the object is decomposed by specifying either the regular, absolute or relative termination criteria. The regular method decomposes each cell normally until the decomposition level is reached whereas the absolute method only decomposes boundary cells. The relative termination criterion is similar to the absolute method but operates from a decomposed cell rather than the universe cell.

The Inspection Planner uses the absolute termination criterion because this is the most economic method for analysing the boundary cells with which the planner is principally concerned. Clearly the determination of the decomposition level is critical as it effects the accuracy of the geometric representation and the computational intensity of the operation. If the decomposition level is too low the object is poorly defined and the Inspection Planner may, for example, detect erroneous collisions between the probe and the object. Conversely if the decomposition level is too high the object is well defined but the Inspection Planner will suffer from a high computational overhead as each additional decomposition level results in an eight-fold increase in the number of boundary cells.

Finally it is also possible to decompose the objects into either slices or columns depending on the requirements of the application. The Inspection Planner decomposes the object into cells as described above, which is also known as the octree method by virtue of the fact that each cell is decomposed into eight smaller cells.
Appendix C:

The Micro-Switch Cover Study: June 1989

Preface

This case-study was undertaken and written mid-way through the project and has been included in this thesis to provide an indication of progress during the research. The case-study describes the earlier feature-based Inspection Planner and in retrospect it can be seen that it marks the final development along this route before features were rejected in favour of the SDSM-based system described in the thesis.

C.1 Introduction

The subject of the case study is a Micro-Switch Cover (figure C.1) supplied by Lucas Engineering and Systems. The purpose of the case-study is to test the feature-based version of the Inspection Plan and Code Generator (IPCG) with respect to three hypotheses:

1. The Operation Type Determination activity plans linear dimensions, radial dimensions and the position geometric location tolerance.

2. The Setup Determination activity plans the probe and component orientations.

3. The Inspection Code Generator accurately translates the Inspection Plan into a Coordinate Measuring Machine (CMM) part program.

These hypotheses were selected for testing as they reflected recent research developments within the IPCG at the time of the experiment.

C.2 Equipment

This section describes the hardware and software used during the experiment.
The IPCG software was run on a Sun 3/50 work-station supported by a Sun 3/160 file-server. The Inspection Machine Controller software was run on a VAX 11/730 mini-computer. All practical experimentation was carried out on a Ferranti Merlin 750 Coordinate Measuring Machine (CMM) controlled by a Hewlett-Packard 300 series computer.

The IPCG is comprised of three distinct software packages:

- the Inspection Machine Planner (IMP)
- the Inspection Code Generator (ICG)
- the Inspection Machine Controller (IMC)

The following sections briefly describe the function of each software package and the limitations of the implementation at the time of the experiment.

C.2.1 The Inspection Machine Planner

The IMP uses a geometric description of the component and a list of relationships to be measured in order to generate an Inspection Plan which describes how the component should be inspected on a CMM.

The IMP is the most complicated part of the IPCG and is therefore further subdivided into four activities:

- Data Retrieval
- Operation Type Determination
- Setup Determination
- Operation Data Determination
Appendix C - The Micro-Switch Cover Case Study

C.2.1.1 Data Retrieval

The Data Retrieval activity consists of the following sub-activities:

- **Retrieve Library Form Features.** The Library Form Features include a description of each entity (surfaces and centre-lines) and a list of Approach Directions and a normal for each surface. The current array of Library Form Features implemented within the IPCG consists of block, pocket, through-pocket, channel, step, boss, blind-hole and through-hole.

- **Parametrize Library Form Features to describe the component.** The parametrized Form Feature information is a unique name, a list of moves and rotations and a list of parameter values for each Form Feature.

- **Retrieve Relationships.** Each relationship has a type, nominal and tolerance information and pointers to Form Feature surfaces. A reduced list of relationships can be manually selected.

- **Retrieve CMM description.** The CMM description is the dimensions of a single probe stylus to be used on the CMM.

All external data is currently hard-wired within procedures in the IPCG software.

C.2.1.2 Operation Type Determination

The Operation Type Determination activity consists of the following sub-activities:

- **Identify Measuring and Probing Operation types.** Measuring and Probing Operations are identified using a rule-base suitable for all dimensions and geometric tolerances described in BS308.

- **Determine Sub-Operation quantities.** The Sub-Operation quantities are currently retrieved from a look-up table. Presently the look-up table supplies
Appendix C - The Micro-Switch Cover Case Study

three Sub-Operations for a line Surface Representation, four for a plane and five for a circle.

C.2.1.3 Setup Determination

The Setup Determination activity consists of the following sub-activities:

- classify Approach Directions
- identify Setup orientations
- identify Master Probing Operations
- identify Operation Groups

Probe Approach Directions for each surface to be probed are the most important data for determining Setups within the IMP. A limited amount of Approach Direction information is extracted from each Library Form Feature, namely whether the probe can access the surface from that Approach Direction or not. However once the feature has been parametrized by the designer and combined with other features to form the component, some Approach Directions may have become corrupted. The Approach Directions are therefore classified by examining the boundaries of the surface to be probed and the length of the probe stylus. This enables the percentage coverage to be calculated so that the best Approach Direction can be determined. The best Approach Direction is the one with the highest percentage coverage of the surface. This approach is currently limited by the fact that no consideration is given to neighbouring surfaces.

When determining Setup Orientations, the IMP assumes that the CMM is equipped with a Motorised Head which has two rotational degrees of freedom. This means that the majority of components can be fully inspected in no more than two Setups. Setup Orientations are identified by examining each Setup axis and
Appendix C - The Micro-Switch Cover Case Study

determining:

1. whether it is possible to measure all the relationships

2. whether it is possible to measure all the relationships from one direction

3. the ranked stability of the Setup

Using this prioritised list, the IMP chooses a Setup Orientation and then allocates Measuring Operations to the Setup which gives the Probing Operations the greatest coverage of the surfaces involved.

Master Probing Operations are determined for both the probing operation and the surface representation.

Operation groups are identified by grouping the Probing Operations according to their highest classified Approach Directions.

C.2.1.4 Operation Data Determination

The Operation Data Determination activity consists of the following sub-activities:

- identify Safe Rapid Planes
- identify Datum-Setting Operation types
- identify Probing Data
- identify Sub-Operation links

Safe Rapid Planes and Datum-setting operations are currently identified manually. Probing Points are distributed in simple pre-defined patterns according to the type of surface representation involved. Surface Point and Probing Point coordinates are calculated from the surface information attached to the Form Features. The path between a pair of Sub-Operations is analysed to determine whether the probe needs to
retract to the Safe Rapid Plane. This is currently determined using a simple rule-base which refers to knowledge attached to form features.

C.2.2 The Inspection Code Generator

The ICG translates the Inspection Plan into a part program suitable for a particular CMM. The ICG used in this experiment generates code in the Ferranti programming language.

C.2.3 The Inspection Machine Controller

The IMC loads the generated part program down to the CMM and returns the results to the host computer.

C.3 Method and Results

The form feature data required by the Micro-Switch Cover was implemented within the IMP and run. The resultant Inspection Plan was translated by the Inspection Code Generator into a CMM part program. This part program was transferred to the CMM using the IMC. The part program was run on the CMM.

This section describes in detail how the experiment was executed with regard to each part of the IPCG.

C.3.1 The Inspection Machine Planner

C.3.1.1 Data Retrieval

- Retrieve Library Form Features. The Library Form Features used in this experiment were the boss (figure C.2a), through-hole (figure C.2b) and blind-hole (figure C.2c). They were parametrized and transformed to construct the Micro-Switch Cover as shown in figure C.3.
Appendix C - The Micro-Switch Cover Case Study

- Retrieve Relationships. A maximum of fourteen relationships could be inspected owing to the incomplete geometric description provided by the Form Features, which included two linear dimensions, eight radial dimensions and four position tolerances (figure C.4).

- Retrieve CMM description. A single probe was used for the inspection and this had a 2mm ball diameter and a 15mm stylus length.

C.3.1.2 Operation Type Determination

Fourteen measuring operation types were determined (Table C.1).

<table>
<thead>
<tr>
<th>Relationship</th>
<th>Measurement Type</th>
<th>Working Plane</th>
<th>Number of Probing Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>diameter</td>
<td>Z</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>diameter</td>
<td>Z</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>circle_circle</td>
<td>X</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>circle_circle</td>
<td>Y</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>diameter</td>
<td>Z</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>diameter</td>
<td>Z</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>diameter</td>
<td>Z</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>diameter</td>
<td>Z</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>circles_position</td>
<td>X &amp; Y</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>circles_position</td>
<td>X &amp; Y</td>
<td>2</td>
</tr>
<tr>
<td>12</td>
<td>circles_position</td>
<td>X &amp; Y</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>circles_position</td>
<td>X &amp; Y</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>diameter</td>
<td>Z</td>
<td>1</td>
</tr>
</tbody>
</table>

Table C.1: Measuring Operation Types

The measurement type refers to a particular method for measuring a relationship.

The measurement type *diameter* implies that a circle will be measured and the diameter element of the result will be compared against a nominal and plus and minus tolerances.

The measurement type *circle_circle* implies that two circles will be measured and the distance between them in one axis will be compared against a nominal and plus and minus tolerances.
The measurement type *circles_position* implies that two circles will be measured and the distance between them in each axis will be compared against a nominal. The deviations from these comparisons will be compared against a single position tolerance.

The working plane signifies the plane in which the measurement should be made. The position relationship has two working planes as two measurements need to be made to measure position.

It might be noted that Relationship 3 could not be measured and the reasons for this are discussed in section C.4.1.3.

Nineteen probing operation types were determined (Table C.2).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>main boss</td>
<td>1</td>
<td>circle</td>
<td>Z</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>main hole</td>
<td>1</td>
<td>circle</td>
<td>Z</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>main boss</td>
<td>1</td>
<td>circle</td>
<td>Z</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>main hole</td>
<td>1</td>
<td>circle</td>
<td>Z</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>tapped hole</td>
<td>1</td>
<td>circle</td>
<td>Z</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>main boss</td>
<td>1</td>
<td>circle</td>
<td>Z</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>lower left hole</td>
<td>1</td>
<td>circle</td>
<td>Z</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>lower right hole</td>
<td>1</td>
<td>circle</td>
<td>Z</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>upper right hole</td>
<td>1</td>
<td>circle</td>
<td>Z</td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>upper left hole</td>
<td>1</td>
<td>circle</td>
<td>Z</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>main boss</td>
<td>1</td>
<td>circle</td>
<td>Z</td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>lower left hole</td>
<td>1</td>
<td>circle</td>
<td>Z</td>
</tr>
<tr>
<td>13</td>
<td>11</td>
<td>main boss</td>
<td>1</td>
<td>circle</td>
<td>Z</td>
</tr>
<tr>
<td>14</td>
<td>11</td>
<td>lower right hole</td>
<td>1</td>
<td>circle</td>
<td>Z</td>
</tr>
<tr>
<td>15</td>
<td>12</td>
<td>main boss</td>
<td>1</td>
<td>circle</td>
<td>Z</td>
</tr>
<tr>
<td>16</td>
<td>12</td>
<td>upper right hole</td>
<td>1</td>
<td>circle</td>
<td>Z</td>
</tr>
<tr>
<td>17</td>
<td>13</td>
<td>main boss</td>
<td>1</td>
<td>circle</td>
<td>Z</td>
</tr>
<tr>
<td>18</td>
<td>13</td>
<td>upper left hole</td>
<td>1</td>
<td>circle</td>
<td>Z</td>
</tr>
<tr>
<td>19</td>
<td>14</td>
<td>tapped hole</td>
<td>1</td>
<td>circle</td>
<td>Z</td>
</tr>
</tbody>
</table>

Table C.2: Probing Operation Types

The Surface Representation is the geometrical entity against which the CMM's controller will attempt to fit the points taken from the surface. The IMP presently selects Surface Representations from a choice of line, plane, circle or cylinder. The Surface
Representation is determined directly from the measurement type.

C.3.1.3 Setup Determination

The Approach Directions were classified as shown in Table C.3.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-Z</td>
<td>100.0%</td>
<td>11</td>
<td>-Z</td>
<td>100.0%</td>
</tr>
<tr>
<td>2</td>
<td>-Z</td>
<td>100.0%</td>
<td>12</td>
<td>+Z</td>
<td>61.2%</td>
</tr>
<tr>
<td>3</td>
<td>-Z</td>
<td>100.0%</td>
<td>13</td>
<td>-Z</td>
<td>100.0%</td>
</tr>
<tr>
<td>4</td>
<td>+Z</td>
<td>100.0%</td>
<td>14</td>
<td>+Z</td>
<td>61.2%</td>
</tr>
<tr>
<td>5</td>
<td>-Z</td>
<td>100.0%</td>
<td>15</td>
<td>-Z</td>
<td>100.0%</td>
</tr>
<tr>
<td>6</td>
<td>-Z</td>
<td>100.0%</td>
<td>16</td>
<td>+Z</td>
<td>61.2%</td>
</tr>
<tr>
<td>7</td>
<td>+Z</td>
<td>61.2%</td>
<td>17</td>
<td>-Z</td>
<td>61.2%</td>
</tr>
<tr>
<td>8</td>
<td>+Z</td>
<td>61.2%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>-Z</td>
<td>61.2%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>+Z</td>
<td>61.2%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-Z</td>
<td>61.2%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table C.3: Best Approach Directions

It might be noted that the surfaces with the 61.2% coverage are the four through-holes at the corners of the Micro-Switch Cover. These holes are nominally 0.980 inches deep and therefore could not be fully penetrated by the 0.6 inch probe stylus.

When determining Setup Orientations, the IMP found that all relationships could be measured in each Setup axis except in the Z axis. Similarly the Z axis was found to be the only Setup axis which needed two Setup Directions. Finally the ranking in decreasing order of stability was found to be Z, X and Y. As a result of this process the planner suggested the X axis as the Setup axis, and all fourteen Measuring Operations were allocated to the +X Setup.

Master Probing Operations and Surface Representations were determined for each Probing Operation as shown in Table C.4.
Table C.4: Master Probing Operations

<table>
<thead>
<tr>
<th>Probing Operation Number</th>
<th>Master Probing Operation</th>
<th>Master Surface Representation</th>
<th>Probing Operation Number</th>
<th>Master Probing Operation</th>
<th>Master Surface Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>11</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>12</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>14</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>4</td>
<td>15</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>7</td>
<td>17</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>8</td>
<td>18</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>9</td>
<td>19</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

All Master Surface Representations are the same as the Master Probing Operations as all Probing Operations use the same Surface Representation (Table C.2).

Master Probing Operations 1 and 2 were allocated to the +X Operation Group and all the rest to the -X Operation Group.

C.3.1.4 Operation Data Determination

Safe Rapid Planes, Datum-Setting Operations and Sub-Operation links were all determined with the aid of manually inputted data. The Probing Data were determined automatically but are not tabulated as there are so many.

C.3.2 The Inspection Code Generator

The completed Inspection Plan (refer to section C.6) was translated into the Ferranti CMM Programming Language by the ICG.

C.3.3 The Inspection Machine Controller

The Part Program was down-loaded to the CMM by the IMC. Because the datum-setting facility has not yet been researched, the CMM was taught how to set the axis system and datum manually. The part program ran successfully and the results
Appendix C - The Micro-Switch Cover Case Study

are shown in Table C.5.

<table>
<thead>
<tr>
<th>Relationship</th>
<th>Measurement</th>
<th>Actual</th>
<th>Nominal</th>
<th>Deviation</th>
<th>+ Tol</th>
<th>- Tol</th>
<th>Out of Tol</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>diameter</td>
<td>.62474</td>
<td>.62500</td>
<td>-.00026</td>
<td>.00500</td>
<td>-.00500</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>diameter</td>
<td>.37502</td>
<td>.37500</td>
<td>.00002</td>
<td>.00500</td>
<td>-.00500</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>linear (y)</td>
<td>.55084</td>
<td>.54800</td>
<td>.00284</td>
<td>.00500</td>
<td>-.00500</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>linear (z)</td>
<td>.64610</td>
<td>.65000</td>
<td>-.00390</td>
<td>.00500</td>
<td>-.00500</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>diameter</td>
<td>.22211</td>
<td>.22200</td>
<td>.00011</td>
<td>.00350</td>
<td>-.00350</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>diameter</td>
<td>.22252</td>
<td>.22200</td>
<td>.00052</td>
<td>.00350</td>
<td>-.00350</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>diameter</td>
<td>.22210</td>
<td>.22200</td>
<td>.00010</td>
<td>.00350</td>
<td>-.00350</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>diameter</td>
<td>.22205</td>
<td>.22200</td>
<td>.00005</td>
<td>.00350</td>
<td>-.00350</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>linear (y)</td>
<td>.71951</td>
<td>.71800</td>
<td>.00151</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>linear (z)</td>
<td>.89044</td>
<td>.89000</td>
<td>.00044</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>position</td>
<td></td>
<td></td>
<td>.00301</td>
<td>.00800</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>linear (y)</td>
<td>.72074</td>
<td>.71800</td>
<td>.00274</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>linear (z)</td>
<td>.88971</td>
<td>.89000</td>
<td>.00029</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>position</td>
<td></td>
<td></td>
<td>.00549</td>
<td>.00800</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>linear (y)</td>
<td>1.21521</td>
<td>1.21800</td>
<td>-.00279</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>linear (z)</td>
<td>.89128</td>
<td>.89000</td>
<td>.00128</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>position</td>
<td></td>
<td></td>
<td>.00559</td>
<td>.00800</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>linear (y)</td>
<td>1.21642</td>
<td>1.21800</td>
<td>-.00158</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>linear (z)</td>
<td>.88882</td>
<td>.89000</td>
<td>.00118</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>position</td>
<td></td>
<td></td>
<td>.00317</td>
<td>.00800</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>diameter</td>
<td>.51910</td>
<td>.52000</td>
<td>-.00090</td>
<td>.00250</td>
<td>-.00250</td>
<td></td>
</tr>
</tbody>
</table>

Table C.5: The Measured Results

C.4 Discussion

This section discusses the performance of the IPCG during the experiment.

C.4.1 The Inspection Machine Planner

C.4.1.1 Data Retrieval

Form Features were used in this experiment to import both a geometrical representation and Approach Directions. However it was found that only part of the geometry of the Micro-Switch Cover could easily be described using Form Features, which would be unacceptable in a manufacturing environment as it would almost certainly prevent all the relationships from being planned and hence inspected.
In defence of Form Features, it could be argued that other parts of the geometry could be defined as Form Features (the figure-of-eight boss for example) but this would still leave most of the remaining geometry undefined.

The major problem with using Form Features in this case-study appears to be two-fold. Firstly some of the geometry is just too complicated to describe parametrically, such as the outer profile of the body of the Micro-Switch Cover (figure C.5). Secondly many of the other parts of the geometry could be described as Form Features but would become corrupted due to interaction with other Form Features, such as the large inner pocket which interacts with the two curved-slots (figure C.5).

The alternative to using Form Features is to build the geometric representation using simpler entities such as primitives, as in Constructive Solid Geometry, but this generally precludes the inheritance of manufacturing information. This information (Approach Directions) can be derived automatically in the case of the IPCG but at some cost with respect to execution speed. The proposed solution, which will be the subject of future experiments, is the combination of primitive geometry and Form Features. It is suggested that this will allow both the entire component to be geometrically described and manufacturing information to be inherited when possible.

C.4.1.2 Operation Type Determination

All the relationships were planned satisfactorily in all respects.

However a question-mark must appear above the simplistic method for determining Sub-Operation quantities, which are currently determined using a look-up table (section C.2.1.2). When programming a CMM manually, the user can choose the number of points from a pre-defined range that define a Surface Representation and in simple terms the more points used the better the Surface Representation. Inevitably
there is a trade-off between the quality of a Surface Representation and the inspection cycle-time and additionally it is felt that a stage can be reached where increasing the number of points will have no significant effect on the Surface Representation with respect to the measurement that is sought. It is proposed that this area is not well understood and is a worthwhile subject for future research. In the meantime, the look-up table approach at least generates a figure that can be used by down-stream activities.

C.4.1.3 Setup Determination

The method used for Approach Direction Classification works well for all but Relationship 3 (figure C.4). The reason for this is that the current method takes no account of neighbouring surfaces, which means that the activity determines that the probe will have 100% access to the upper hole, but when the part program was run the probe body collided with the outer wall of the component. This is a significant error and likely to be a frequent eventuality in most typical components. An alternative method which theoretically overcomes this problem is based on the geometric analysis of the component and this method will be further researched and tested during the remaining part of the project.

Another aspect of the Approach Direction Classification activity which is worth pursuing is the choice of probe stylus. If a range of probe stylus was available to the planner this activity could suggest an alternative longer stylus for Relationship 3. This enhancement of the Approach Direction Classification activity will be researched during the remaining part of the project if a probe-changer becomes available on the CMM and if time allows.

The Setup orientations, Master Probing Operations and Operation Groups were planned satisfactorily, although in the case of the Setup orientation planning activity it
is not felt that this case-study conclusively proves the efficacy of the approach. This is because the activity of planning Setup orientations is complex and the geometry of the case-study is essentially simple. It is also expected from the knowledge gained in this experiment that the Datum-Setting Operation Planning activity, to be researched in the forthcoming months, will necessitate the addition of an extra rule which checks whether datums can be set in both proposed Setup orientations.

C.4.1.4 Operation Data Determination

Safe Rapid Planes, Datum-setting operations and Probing Data were all planned satisfactorily.

C.4.2 The Inspection Code Generator

The part program ran satisfactorily on the CMM once Relationship 3 had been manually de-selected (section C.2.1.1) for the reasons explained in section C.4.1.3.

C.4.3 The Inspection Machine Controller

The part program was down-loaded to the CMM and run remotely. The results were printed out at the CMM printer.

C.5 Conclusions

The evidence provided by this experiment suggests that, for the Lucas Micro-Switch Cover, all three hypotheses listed in section C.1.0 are true with the following exclusions:

1. the Sub-Operation quantity determination sub-activity is satisfactory in its current form, but a more thorough approach would be desirable. Work in this area would be a suitable candidate for a future research programme.
2. the Approach Direction Classification sub-activity takes no account of neighbouring surfaces and will therefore occasionally produce inaccurate results. Work in this area is already scheduled, which should eliminate this problem.

3. a Probe Selection sub-activity would increase the probability of the IPCG being able to plan the measurement of all relationships. A worthy area for future work relying on the availability of a probe-changer on the CMM and sufficient time to carry it out.

4. the Setup orientations were planned satisfactorily but it is felt that future research in the area of Datum-Setting Operation Planning will necessitate the incorporation of an additional rule. Further planned case-studies will test the Setup orientation sub-activity after any modifications to the rule-base.
C.6 The CMM Part Program

1 Part: SUB Part
2
3 * program to inspect a X setup of micro_switch_cover
4 on 22-JUN-1989 at 09:03
5
6 COM /C2 / X,Y,Z,R,A,D,D2,Tpsn,Form,Pnt(*),Dos(*),N
7 COM /C3 / Mdst(*),Wdst(*),Rdst(*),Sys(*)
8 COM /C5 / Part(*),Tps(*),FC
9 COM /C6 / Pl,Plb_tip(*),Tps
10 COM /C7 / Ln(*),Cur(*),Pln(*),Cy(*),Spl(*)
12 COM /Cl2 / R 34,T 34,S 34,WC(*),W(*)
13 COM /Res / HsR(*),Sn
14 Print=ON
15 Allocat CIR.(7,4)
16 Inch
17 Maxvel
18 Pnt sel("Pln")
19 Sel Up(0.0)
20 Display( "MOVE PROBE TO START POSITION")
21 Bcp
22 Wai
23 Locat
24 Master("X,Y,Z")
25 Speed(100.0)
26 Prep(10.0)
27 Ptol(100.0)
28 Wkg_pln("XY")
29 Sel(1)
30 I
31 I level, align and set datum
32 I
33 Meas("Point",1,Temp Z)
34 Master("Z")
35 Meas("Point",1,Temp X)
36 Master("X")
37 Meas("Point",1,Temp Y)
38 Master("Y")
39 Meas("Plane 1","X Plane")
40 Level("P")
41 Master("X")
42 Meas("Line 1","X Line")
43 Align(0.0,0.1)
44 Meas("Circle 1","Bott")
45 Master("Y")
46 Meas("Move Y")
47 Move(0.010,0.010,5.000)
48 I
49 I inspecting the 4 X operation group
50 I
51 Move(0.762,0.350,5.000)
52 Sel_tip(1.0)
53 I
54 I probing surface 1 of main boss
55 I
56 Move(-0.120,0.350,5.000)
57 Move(-0.120,0.350,0.254)
58 Move(0.762,0.350,0.254)
59 Touch(0.762,0.253,0.154)
60 Move(0.762,0.350,0.254)
61 I
62 Move(-0.120,0.350,0.254)
63 Move(-0.120,-0.340,0.411)
64 Move(-0.120,-0.312,0.800)
65 Touch(0.762,0.097,0.297)
66 Move(0.762,-0.134,0.411)
67 I
68 Move(-0.120,-0.343,0.000)
69 Move(-0.120,-0.433,0.000)
70 Move(0.762,-0.433,0.000)
71 Touch(0.762,0.312,0.000)
72 Move(0.762,-0.433,0.000)
73 I
74 Move(-0.120,-0.433,0.000)
75 Move(-0.120,-0.134,0.411)
76 Move(0.762,-0.134,0.411)
77 Touch(0.762,0.097,0.297)
78 Move(0.762,-0.134,0.411)
79 I
80 Move(-0.120,-0.134,0.411)
81 Move(-0.120,0.350,0.254)
82 Move(0.762,0.350,0.254)
83 Touch(0.762,0.253,0.154)
84 Move(0.762,0.350,0.254)
85 I
86 Hold.plt(5)
87 Wkg_pln("YZ")
88 CALL Data_bit_circle(5.1,Cir(*))
89 I
90 I probing surface 1 of main hole
91 I
92 Move(-0.120,0.350,0.234)
93 Move(-0.120,0.055,0.400)
94 Move(0.741,0.055,0.400)
95 Touch(0.741,0.151,0.110)
96 Move(0.741,0.055,0.400)
97 I
98 Move(0.741,-0.021,0.664)
99 Touch(0.741,-0.058,0.179)
100 Move(0.741,-0.021,0.664)
101 I
102 Move(0.741,-0.067,0.000)
103 Touch(0.741,-0.188,0.000)
104 Move(0.741,-0.067,0.000)
105 I
106 Move(0.741,-0.021,-0.064)
107 Touch(0.741,-0.058,-0.179)
108 Move(0.741,-0.021,-0.064)
109 I
110 Move(0.741,0.055,-0.040)
111 Touch(0.741,0.151,-0.110)
112 Move(0.741,0.055,-0.040)
113 I
114 Hold_plt(5)
115 Wkg_pln("YZ")
116 CALL Data_bit_circle(5.2,Cir(*))
117 I
118 Move(-0.120,0.055,-0.040)
119 Move(-0.120,0.055,0.960)
120 I
121 I inspecting the X operation group
122 I
123 Move(1.259,0.661,4.960)
124 Sel_tip(3.0)
125 I
126 I probing surface 1 of tapped hole
127 I
128 Move(1.420,0.661,4.960)
129 Move(1.259,0.661,0.732)
130 Move(1.259,0.661,0.732)
131 Touch(1.259,0.758,0.803)
132 Move(1.259,0.661,0.732)
133 I
134 Move(1.259,0.505,0.783)
135 Touch(1.259,0.468,0.897)
136 Move(1.259,0.505,0.783)
137 I
138 Move(1.259,0.408,0.650)
139 Touch(1.259,0.238,0.650)
140 Move(1.259,0.408,0.650)
141 I
142 Move(1.259,0.505,0.518)
143 Touch(1.259,0.468,0.403)
144 Move(1.259,0.505,0.518)
145 I
146 Move(1.259,0.661,0.567)
147 Touch(1.259,0.758,0.497)
148 Move(1.259,0.661,0.567)
149 I
150 Hold_plt(5)
151 Wkg_pln("YZ")
152 CALL Data_bit_circle(5.3,Cir(*))

Appendix C - The Micro-Switch Cover Case Study
<table>
<thead>
<tr>
<th>Line</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>153</td>
<td>1 probing surface 1 of lower left hole</td>
</tr>
<tr>
<td>154</td>
<td>1 probing surface 1 of lower right hole</td>
</tr>
<tr>
<td>155</td>
<td>1 probing surface 1 of upper left hole</td>
</tr>
<tr>
<td>156</td>
<td>Move(1.420,0.661,0.567)</td>
</tr>
<tr>
<td>157</td>
<td>Move(1.420,0.726,0.896)</td>
</tr>
<tr>
<td>158</td>
<td>Move(0.680,0.726,0.896)</td>
</tr>
<tr>
<td>159</td>
<td>Touch(0.680,-0.628,-0.825)</td>
</tr>
<tr>
<td>160</td>
<td>Move(0.680,0.726,-0.896)</td>
</tr>
<tr>
<td>161</td>
<td>1 probing surface 1 of upper left hole</td>
</tr>
<tr>
<td>162</td>
<td>Move(0.680,0.715,0.898)</td>
</tr>
<tr>
<td>163</td>
<td>Touch(0.680,-0.752,-0.875)</td>
</tr>
<tr>
<td>164</td>
<td>Move(0.680,0.715,-0.898)</td>
</tr>
<tr>
<td>165</td>
<td>1 probing surface 1 of upper right hole</td>
</tr>
<tr>
<td>166</td>
<td>Move(0.680,-0.709,-0.890)</td>
</tr>
<tr>
<td>167</td>
<td>Touch(0.680,-0.829,-0.890)</td>
</tr>
<tr>
<td>168</td>
<td>Move(0.680,-0.709,-0.890)</td>
</tr>
<tr>
<td>169</td>
<td>1 probing surface 1 of right hole</td>
</tr>
<tr>
<td>170</td>
<td>Move(0.680,0.715,0.898)</td>
</tr>
<tr>
<td>171</td>
<td>Touch(0.680,-0.752,-0.955)</td>
</tr>
<tr>
<td>172</td>
<td>Move(0.680,0.715,-0.955)</td>
</tr>
<tr>
<td>173</td>
<td>1 probing routine completed</td>
</tr>
<tr>
<td>174</td>
<td>Move(0.680,0.715,-0.955)</td>
</tr>
<tr>
<td>175</td>
<td>Touch(0.680,-0.628,-0.955)</td>
</tr>
<tr>
<td>176</td>
<td>Move(0.680,-0.752,-0.955)</td>
</tr>
<tr>
<td>177</td>
<td>1 calculatung measurement</td>
</tr>
<tr>
<td>178</td>
<td>Hold_pos(5)</td>
</tr>
<tr>
<td>179</td>
<td>Wkg_pln(&quot;YZ&quot;)</td>
</tr>
<tr>
<td>180</td>
<td>CALL Data_fit_circle(5,4,Cir( *))</td>
</tr>
<tr>
<td>181</td>
<td>1 calculating measurements</td>
</tr>
<tr>
<td>182</td>
<td>1 calculating measurements</td>
</tr>
<tr>
<td>183</td>
<td>1 calculating measurements</td>
</tr>
<tr>
<td>184</td>
<td>Move(1.420,0.726,-0.895)</td>
</tr>
<tr>
<td>185</td>
<td>Move(1.420,0.726,0.895)</td>
</tr>
<tr>
<td>186</td>
<td>Move(0.680,0.726,0.995)</td>
</tr>
<tr>
<td>187</td>
<td>Touch(0.680,-0.628,0.955)</td>
</tr>
<tr>
<td>188</td>
<td>Move(0.680,0.726,0.995)</td>
</tr>
<tr>
<td>189</td>
<td>1 calculating measurements</td>
</tr>
<tr>
<td>190</td>
<td>Move(0.680,-0.715,0.811)</td>
</tr>
<tr>
<td>191</td>
<td>Touch(0.680,-0.752,0.955)</td>
</tr>
<tr>
<td>192</td>
<td>Move(0.680,-0.715,0.955)</td>
</tr>
<tr>
<td>193</td>
<td>1 calculating measurements</td>
</tr>
<tr>
<td>194</td>
<td>Move(0.680,-0.709,0.890)</td>
</tr>
<tr>
<td>195</td>
<td>Touch(0.680,-0.829,0.890)</td>
</tr>
<tr>
<td>196</td>
<td>Move(0.680,-0.709,0.890)</td>
</tr>
<tr>
<td>197</td>
<td>1 calculating measurements</td>
</tr>
<tr>
<td>198</td>
<td>Move(0.680,0.715,0.898)</td>
</tr>
<tr>
<td>199</td>
<td>Touch(0.680,-0.752,0.785)</td>
</tr>
<tr>
<td>200</td>
<td>Move(0.680,0.715,0.955)</td>
</tr>
<tr>
<td>201</td>
<td>1 calculating measurements</td>
</tr>
<tr>
<td>202</td>
<td>Move(0.680,0.726,0.896)</td>
</tr>
<tr>
<td>203</td>
<td>Touch(0.680,-0.628,0.825)</td>
</tr>
<tr>
<td>204</td>
<td>Move(0.680,0.726,0.896)</td>
</tr>
<tr>
<td>205</td>
<td>1 calculating measurements</td>
</tr>
<tr>
<td>206</td>
<td>Hold_pos(5)</td>
</tr>
<tr>
<td>207</td>
<td>Wkg_pln(&quot;YZ&quot;)</td>
</tr>
<tr>
<td>208</td>
<td>CALL Data_fit_circle(5,5,Cir( *))</td>
</tr>
<tr>
<td>209</td>
<td>1 calculating measurements</td>
</tr>
<tr>
<td>210</td>
<td>1 calculating measurements</td>
</tr>
<tr>
<td>211</td>
<td>1 calculating measurements</td>
</tr>
<tr>
<td>212</td>
<td>Move(1.420,0.726,0.896)</td>
</tr>
<tr>
<td>213</td>
<td>Move(1.420,1.211,0.895)</td>
</tr>
<tr>
<td>214</td>
<td>Move(0.680,1.211,0.895)</td>
</tr>
<tr>
<td>215</td>
<td>Touch(0.680,1.308,0.895)</td>
</tr>
<tr>
<td>216</td>
<td>Move(0.680,1.211,0.895)</td>
</tr>
<tr>
<td>217</td>
<td>1 calculating measurements</td>
</tr>
<tr>
<td>218</td>
<td>Move(0.680,1.221,0.881)</td>
</tr>
<tr>
<td>219</td>
<td>Touch(0.680,1.184,0.995)</td>
</tr>
<tr>
<td>220</td>
<td>Move(0.680,1.221,0.881)</td>
</tr>
<tr>
<td>221</td>
<td>1 calculating measurements</td>
</tr>
<tr>
<td>222</td>
<td>Move(0.680,1.228,0.890)</td>
</tr>
<tr>
<td>223</td>
<td>Touch(0.680,1.107,0.890)</td>
</tr>
<tr>
<td>224</td>
<td>Move(0.680,1.228,0.890)</td>
</tr>
<tr>
<td>225</td>
<td>1 calculating measurements</td>
</tr>
<tr>
<td>226</td>
<td>Move(0.680,1.221,0.898)</td>
</tr>
<tr>
<td>227</td>
<td>Touch(0.680,1.184,0.783)</td>
</tr>
<tr>
<td>228</td>
<td>Move(0.680,1.221,0.898)</td>
</tr>
<tr>
<td>229</td>
<td>1 calculating measurements</td>
</tr>
<tr>
<td>230</td>
<td>Move(0.680,1.211,1.065)</td>
</tr>
<tr>
<td>231</td>
<td>Touch(0.680,1.308,0.825)</td>
</tr>
<tr>
<td>232</td>
<td>Move(0.680,1.211,0.896)</td>
</tr>
<tr>
<td>233</td>
<td>1 calculating measurements</td>
</tr>
</tbody>
</table>
315 P_tol(" R.0.005")
316 M_tol(" R.0.005")
317 Output(" P.R")
318 !
319 CALL Recall_circle(4.1,Cir( *))
320 Wie_ph(" YZ")
321 Recall(" C")
322 Seq(6)
323 Output(" PS 4")
324 Nominal(" D.0.222")
325 P_tol(" D.0.0035")
326 M_tol(" D.0.0035")
327 Output(" P.4.D")
328 !
329 CALL Recall_circle(5.1,Cir( *))
330 Wie_ph(" YZ")
331 Recall(" C")
332 Seq(7)
333 Output(" PS 4")
334 Nominal(" D.0.222")
335 P_tol(" D.0.0035")
336 M_tol(" D.0.0035")
337 Output(" P.4.D")
338 !
339 CALL Recall_circle(6.1,Cir( *))
340 Wie_ph(" YZ")
341 Recall(" C")
342 Seq(8)
343 Output(" PS 4")
344 Nominal(" D.0.222")
345 P_tol(" D.0.0035")
346 M_tol(" D.0.0035")
347 Output(" P.4.D")
348 !
349 CALL Recall_circle(7.1,Cir( *))
350 Wie_ph(" YZ")
351 Recall(" C")
352 Seq(9)
353 Output(" PS 4")
354 Nominal(" D.0.222")
355 P_tol(" D.0.0035")
356 M_tol(" D.0.0035")
357 Output(" P.4.D")
358 !
359 CALL Recall_circle(1,1,Cir( *))
360 CALL Recall_circle(4,2,Cir( *))
361 Wie_ph(" XZ")
362 P_tol(" R.0.005")
363 M_tol(" R.0.005")
364 Output(" P.R")
365 !
366 !
367 P_tol(" R.0.005")
368 M_tol(" R.0.005")
369 Output(" P.R")
370 !
371 !
372 P_tol(" R.0.005")
373 M_tol(" R.0.005")
374 Output(" P.R")
375 !
376 CALL Recall_circle(1,1,Cir( *))
377 CALL Recall_circle(5,2,Cir( *))
378 Wie_ph(" XZ")
379 P_tol(" R.0.005")
380 M_tol(" R.0.005")
381 Output(" P.R")
382 !
383 !
384 P_tol(" R.0.005")
385 M_tol(" R.0.005")
386 Output(" P.R")
387 !
388 !
389 P_tol(" R.0.005")
390 M_tol(" R.0.005")
391 Output(" P.R")
392 !
393 CALL Recall_circle(1,1,Cir( *))
394 CALL Recall_circle(6,2,Cir( *))
395 Wie_ph(" XZ")
396 P_tol(" R.0.005")
397 M_tol(" R.0.005")
398 Output(" P.R")
399 !
400 !
401 !
402 !
403 !
404 !
405 !
406 P_tol(" R.0.005")
407 Output(" P.R")
408 Output(" P.R")
409 !
410 CALL Recall_circle(1,1,Cir( *))
411 CALL Recall_circle(7,2,Cir( *))
412 Wie_ph(" XZ")
413 P_tol(" R.0.005")
414 M_tol(" R.0.005")
415 Output(" P.R")
416 Output(" P.R")
417 !
418 !
419 !
420 !
421 !
422 !
423 !
424 !
425 !
426 !
427 CALL Recall_circle(3,1,Cir( *))
428 Wie_ph(" YZ")
429 Recall(" C")
430 Seq(10)
431 !
432 !
433 !
434 !
435 !
436 !
437 !
438 !
439 !
440 !
441 !
442 SUBEND
443 !
444 !
445 !
446 !
447 !
448 !
449 NEXT I
450 SUBEND
451 !
452 !
453 !
454 !
455 !
456 NEXT I
457 SUBEND
458 !
459 !
460 !
461 !
462 SUBEND
463 !
464 !
465 !
466 !
467 !
468 !
469 !
470 !
471 !
472 !
473 !
474 !
475 !
476 !
477 !
478 !
479 !
480 !
481 !
482 !
483 !
484 !
485 !
486 !
487 !
488 !
489 !
490 !
491 !
492 !
493 !
494 !
495 !
496 !
497 !
498 !
499 !
500 !
501 !
502 !
503 !
504 !
505 !
506 NEXT I
507 SUBEND
Appendix D:

A Users Guide for the Inspection Plan and Code Generator
Experimental Software

D.1 Introduction

The purpose of this appendix is to serve as a guide for the use of the experimental Inspection Plan and Code Generator (IPCG) software by describing both the sequence of events required to operate the software and the output that it generates.

The guide is based on a component known as the robot Wrist Motor Body (figure D.1 attached to the rear cover of the thesis) which is supplied by GEC(FAST) who collaborate with the ISS research project.

D.2 Before running the software

The Inspection Planning software was implemented in Ada and compiled and linked under version 4.0 of Unix on a Sun Series 3 work-station. It can be run in any directory which has the following sub-directories:

- programs
- output
- output/app_dirs
- temp

The programs sub-directory is the most important from the users point of view as it receives the part program files once they have been generated.

The output and output/app_dirs sub-directories are used by the Inspection Planner to receive planning information which can be used to monitor the progress of the system as they explain how each stage of inspection has been planned.
Appendix D - A Users Guide

The temp sub-directory was used during the implementation of the software to facilitate speedier test cycles. This was achieved by outputting detail data during CPU-intensive activities which included retrieving surface data, generating Probe Approach Directions, identifying probing points and identifying probing paths. This detail data could then be read on subsequent executions of the software to avoid unnecessary recomputation. Under normal operational circumstances the temp sub-directory should be cleared before execution of the Inspection Planning software.

D.3 Entering the Product Modelling Environment

The Inspection Planner operates within the Product Modelling environment (refer to chapter 6) alongside the other manufacturing applications of the project. In software terms this has been achieved by linking each of the application programs with the Structure Editor software which supports the Product Modelling environment. Therefore only one executable file is required which enables the user to enter the Product Modelling environment and execute any or all of the ISS project manufacturing applications.

Under the Unix operating system the executable file is run by typing its file-name whereupon the software requests the file-names of both a meta-structure and a dependent data structure (refer to section 6.5.1). The meta-structure required for this experiment is version 3.8 of the ISS project meta-structure and the dependent data structure, or Product Model, should be one constructed to describe the Wrist Motor Body.

The system walks the meta-structure to create an internal data-structure and then populates it with the data describing the Wrist Motor Body before allowing the user access to the system at the ROOT level (figure 6.4). The user is now permitted to navigate and examine any part of the Product Model which is achieved using a
comprehensive set of commands listed by entering the ? character. To operate the
Inspection Planning application the user needs to position the cursor above the
relevant part of the Product Model which can be done by entering 'E3 Solid followed
by N. These commands have the effect of positioning the cursor above the CSG
representation of the geometry.

The applications menu is summoned by entering D and contains the following
options:

1. Lambda evaluation
2. B-spline evaluator
3. SDSM evaluator
4. CDS interface
5. 3D wire-frame
6. Multiple evaluators
7. Dimensions & Tolerances
8. Machine planning
9. Inspection planning
10. Manufacturing data analysis
11. Other options
12. Miscellaneous

D.4 Entering the Inspection Planning environment

The Inspection Planner is enabled by selecting option 9 of the manufacturing
applications menu and this presents the Inspection Planning menu as shown below:

1. Initialize
2. Get CMM Data
3. Create CSG tree
4. Get object bounds
5. Decompose object
6. Get Centre of Gravity
7. Draw Wire-frame
8. Draw SDSM
9. Walk Relationship Graph
10. Plan Inspection
11. Generate Part Program
12. Get Measurements
13. Return to Main Menu

Typically the Inspection Planner is used by selecting each menu item in sequence, as described in the following sub-sections.

D.4.1 Initialize

This item deletes any data structures that have been previously created and sets up a collection of boolean operators that prevent the user from selecting the subsequent menu items in the wrong order. The user is also asked whether planning data is required and if this is requested the IPCG outputs data during the planning process which can be used to monitor its progress. Under these circumstances the user is also asked whether interrupts are required which have the effect of halting the Inspection Planner after each planning activity until the user hits carriage return.

D.4.2 Get CMM Data

The dimensions of the probe used on the CMM are set using an internal IPCG procedure. The probe body radius is set at 5mm, the probe stylus length at 15mm and the stylus tip diameter at 2mm.

Theoretically it is possible to store this information in the Product Modelling environment under *manufacturing information* (figure 6.4) and this would have the advantage that a number of different CMMs could be described and in greater detail. However this facility was not available at the time of this experiment.

D.4.3 Create CSG tree

The geometry of the Wrist Motor Body is represented as a CSG tree and when
this item is selected the tree is walked and an Ada representation of the tree is constructed. Research personnel at Leeds University created the CSG model of the component and it was agreed that an incomplete representation of the Wrist Motor Body would be acceptable. The geometric elements which were not implemented included all the threaded holes, the four φ15mm counter-bores and both the φ37.77/37.85 undercut in the main bore and the 45° chamfer. All of these features are either difficult to model or impossible to inspect with the available probe configuration.

D.4.4 Get object bounds

The object bounds nominate the region of the component that will be spatially divided for the purposes of geometric analysis. Normally the entire component should be sub-divided and therefore the object bounds are described by the lower and upper coordinates of the smallest box that totally encloses the component. The user enters these coordinates which in the case of the Wrist Motor Body are (0.0,0.0,−37.5) and (154.0,50.0,37.5).

D.4.5 Decompose object

This item spatially decomposes the CSG tree retrieved by menu item 3 in a series of steps. Firstly it determines the decomposition level which will result in a cell size that totally encloses the stylus tip retrieved by menu item 2, and the user is asked whether this decomposition level is acceptable. For this experiment a decomposition level of six is recommended which results in a cell size of 2.4mm. The component bounds supplied by menu item 4 are then expanded by 6mm in each direction in order to provide clearance planes for collision-free probe movements. Finally a spatially decomposed representation of the component is constructed from the CSG tree with respect to the previously determined expanded bounds and decomposition level.
D.4.6 Get Centre of Gravity

The centre of gravity of the component is computed for the Component Setup Planning activity and is achieved by analysing the position and volume of each full cell within the spatially decomposed representation of the component. For the Wrist Motor Body the centre of gravity is computed to be at (69.2,22.4,0.0) with respect to the axis system of the CSG representation as shown in figure D.2.

D.4.7 Draw Wire-frame

A wire-frame image of the component can be generated for visualisation purposes and displayed in a separate graphics window.

D.4.8 Draw SDSM

This item generates an image of the boundary cells of the spatially decomposed representation of the component and displays it in a separate graphics window. This facility is useful for verifying the extent of the spatial decomposition.

D.4.9 Walk Relationship Graph

The Inspection Planner has been integrated with the Relationship Graph data structure (refer to section 6.5.1) within the Product Modelling environment which represents the constraints that have been applied to the component. However the version of the Inspection Planner that incorporates this link was not available for this experiment and the constraints were instead retrieved from an internal procedure of the IPCG.

A representative sample of constraints were applied to the CSG tree as enumerated in figure D.1. Constraints 5 through to 12, 36, 39, 40 were not applied because the corresponding geometry was not implemented. Constraints 22,23,24 and
26 were not applied because the probe configuration used on the CMM is not be able to access the geometry. Constraints 31, 41, 42, 43 and 44 were not applied because these types of measurements could not easily be measured on a CMM. Constraints 13, 33 and 49 were not applied because these types of measurements have yet to be implemented within the Operation Type Planner of the IPCG.

Once the IPCG has created the internal Relationship Graph an information file called relationships is output to the output sub-directory. The following three constraints are examples from this file.

Relationship 1
Type: radial_dimension  
Measurement: radius  
Nominal Value: 1.20000E+01  
Nominal Axis: Y  
Plus tolerance: 1.00000E-01  
Minus tolerance: 1.00000E-01  
Node: cylindrical surface 12

Relationship 14
Type: linear_dimension  
Nominal Value: 2.50000E+01  
Nominal Axis: Y  
Plus tolerance: 1.00000E-01  
Minus tolerance: -1.00000E-01  
Node 1: planar surface 29  
Node 2: planar surface 33

Relationship 37
Type: concentricity  
Tolerance: 1.00000E-02  
Datum Node: cylindrical surface 54  
Test Node: cylindrical surface 57

D.4.10 Plan Inspection

This item generates an Inspection Plan by running through each part of the inspection planning process in sequence. Each step of this process is described in detail in section D.5.
D.4.11 Generate Part Program

Once an Inspection Plan has been created part programs can be generated for a particular CMM, which for this experiment was a Ferranti Merlin 750 Coordinate Measuring Machine (CMM) controlled by a Hewlett-Packard 300 series computer. A part program is generated for each Setup of the component and is written to the programs sub-directory. Excerpts from the part programs generated for the Wrist Motor Body can be found in sections D.6 and D.7.

D.4.12 Get Measurements

Once the part programs have been run, the results can be passed back to the IPCG software and incorporated within the Inspection Plan. This is achieved by copying the results text file from the CMM to the file called results in the IPCG directory and selecting menu item 12 of the Inspection Planning menu.

D.4.13 Return to Main Menu

This item allows the user to return to the main menu where the program can be terminated or other manufacturing applications can be run.

D.5 The Inspection Planning process

Once item 10 of the inspection planning menu has been selected the system automatically steps through each stage of the inspection planning process as detailed below.

D.5.1 Retrieving surface data

This activity analyses each surface which is constrained by a dimensional or geometric tolerance to determine its bounds and the locations of the bounding SDSM cells.
A standard SDSM query is used to generate a list of all the cells through which a half-space passes and an IPCG procedure is used to determine which of the cells actually lie on the surface. This is necessary because the half-space is by definition infinite and is therefore both certain to extend beyond the physical surface and likely to pass through other parts of the solid.

The simplest method for determining surface cells is to compile a list of boundary cells through which the half-space passes, which is easily achieved within the SDSM environment. At first glance this appears to have solved the problem but under certain circumstances such as those illustrated in figures D.4 and D.5 incorrect cells can be selected. Figure D.4a shows a worst-case surface intersection on the Wrist Motor Body which the IPCG must analyse and figure D.4b shows the corresponding surfaces in isolation. Figure D.5a shows the boundary cells through which the half-space passes and it is immediately clear that too many cells have been selected because the cylindrical surface has caused some of the cells that the half-space passes though to become boundary cells. The result of this approach is that the procedure tends to determine over-sized surface bounds which will cause down-line activities to plan operations incorrectly.

Therefore the procedure is modified to reject the boundary cells through which more than one half-space pass and the cells that this method selects are shown in figure D.5b. In most cases this modified approach would generate a set of undersized surface bounds that would be adequate for the purposes of down-line planning activities. However in the example depicted in figures D.4 and D.5 the approach determines a substantially undersized set of surface bounds because the cylindrical surface passes through a significant number of the boundary cells of the planar surface and therefore causes them to become invalidated. In extreme cases such as surfaces
19, 23, 27, 29 and 42 of the Wrist Motor Body, where the length of the planar surface is short in relation to the radius of the intersecting cylindrical surface, no surface cells can be found and the surface has to be rejected. This drawback is tolerated because the alternative SDSM-based method results in an incorrect part program and also because a more effective SDSM-based procedure for determining surface bounds has yet to be found. It is suggested that future work in this area either examines the possibility of avoiding the requirement to determine surface bounds or investigates the suitability of an alternative method of geometric analysis for this function.

D.5.2 Rejecting relationships with invalid surfaces

As described in the previous section, a limitation of the cell analysis procedure may result in the rejection of certain surfaces because they cannot be isolated in the SDSM representation of the part. This activity rejects any relationships that constrain rejected surfaces and in the case of the Wrist Motor Body these include relationships 3, 4, 32, 35 and 50.

D.5.3 Generating Probe Approach Directions

Probe Approach Directions (PADs) are generated and tested for each surface that requires measurement. This is achieved through the analysis of the spatial decomposition of the CSG representation of the object using the PME method described in chapter 10 of this thesis. Before the analysis begins, the user is asked whether the optimize facility should be used which allows the PAD Generator to cease analysis of PADs that have less than 100% surface coverage if a 100% PAD has been found already. This facility significantly reduces the computational requirements of the approach, without affecting the quality of the results.
The PAD Generator takes approximately 20 minutes to compute the Probe Approach Directions for every surface that required measurement as shown in Table D.1. This information is presented as the percentage areas that could be accessed by the probe from each of the six principal axes. Note that *opt* means that the Planner was optimizing analysis by halting the analysis of PADs that have less than 100% surface coverage if a 100% PAD has been found already.

<table>
<thead>
<tr>
<th>Surface Number</th>
<th>% +X</th>
<th>% −X</th>
<th>% +Y</th>
<th>% −Y</th>
<th>% +Z</th>
<th>% −Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0</td>
<td>100.0</td>
<td><em>opt</em></td>
<td><em>opt</em></td>
<td><em>opt</em></td>
<td><em>opt</em></td>
</tr>
<tr>
<td>2</td>
<td>100.0</td>
<td>0.0</td>
<td><em>opt</em></td>
<td><em>opt</em></td>
<td><em>opt</em></td>
<td><em>opt</em></td>
</tr>
<tr>
<td>3</td>
<td>10.9</td>
<td>0.0</td>
<td>0.0</td>
<td>100.0</td>
<td><em>opt</em></td>
<td><em>opt</em></td>
</tr>
<tr>
<td>4</td>
<td>2.9</td>
<td>7.8</td>
<td>100.0</td>
<td>0.0</td>
<td><em>opt</em></td>
<td><em>opt</em></td>
</tr>
<tr>
<td>5</td>
<td>14.3</td>
<td>21.4</td>
<td>33.3</td>
<td>8.3</td>
<td>0.0</td>
<td>100.0</td>
</tr>
<tr>
<td>6</td>
<td>14.3</td>
<td>21.4</td>
<td>33.3</td>
<td>8.3</td>
<td>100.0</td>
<td>0.0</td>
</tr>
<tr>
<td>9</td>
<td>0.0</td>
<td>10.3</td>
<td>25.7</td>
<td>22.3</td>
<td>0.0</td>
<td>100.0</td>
</tr>
<tr>
<td>12</td>
<td>0.0</td>
<td>0.0</td>
<td>27.1</td>
<td>20.3</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>14</td>
<td>0.0</td>
<td>10.3</td>
<td>25.7</td>
<td>22.3</td>
<td>100.0</td>
<td>0.0</td>
</tr>
<tr>
<td>17</td>
<td>0.0</td>
<td>0.0</td>
<td>27.1</td>
<td>20.3</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>27</td>
<td>0.0</td>
<td>0.0</td>
<td>26.7</td>
<td>20.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>33</td>
<td>0.0</td>
<td>100.0</td>
<td>100.0</td>
<td>0.0</td>
<td><em>opt</em></td>
<td><em>opt</em></td>
</tr>
<tr>
<td>42</td>
<td>0.0</td>
<td>0.0</td>
<td>26.7</td>
<td>20.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>46</td>
<td>0.0</td>
<td>26.4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>49</td>
<td>12.9</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>53</td>
<td>0.0</td>
<td>0.0</td>
<td>30.8</td>
<td>8.2</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>54</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>25.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>57</td>
<td>0.0</td>
<td>0.0</td>
<td>42.9</td>
<td>28.6</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>72</td>
<td>0.0</td>
<td>0.0</td>
<td>100.0</td>
<td><em>opt</em></td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>77</td>
<td>0.0</td>
<td>0.0</td>
<td>30.8</td>
<td>8.6</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>78</td>
<td>0.0</td>
<td>0.0</td>
<td>100.0</td>
<td><em>opt</em></td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>82</td>
<td>0.0</td>
<td>26.4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>85</td>
<td>12.9</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Table D.1: Percentage Surface Coverages

Percentage surface coverages are accurately determined for the majority of surfaces that required measurement and a good example of this is given by the results for surface 9 as shown below:

+X 0.0%

−X 10.3%
Each surface coverage is sufficiently accurate considering the length of the probe stylus and the geometry of the surface.

Problems were encountered with some of the cylindrical surfaces and in particular with surfaces 53 and 77, because these surfaces are so close to surface 6 that the thin wall that separates the surfaces is completely comprised of boundary cells rather than full cells. As the PME method only flags a collision when full cells are detected the PAD Generator determines that the +Z PAD has 100% access which is clearly erroneous. To avoid this problem for this experiment, the PAD Generator was restricted for cylindrical surfaces to analysing only those PADS that were coaxial with the surface centre-line. This has the unfortunate effect that the 100% +Z PAD is not detected for surface 42 but it is tolerated because the second-best 26.7% +Y PAD is selected instead.

The reason why collisions are only flagged when full cells are detected is because the function of the probe is to touch the surface and to do this it must enter the boundary cells of the surface. Therefore if the PAD Generator flagged collisions with boundary cells it would reject every single Probe Approach Direction. A more effective method of collision detection could be devised where only PME cells allowed to interact with boundary cells would be those of the stylus when it is in contact with the surface. This would effectively solve the problem encountered when analysing surfaces 53 and 77 and would be relatively straight-forward to implement.
D.5.4 Rejecting relationships with invalid approach directions

This activity rejects any relationships which constrain surfaces where a suitable PAD could not be found. As PADs could be found for each surface of the Wrist Motor Body no relationships were rejected.

D.5.5 Identifying datum setting operations

Axis System and Datum Setting Operations are determined by identifying the surfaces which are most heavily constrained on the component. These surfaces are therefore reasoned to be the most important surfaces and are used by the axis system and datum setting operations. The axis system and datum setting operations are determined by a rule-base similar to the one used for identifying Measuring and Probing Operation types.

For the Wrist Motor Body the Operation Type Planner found that surface 57 (figure D.1) was the most heavily constrained surface and so used this surface to set up the principal axis of the component axis system. Surface 9 was used to align the axis system and surfaces 57 and 4 were used to set the datum. This information is written to a file called `datum_ops` in the `output` sub-directory as listed below:

```
Level Probing Operation
Probing Operation: 1
   Surface Rep: cylinder
   Node: 57

Align Probing Operation
Probing Operation: 2
   Surface Rep: line
   Node: 9
   Working Plane: y

Alignment Angle: 0.000000E+ 00
Datum point < 109.0000 | 0.0000 | 0.0000>
```

Circle Probing Operation
Datum axis Y
Probing Operation : 3
   Surface Rep: circle
   Node: 57
   Working Plane: y

Line Probing Operation
Datum axis Y
Probing Operation : 4
   Surface Rep: line
   Node: 4
   Working Plane: z

D.5.6 Identifying measuring and probing operation types

Measuring and Probing Operations are identified using a simple rule-base as described in section 9.5.

Thirty-three measuring operations were planned as shown in Table D.2.
The Operation Type Planner created fifty-five Probing Operations as shown in Table D.3.
Table D.3: Probing Operation Types

D.5.7 Identifying the number of sub-operations required for each probing operation

The Sub-Operation quantities are currently retrieved from a look-up table as described in section 9.5.1. The look-up table supplies three Sub-Operations for a line Surface Representation, four for a plane, five for a circle and six for a cylinder.

D.5.8 Setting best approach directions

This activity analyses the Probe Approach Directions (PAD) generated for each surface and selects the one with the greatest surface coverage. If there is more than one PAD with greatest surface coverage then all of these are selected.
D.5.9 Identifying setups

This facility determines a minimum set of component orientations that will allow every constraint that requires measurement to be probed according to its principal Probe Approach Direction (refer to chapter 11). The first step of this process is to compile a list of component orientations which are parallel with the six axes of the component and allow the previously determined axis system and datum setting operations to be carried out. The planner then counts the number of constraints that can be measured in each component orientation and selects the orientation, which allows the highest number of constraints to be measured. The criterion for determining whether a constraint can be measured in a particular component orientation is based on whether any of the constrained surfaces have a single principal PAD that is opposite to the component orientation. If this is the case the CMM would be expected to probe a surface in the +Z direction with respect to its own axis system, which is clearly impossible and would therefore preclude the constraint from measurement. Once the first component orientation has been allocated the planner repeats the process by counting the remaining constraints that can be measured in each component orientation and again selecting the orientation that allows the highest number of constraints to be measured.

If at any stage in the process the planner finds that there are more than one component orientations that have the highest number of constraints, it selects the one which is the most stable. This is achieved by using the previously determined centre of gravity and by determining its height above the Z plane in each orientation. The one with the lowest centre of gravity is reasoned to be the most stable.

The Component Setup Planner counted the number of constraints that could be measured in each orientation (refer to Table D.4) and found that -X, -Y and +Z
allowed the same number of constraints to be measured but \(-Y\) had the lowest centre of gravity and was therefore selected. At this stage the user is given the opportunity to accept or modify this suggestion as required.

<table>
<thead>
<tr>
<th>Component Orientation</th>
<th>Number of Constraints</th>
<th>Height of Centre of Gravity (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>+X</td>
<td>26</td>
<td>90.8</td>
</tr>
<tr>
<td>-X</td>
<td>30</td>
<td>75.2</td>
</tr>
<tr>
<td>+Y</td>
<td>0</td>
<td>33.6</td>
</tr>
<tr>
<td>-Y</td>
<td>30</td>
<td>28.4</td>
</tr>
<tr>
<td>+Z</td>
<td>30</td>
<td>44.0</td>
</tr>
<tr>
<td>-Z</td>
<td>0</td>
<td>44.0</td>
</tr>
</tbody>
</table>

Table D.4: The First Component Setup Data

The Planner recounted the remaining constraints and found that this time \(+X\), \(-X\) and \(+Z\) allowed the same number of constraints to be measured but \(+Z\) had the lowest centre of gravity and was therefore selected (refer to Table D.5). Again the user has the opportunity to accept or modify this result.

<table>
<thead>
<tr>
<th>Component Orientation</th>
<th>Number of Constraints</th>
<th>Height of Centre of Gravity (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>+X</td>
<td>3</td>
<td>90.8</td>
</tr>
<tr>
<td>-X</td>
<td>3</td>
<td>75.2</td>
</tr>
<tr>
<td>+Y</td>
<td>0</td>
<td>33.6</td>
</tr>
<tr>
<td>-Y</td>
<td>0</td>
<td>28.4</td>
</tr>
<tr>
<td>+Z</td>
<td>3</td>
<td>44.0</td>
</tr>
<tr>
<td>-Z</td>
<td>0</td>
<td>44.0</td>
</tr>
</tbody>
</table>

Table D.5: The Second Component Setup Data

The Planner therefore allocated Measuring Operations 17, 30 and 37 to the second Setup and all of the other Measuring Operations to the first.

An information file entitled setups is written to the output sub-directory and this describes each Setup in detail. The following text is an excerpt from the file generated for the Wrist Motor Body.
D.5.10 Identifying master probing operations

It is often the case that the planner determines that some surfaces on the component need to be probed more than once because they are constrained by more than one dimensional or geometrical tolerance. In these circumstances it is not necessary to probe the surface more than once because the CMM can use a single Surface Representation for each Measuring Operation dependent on the Probing Operation under consideration.

This activity therefore analyses each Probing Operation to determine whether a
similar operation already exists and if it does the existing operation is flagged as a Master Probing Operation and the duplicated operation is deleted as shown in Table D.6.

<table>
<thead>
<tr>
<th>Probing Operation Number</th>
<th>Master Probing Operation</th>
<th>Probing Operation Number</th>
<th>Master Probing Operation</th>
<th>Probing Operation Number</th>
<th>Master Probing Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>20</td>
<td>20</td>
<td>39</td>
<td>39</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>21</td>
<td>21</td>
<td>40</td>
<td>53</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>22</td>
<td>22</td>
<td>41</td>
<td>49</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>23</td>
<td>54</td>
<td>42</td>
<td>48</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>24</td>
<td>26</td>
<td>43</td>
<td>49</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>25</td>
<td>54</td>
<td>44</td>
<td>46</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>26</td>
<td>26</td>
<td>45</td>
<td>54</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>27</td>
<td>54</td>
<td>46</td>
<td>46</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>28</td>
<td>28</td>
<td>47</td>
<td>47</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>29</td>
<td>36</td>
<td>48</td>
<td>48</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>30</td>
<td>54</td>
<td>49</td>
<td>49</td>
</tr>
<tr>
<td>12</td>
<td>21</td>
<td>31</td>
<td>36</td>
<td>50</td>
<td>55</td>
</tr>
<tr>
<td>13</td>
<td>13</td>
<td>32</td>
<td>39</td>
<td>51</td>
<td>53</td>
</tr>
<tr>
<td>14</td>
<td>14</td>
<td>33</td>
<td>36</td>
<td>52</td>
<td>54</td>
</tr>
<tr>
<td>15</td>
<td>54</td>
<td>34</td>
<td>39</td>
<td>53</td>
<td>53</td>
</tr>
<tr>
<td>16</td>
<td>21</td>
<td>35</td>
<td>38</td>
<td>54</td>
<td>30</td>
</tr>
<tr>
<td>17</td>
<td>54</td>
<td>36</td>
<td>36</td>
<td>55</td>
<td>55</td>
</tr>
<tr>
<td>18</td>
<td>22</td>
<td>37</td>
<td>39</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>19</td>
<td>38</td>
<td>38</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table D.6: Master Probing Operations

D.5.11 Identifying operation groups

The IPCG assumes that the CMM is equipped with a motorised probe head and a single probe stylus. The planner groups Probing Operations that have the same principal Probe Approach Direction into Operation Groups and assigns each group with a motorised probe head orientation equivalent to the common principal Probe Approach Direction (refer to chapter 12).

For the Wrist Motor Body experiment the Probing Operations are allocated to Operation Groups as shown in Table D.7.
Table D.7: Operation Groups

<table>
<thead>
<tr>
<th>Operation Group</th>
<th>Setup</th>
<th>PAD</th>
<th>Probing Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>+Y</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>-Z</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>+Y</td>
<td>3,4</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>+Z</td>
<td>28,38</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>-Z</td>
<td>36,39</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>+Y</td>
<td>5,6,7,8,9,10,11,48,49,53,54,55</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>+X</td>
<td>46,21,22</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>-X</td>
<td>19,20,47</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>+Y</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>-Z</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>+Y</td>
<td>3,4</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>+Y</td>
<td>13,27</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>-Y</td>
<td>14,26</td>
</tr>
</tbody>
</table>

D.5.12 Identifying the probing data for each sub-operation

Probing Points are distributed in simple pre-defined patterns according to the type of surface representation involved. Because these patterns are based on simple geometrical shapes the points are subsequently checked against a map of the probeable cells for each surface to ensure that they are valid.

For the Wrist Motor Body experiment Several points were moved by the Probing Point Planner so that they would be positioned over probeable parts of the surface. A good example of this is given by surface 2 where the original point template for a line generated points according to figure D.3a. The Probing Point Planner subsequently moved all three points to probeable areas of the surface (figure D.3b).

An information file called probing_data is output to the output sub-directory. The following two Probing Operations are examples from this file.

Probing Operation: 1
Node: 57
Operation Group: 1
Sub Operation: 1
Probing Point: < 104.2570 | 10.7844 | 6.4486>
Surface Coord: < 99.2156 | 10.7844 | 13.3031>
Sub Operation: 2
Probing Point: <116.0640 10.7844 3.7652>
Surface Coord: <122.8660 10.7844 7.3906>
Sub Operation: 3
Probing Point: <104.2570 10.7844 -6.4486>
Surface Coord: <99.2156 10.7844 -13.3031>
Sub Operation: 4
Probing Point: <104.2570 4.8719 6.4486>
Surface Coord: <99.2156 4.8719 13.3031>
Sub Operation: 5
Probing Point: <116.0640 4.8719 3.7652>
Surface Coord: <122.8660 4.8719 7.3906>
Sub Operation: 6
Probing Point: <104.2570 4.8719 -6.4486>
Surface Coord: <99.2156 4.8719 -13.3031>

Probing Operation: 2
Node: 9
Operation Group: 2
Sub Operation: 7
Probing Point: <93.3031 25.5656 34.0844>
Surface Coord: <93.3031 25.5656 34.0844>
Sub Operation: 8
Probing Point: <111.0410 25.5656 34.0844>
Surface Coord: <111.0410 25.5656 28.0844>
Sub Operation: 9
Probing Point: <131.7340 25.5656 34.0844>
Surface Coord: <131.7340 25.5656 28.0844>

D.5.13 Identifying probing operation reference numbers

This activity is required because the Ferranti programming language is incapable of supporting more than a small number of each Surface Representation. Therefore extra subroutines are added to the end of the Ferranti part program (refer to section D.6 and D.7) which create new Surface Representations which are referenced by numbers attached to each Master Probing Operation by this activity.

D.5.14 Identifying sub-operation links

Probe Paths are determined between each pair of consecutive Offset Points using the PME method described in chapter 14. The planner determines whether the points are on the same plane normal to the PAD, and if they are not, a temporary point is
added by projecting the lower point parallel to the PAD and on to the aforementioned plane. The path between the two points on the same plane is checked using the PME method and if it is collision-free a Probing Path is constructed that passes between the two Offset Points and the temporary point. If a collision is detected, two temporary points are added by projecting both Offset Points along the PAD and on to the bounding box of the component. A Probing Path is then constructed that passes between the two Offset Points and the two temporary points.

An information file called *subop_links* is output to the *output* sub-directory.
D.6 The Part Program for the first Setup in the Ferranti part programming language

Part: SUB Part
1
1 Program to inspect 'V' setup of test Yang prog
2
1 on 05-JUL-1990 at 18:22
3
1
6 COM / C2/ X. Y, Z, R, A, D, D1, TPnP, Form, Puc( ), Dcu( ), X, N
7 COM / C3/ Mat( ), Wcu( ), Rcu( ), Sys( )
8 COM / C5/ Puc( ), Pcf( ), Pct( )
9 COM / C6/ Pd, Prc( )
10 COM / Cu/ Ln( ), Ln( ), Cyn( ), Spk( )
12 COM / C14/ R_set( ), R_2d( ), W( )
13 COM / Rset/ Work( ), Sn
14 Proc=" ON "
15 LOCATE Ln(11,7)
16 LOCATE Cyn(14,4)
17 LOCATE Cyn(1,7)
18 Move
19 Manual
20 Rot_cyl
21 Pro_set(78.97)
22 Seq(1)
23 Speed(125.0)
24 Prpom(10.0)
25 Ptol(0.0)
26 Sol_tip(2.0)
27 Display(" MOVE PROBE -6MM FROM DATUM")
28 Beep
29 Wait
30 Wg_phn("YZ")
31 Rot_at(90.000)
32 Locate
33 Mucr(14,2)
34 Datum("T Y0.000")
35 Datum("T X-100.000")
36 Datum(1 Y0.000)
37 Datum(T Z0.000)
38 !
39 ! initialization complete!
40 !
41 !
42 ! start operation group 1
43 !
44 !
45 ! Sol_tip(2.0)
46 ! Move(106.616, -6.000, 7.642)
47 !
48 ! Probing node 57
49 !
50 !
51 !
52 !
53 !
54 !
55 !
56 !
57 !
58 !
59 !
60 !
61 !
62 !
63 !
64 !
65 !
66 !
67 !
68 !
69 !
70 !
71 !
72 !
73 !
74 !
75 !
76 !
77 !
78 !
79 !
80 !
81 !
82 !
83 !
84 !
85 !
86 !
87 !
88 !
89 !
90 !
91 !
92 !
93 !
94 !
95 !
96 !
97 !
98 !
99 !
100 !
101 !
102 !
103 !
104 !
105 !
106 !
107 !
108 !
109 !
110 !
111 !
112 !
113 !
114 !
115 !
116 !
117 !
118 !
119 !
120 !
121 !
122 !
123 !
124 !
125 !
126 !
127 !
128 !
129 !
130 !
131 !
132 !
133 !
134 !
135 !
136 !
137 !
138 !
139 !
140 !
141 !
142 !
143 !
144 !
145 !
146 !
147 !
148 !
149 !
150 !
151 !
152 !
153 !
Appendix D - A User's Guide

153 Move(75.573,-5.252,1.427)
154 I
155 Move(99.839,-5.252,1.427)
156 Touch(99.839,0.748,1.427)
157 Move(99.839,-5.252,1.427)
158 I
159 Hold_pt(3)
160 WkB_pin("XY")
161 CALL Data_ft_line(3.3,Ln(_ *))
162 Move(99.839,-6.000,1.427)
163 Move(99.839,-71.000,1.427)
164 CALL Recall_circle(1.3,Cir(_ *))
165 Recall("C")
166 Mute("Z")
167 Mute("X")
168 CALL Recall_line(2.4,Ln(_ *))
169 Recall("L")
170 Mute("Y")
171 Touch(T X:109.000")
172 Touch(T Y:0.000")
173 Touch(T Z:0.000")
174 I
175 I inspecting the "Z" operation group
176 I
177 I
178 I start operation group 4
179 I
180 Set_xy(6.0)
181 Move(95.545,-171.000,-44.000)
182 Move(95.545,23.573,-44.000)
183 I
184 I probing node 14 (PO 38:00 4)
185 I
186 Move(95.545,23.573,-32.958)
187 Touch(95.545,23.573,-29.958)
188 Move(95.545,23.573,-32.958)
189 I
190 Move(112.664,23.573,-32.958)
191 Touch(112.664,23.573,-29.958)
192 Move(112.664,23.573,-32.958)
193 I
194 Move(132.636,23.573,-32.958)
195 Touch(132.636,23.573,-29.958)
196 Move(132.636,23.573,-32.958)
197 Move(132.636,23.573,-41.518)
198 I
199 Hold_pt(3)
200 WkB_pin("XZ")
201 CALL Data_ft_line(3.3,Ln(_ *))
202 I
203 I probing node 6 (PO 28:00 4)
204 I
205 Move(32.776,17.867,-41.518)
206 Touch(32.776,17.867,-38.518)
207 Move(32.776,17.867,-41.518)
208 I
209 Move(44.189,17.867,-41.518)
210 Touch(44.189,17.867,-38.518)
211 Move(44.189,17.867,-41.518)
212 I
213 Move(52.748,17.867,-41.518)
214 Touch(52.748,17.867,-38.518)
215 Move(52.748,17.867,-41.518)
216 I
217 Hold_pt(3)
218 WkB_pin("XZ")
219 CALL Data_ft_line(3.4,Ln(_ *))
220 I
221 Move(52.748,17.867,-44.000)
222 Move(52.748,-171.000,-44.000)

-----

758 Move(160.000,26.146,7.775)
759 Move(160.000,-171.000,7.775)
760 I
761 I probing routine completed
762 I
763 I
764 I calculating measurements
765 I
766 Output("HP 4")
767 I
768 CALL Recall_circle(2.1,Cir(_ *))
769 CALL Recall_circle(3.2,Cir(_ *))
770 WkB_pln("XY")
771 Pt.(p.211 C° ,C° T° ,1)
772 T= R
773 WkB_pln("YZ")
774 Pt.(p.211 C° ,C° T° ,1)
775 Z=T
776 X= T
777 Sec(67)
778 Output("FS 4;")
779 Nominal("X67.000")
780 Nominal("Z31.000")
781 Pt.rol("T0.0100")
782 Output("P 2.XZ")
783 Output("P 4.77")
784 I
785 CALL Recall_circle(2.1,Cir(_ *))
786 CALL Recall_circle(4.4,Cir(_ *))
787 WkB_pln("XY")
788 Pt.(p.211 C° ,C° T° ,1)
789 T= R
790 WkB_pln("YZ")
791 Pt.(p.221 C° ,C° T° ,1)
792 Z= R
793 X= T
794 Sec(65)
795 Output("FS 4;")
796 Nominal("X67.000")
797 Nominal("Z31.000")
798 Pt.rol("T0.0100")
799 Output("P 2.XZ")
800 Output("P 4.77")
801 I
802 CALL Recall_circle(4.1,Cir(_ *))
803 WkB_pln("XZ")
804 Recall("C")
805 Sec(65)
806 Output("FS 4;")
807 Nominal("D3.000")
808 Pt.rol("D0.1000")
809 M.tol("D-0.1000")
810 Output("T 4.D")

-----

1105 I
1106 I measurements completed
1107 I
1108 Pack("X.Z.Y")
1109 I
1110 I inspection completed
1111 I
1112 SUBEND
1113 I
1114 SUB Data_ft_line(N.0 points,Register,Ln(_ *))
1115 COM /\C/Ln(_ ),Cir(_ ),Pln(_ ),Cyl(_ ),Spk(_ )
1116 Ln(No_point,1)
1117 FOR I = 0 TO 7
1118 Ln(REGISTER)= Ln(From,I)
1119 NEXT I
1120 SUBEND
1121 I
1122 SUB Recall_line(From.To,Ln(_ *))
1123 COM /\C/Ln(_ ),Cir(_ ),Pln(_ ),Cyl(_ ),Spk(_ )
1124 FOR I = 0 TO 7
1125 Ln(To,I)= Ln(From,I)
1126 NEXT I
1127 SUBEND
1128 I
1129 SUB Data_ft_circle(No_points,Register,Cir(_ *))
1130 COM /\C/Ln(_ ),Cir(_ ),Pln(_ ),Cyl(_ ),Spk(_ )
1131 Circle(No_points,1)
1132 FOR I = 0 TO 4
1133 Cir.(Register)= Cir(1.1)
1134 NEXT I
1135 SUBEND
1136 I
1137 SUB Recall_circle(From.To,Cir(_ *))
1138 COM /CIV Ln(*),Cir(*),Pin(*),Cyl(*),Sph(*)
1139 FOR I= 0 TO 4
1140 Cir(To.I) = Cir(From.I)
1141 NEXT I
1142 SUBEND
1143 1
1144 SUB Data_fit_cyl(No_points,Register,Cyl(*))
1145 COM /CIV Ln(*),Cir(*),Pin(*),Cyl(*),Sph(*)
1146 Cy(No_points,No_points,1)
1147 FOR I= 0 TO 7
1148 Cy(1,Register,1) = Cy(1,1)
1149 NEXT I
1150 SUBEND
1151 1
1152 SUB Recall_cyl(From.To,Cyl(*))
1153 COM /CIV Ln(*),Cir(*),Pin(*),Cyl(*),Sph(*)
1154 FOR I= 0 TO 7
1155 Cy(To.I) = Cy(From.I)
1156 NEXT I
1157 SUBEND
D.7 The Part Program for the second Setup in the Ferranti part programming language

1    Part:SUB Part
2    1
3    3 program to inspect 't' setup of tool Zpos prog
4    1 on 05-JUL-1990 at 18:23
5    1
6    COM /C3/ X,Y,Z,R,A,D,D2,Tpm,Ferm,Prn(),Doa(*)N
7    COM /C3/ Msta(),Wsta(),Rsta(),Spf(*)
8    COM /C3/ Pst(),Tps(),Pst()N
9    COM /C3/ Pst(),Tps(),Pst()N
10   COM /C3/ Ln( ),Lin( ),Cyl( ),Sph( )
12   COM /C14/ R_3d( ),R_2d( ),W( )
13   COM /Rex/ RstL(),Sm
14   Partdir "ON" ! For geometric tolerance printout
15   ALLOCATE Ln.(4,7)
16   ALLOCATE Cir.(3,4)
17   ALLOCATE Cyl(1,7)
18   Menu
19   Manual
20   Rot_ch
21   Prt_set("Prb")
22   Seq(1)
23   Speed(125.0)
24   Prpdp(10.0)
25   Prt(0.100)
26   Sol_tip(2.0)
27   Display("MOVE PROBE -6MM FROM DATUM")
28   Beep
29   Wait
30   Wkt_pln("XY")
31   Ret_x(0.000)
32   Load
33   Master("X,Y,Z")
34   Datum ("T Y.000")
35   Datum ("T X:109.000")
36   Datum ("T Y.000")
37   Datum ("T Z.000")
38   1
39   1 initialization complete
40   1
41   1 start operation group 9
42   1
43   1
44   1
45   1
46   1
47   1
48   1
49   1
50   1
51   1
52   1
53   1
54   1
55   1
56   1
57   1
58   1
59   1
60   1
61   1
62   1
63   1
64   1
65   1
66   1
67   1
68   1
69   1
70   1
71   1
72   1
73   1
74   1
75   1
76   Move(106.616,-6.000,159.000)
77   Wkt_pln("XY")
78   Ret_x(0.000)
79   CALL Recall cil(1,1),Cyl( )
80   Level("C 1")
81   Wkt_pln("XY")
82   Ret_x(0.000)
83   1
84   1 start operation group 10
85   1
86   1
87   Move(95.545,23.573,159.000)
88   Move(95.545,23.573,44.000)
89   1
90   1 probing node 9 (PO 20G 10)
91   1
92   Move(95.545,23.573,35.958)
93   Touch(95.545,23.573,29.958)
94   Move(95.545,23.573,35.958)
95   1
96   Move(112.664,23.573,35.958)
97   Touch(112.664,23.573,29.958)
98   Move(112.664,23.573,35.958)
99   1
100  Move(132.636,23.573,35.958)
101  Touch(132.636,23.573,29.958)
102  Move(132.636,23.573,35.958)
103  1
104  1
105  1
106  1
107  1
108  1
109  1
110  1
111  1
112  1 start operation group 11
113  1
114  1
115  Move(106.616,-6.000,159.000)
116  Move(106.616,-6.000,7.642)
117  1
118  1 probing node 7 (PO 30G 11)
119  1
120  Move(106.616,14.308,7.642)
121  Touch(104.105,14.308,15.692)
122  Move(106.616,14.308,7.642)
123  1
124  Move(110.820,14.308,7.795)
125  Touch(112.664,14.308,15.692)
126  Move(110.820,14.308,7.795)
127  1
128  Move(116.701,14.308,2.186)
129  Touch(124.077,14.308,4.279)
130  Move(116.701,14.308,2.186)
131  1
132  Move(110.820,14.308,7.795)
133  Touch(112.664,14.308,15.692)
134  Move(110.820,14.308,7.795)
135  1
136  Move(106.616,14.308,7.642)
137  Touch(104.105,14.308,15.692)
138  Move(106.616,14.308,7.642)
139  Move(106.616,-5.252,-7.642)
140  1
141  1
142  1
143  1
144  1
145  1 probing node 4 (PO 40G 11)
146  1
147  Move(49.896,-5.252,1.427)
148  Touch(49.896,0.742,1.427)
149  Move(49.896,-5.252,1.427)
150  1

Appendix D - A Users Guide
Appendix D - A Users Guide

151 Move(75.573,-5.252,1.427)
152 Touch(75.573,0.748,1.427)
153 Move(75.573,-5.252,1.427)
154  
155 Move(89.839,-5.252,1.427)
156 Touch(89.839,0.748,1.427)
157 Move(89.839,-5.252,1.427)
158  
159 Hold_part(3)
160 Whg_plot("XY")
161 CALL Data_51_line(5.2,1.427)
162 Move(89.839,-6.000,1.427)
163 Move(89.839,-6.000,159.000)
164 CALL Recall_circle(1.0,1.3,0.000)
165 Recall("C")
166 Mutate("Z")
167 Mutate("X")
168 CALL Recall_line(2.4,.000)
169 Recall("L")
170 Mutate("Y")
171 Datum(3.2,0.000)
172 Datum(1.1,0.000)
173 Datum(1.2,0.000)
174  
175 Inspecting the '+'Y' operation group
176  
177  
178  
179  
180 Sel_tip(2.0)
181 Move(105.723,-6.000,159.000)
182 Move(105.723,-6.000,10.000)
183  
184  
185 Move(105.723,14.308,10.000)
186 Move(105.723,14.308,15.692)
187 Touch(104.105.14.308,15.692)
188 Move(105.723,14.308,10.000)
189  
190 Move(111.502,14.308,10.000)
191 Touch(112.664,14.308,15.692)
192 Move(111.502,14.308,10.000)
193  
194 Move(119.587,14.308,3.000)
195 Touch(124.077,14.308,4.279)
196 Move(119.587,14.308,3.000)
197  
198 Move(111.502,14.308,-10.717)
199 Touch(112.664,14.308,-15.692)
200 Move(111.502,14.308,-10.717)
201  
202 Move(105.723,14.308,-10.506)
203 Move(105.723,14.308,-15.692)
204 Move(105.723,14.308,-10.506)
205 Move(105.723,159.225,10.506)
206  
207 Hold_part(3)
208 Whg_plot("XZ")
209 CALL Data_51_line(3.2,4.000)
210  
211  
212  
213 Move(49.896,-2.252,1.427)
214 Touch(49.896,0.748,1.427)
215 Move(49.896,-2.252,1.427)
216  
217 Move(75.573,-2.252,1.427)
218 Touch(75.573,0.748,1.427)
219 Move(75.573,-2.252,1.427)
220  
221 Move(89.839,-2.252,1.427)
222 Touch(89.839,0.748,1.427)
223 Move(89.839,-2.252,1.427)
224  
225 Hold_part(3)
226 Whg_plot("XY")
227 CALL Data_51_line(3.2,1.427)
228  
229 Move(89.839,-6.000,1.427)
230 Move(89.839,-6.000,159.000)
231  
232  
233  
234  
235 Inspecting the '-'Y' operation group
236  
237  
238  
239  
240  
241  
242  
243 Move(99.089,37.839,1.334)
244 Touch(98.398,37.839,1.427)
245 Move(99.089,37.839,1.334)
246  
247 Move(118.933,37.839,1.159)
248 Touch(121.223,37.839,1.427)
249 Move(118.933,37.839,1.159)
250  
251 Move(118.933,37.839,-1.159)
252 Touch(121.223,37.839,-1.147)
253 Move(118.933,37.839,-1.159)
254  
255 Move(99.089,37.839,-0.942)
256 Touch(98.398,37.839,-1.427)
257 Move(118.933,37.839,-0.942)
258  
259 Move(99.089,37.839,-1.334)
260 Touch(98.398,37.839,-1.427)
261 Move(99.089,37.839,-1.334)
262 Move(99.089,52.552,-1.334)
263  
264 Hold_part(3)
265 Whg_plot("XZ")
266 CALL Data_51_line(5.3,1.427)
267  
268  
269  
270 Move(35.630,52.552,1.427)
271 Touch(35.630,49.225,1.427)
272 Move(35.630,52.552,1.427)
273  
274 Move(67.014,52.552,1.427)
275 Touch(67.014,49.225,1.427)
276 Move(67.014,52.552,1.427)
277  
278 Move(92.692,52.552,4.279)
279 Touch(92.692,49.225,4.279)
280 Move(92.692,52.552,4.279)
281  
282  
283 CALL Data_51_line(3.4.1.427)
284  
285  
286 Move(92.692,56.000,4.279)
287 Move(92.692,56.000,159.000)
288  
289  
290  
291  
292  
293  
294 Output("HP 4,")
295  
296 CALL Recall_circle(3.1,1.427)
297 CALL Recall_circle(2.3,1.427)
298 Seq(37)
299 Output("PS 4,")
300 Conc_hy(1.2,0.010)
301  
302 CALL Recall_circle(3.1,1.427)
303 Whg_plot("XZ")
304 Recall("C")
305 Seq(30)
306 Output("PS 4,")
307 Nominal("D.26.00\)
308 P_sol("D.10.00")
309 M_sol("D.0.01")
310 Output("P.4.D")
311  
312 CALL Recall_line(3.1,1.427)
313 CALL Recall_line(4,2,La_(*))
314 Wg_pic("XY")
315 La_in2(L,1,L,2,1)
316 Seg(17)
317 Output("PS 6:*
318 Nominal("$35.000")
319 P_tol("R6.1000")
320 M_tol("K 0.1000")
321 Output("P 4:R")
322 !
323 ! measurements completed
324 !
325 Park("Z,X,Y")
326 !
327 ! inspection completed
328 !
329 SUBEND
330 !
331 SUB Data_fit_line(No_point,Register,La_(*))
332 COM / Cyl/Ln(*),Cyl(*),Pin(*),Cyl(*),Sph(*)
333 Line(No_point,1)
334 FOR I= 0 TO 7
335 La._(Register,I)= Ln(1,I)
336 NEXT I
337 SUBEND
338 !
339 SUB Recall_line(From,To,Ln_(*))
340 COM / Cyl/Ln(*),Cyl(*),Pin(*),Cyl(*),Sph(*)
341 FOR I= 0 TO 7
342 Ln(To,I)= Ln_(From,I)
343 NEXT I
344 SUBEND
345 !
346 SUB Data_fit_circle(No_point,Register,Cyl_(*))
347 COM / CYL/Ln(*),Cyl(*),Pin(*),Cyl(*),Sph(*)
348 Circle(No_point,1)
349 FOR I= 0 TO 4
350 Cyl_(Register,I)= Cyl(1,I)
351 NEXT I
352 SUBEND
353 !
354 SUB Recall_circle(From,To,Cyl_(*))
355 COM / CYL/Ln(*),Cyl(*),Pin(*),Cyl(*),Sph(*)
356 FOR I= 0 TO 4
357 Cyl(To,I)= Cyl_(From,I)
358 NEXT I
359 SUBEND
360 !
361 SUB Data_fit_cyl(No_point,Register,Cyl_(*))
362 COM / CYL/Ln(*),Cyl(*),Pin(*),Cyl(*),Sph(*)
363 Cyl(No_point,1)
364 FOR I= 0 TO 7
365 Cyl_(Register,I)= Cyl(1,I)
366 NEXT I
367 SUBEND
368 !
369 SUB Recall_cyl(From,To,Cyl_(*))
370 COM / CYL/Ln(*),Cyl(*),Pin(*),Cyl(*),Sph(*)
371 FOR I= 0 TO 7
372 Cyl(To,I)= Cyl_(From,I)
373 NEXT I
374 SUBEND
Appendix E:

Glossary

Axis System

Mutually orthogonal system of axes constructed in Coordinate Measurement to establish the true 3D orientation of the part. Also known as a Reference Frame.

Axis System and Datum Setting Operation

An operation specified by the Inspection Planner that describes how the Axis System and Datum should be set up on a CMM.

Component Geometry

A graph-type structure that describes the features and consequently the CSG half-spaces that are used to define a component.

Component Setup Planner

Inspection Planning process that is concerned with the determination all of information required to set up a component for the inspection of a known set of constraints on a Coordinate Measuring Machine (CMM).

Constraint

Dimension or geometrical tolerance as defined in BS308 or ANSI Y14.5b. Also known as a Relationship.

Datum

The origin of an Axis System.
Design to Manufacture Environment

A collection of design and manufacture activities which include design, Machine Operations Planning and Cutter Path Generation, Inspection Plan and Code Generation and Manufacturing Data Analysis.

Inspection Plan

Generic data structure that describes the entire inspection process for a particular part on a particular type of measuring machine.

Inspection Planner

A computer-based tool that plans the inspection process, which includes operation types, component setups, probe setups, probing points and probing paths.

Measurement Method

Method specified by the Operation Type Planner that describes how a Relationship should be measured.

Measuring Algorithm

CMM controller-based software that allows surface representations to be manipulated so that actual relationship values can be calculated.

Measuring Operation

An operation that describes how a constraint should be measured. It is essentially a record that includes a pointer to the parent constraint, a Measuring Algorithm type and a list of Probing Operations.
Offset Point
A coordinate point which lies on the surface normal at the Probing Distance from the Probing Point. The Offset Point is the point that the probes traverse to at full speed before slowing down to touch the surface.

Operation Group
A list of Probing Operations that have the same principal Probe Approach Direction.

Operation Type Planner
Inspection Machine Planning activity which determines the types of all the operations that need to be carried out in order to inspect a part on a Coordinate Measuring Machine (CMM).

Principal Probe Approach Direction (PAD)
The approach direction which allows the probe to access the greatest proportion of the surface.

Probe Approach Direction (PAD)
A direction along which a probe can safely approach all or part of a surface.

Probe Approach Direction Generator
Inspection Planning process that generates and validates the six major Probe Approach Directions for each surface to be inspected.

Probe Path Planner
Inspection Planning process that is concerned with the generation of collision-free paths for the probe to follow when inspecting a workpiece.
Probe Setup Planner

Inspection Planning process that is concerned with the identification of both the probe configuration that should be used to inspect each surface and the orientation of the motorized probe head.

Probing Distance

The distance that the probe traverses from the Offset Point before touching the surface at the Probing Point. The Probing Distance is set by the Inspection Planner according to the expected positional accuracy of the surface.

Probing Operation

An operation that describes how a surface should be probed. It is essentially a record that includes a Surface Representation type, the working plane (when appropriate) and a list of pointers to the Master Probing Operation, the Master Surface Representation, the surface to be measured and the Sub-Operations.

Probing Path

A collision-free path that the probe follows when traversing between Offset Points. This path is described by a list of coordinate points.

Probing Point

The coordinate point on the surface of the object which the probe traverse towards when inspecting the surface.

Probing Point Planner

Inspection Planning process that is concerned with the generation of Probing Point coordinates for each surface that requires measurement.
Product Model

An instance of the Product Model Data Structure that describes a product.

Product Model Data Structure

A data structure researched by the project which represents electro-mechanical products for the Design to Manufacture Environment of the project.

Product Modelling System

A data management environment that is specifically designed to represent and maintain the Product Model Data Structure defined by the project.

Relationship

Dimension or geometrical tolerance as defined in BS308 or ANSI Y14.5b. Also known as a Constraint.

Setup

An element of the Inspection Plan that describes how the component should be set-up for inspection. This information typically consists of a component orientation, a list of constraints to be measured, the fixture necessary to hold the object and a list of Operation Groups and the corresponding Measuring Operations.

Surface Representation

A mathematical representation of a surface as a simple geometric entity such as a point, line, plane, circle, cylinder, cone or sphere.

Working Plane

A plane specified during the coordinate measuring process on to which two-
dimensional Surface Representations are projected for measurement purposes.
A Historical Summary of Geometric Modelling

**Interactive Computer Graphics**
- "Invented" (Projective) Geometry
- NC Programming Languages

**Wireframes**
- 1955-64
- 2-D Systems based on drafting principles
- Early NC from graphic databases

**Polygonal Schemes**
- 1965-72
- Early hidden-line and visible-surface algorithms for polygonal faces
- Simulators

**Sculptured Surfaces**
- 1973-78
- Early hidden-line and visible-surface algorithms for polygonal faces
- Simulators
- Better algorithms
- Polyhedral smoothing
- Faster simulators
- 3-D animation
- NC contour milling
- Lofted/digitized surfaces
- B-Spline curves and surfaces
- B-Spline subdivision algorithms

**Solid Modelling**
- 1979-84
- Ad hoc experiments using diverse approaches
- Experimental boundary, CSG and sweep-based systems
- Theoretical foundations emerge
- Development of industrial prototypes
- Early production versions
Wireframe Model

Possible Solid Object

Ambiguous Wireframe Models

Fig. 2.2
Non-Sense Wireframe Solid

Fig. 2.3
Physical Object

Wireframe Representation

Visualisation Problems of the Wireframe Representation

Fig. 2.4
The Physical Object

The CSG Representation

The Boundary Representation

CSG and Boundary Representation of a Solid

Fig. 2.5
The Physical Object

Pure Swept Representation

Hybrid Sweep with "Gluing" Facility

Pure and Hybrid Sweep Representation
Examples of Conventional and Geometrical Methods of Constraint
Effects of Dimensional Changes on Geometry

Fig. 2.8
The Product Modelling Environment of the University of Tokyo

Fig. 2.9
The Layer Syntax of the Berlin University Product Model

a) The Layer Syntax

b) The Product Model
Sequential Progression in the Design Process
Part Surface $Z = f(x,y)$

Tool Path on Surface

Tool Path Planned on $xy$-plane

The Cartesian Machining Method

Fig. 3.2
Probe Path Planning by Kawabe et al

Fig. 4.1

a) The probe path selected by the operator
b) Collision checking using a swept volume
c) A collision detected during probe movement
d) A collision-free probe path
The "Generate-and-Test" Approach to Probe Path Planning

Fig. 4.2
Probing Point Distribution on a Bicubic Patch

Fig. 4.3

a) Surface Point Distribution

b) Examples of Probe Movement Points
Feature Interaction

Fig. 4.4
"Blackboard" Architecture

Fig. 4.5
a) A Straight Probe and the Corresponding Abstraction

b) A Direction Cone
a) Machine Axis System and Datum

b) Component Axis System and Datum

The Component Axis System

Fig. 5.1
a) the constraint to be measured

b) the Probing Operations

c) The Measuring Operation

Measuring and Probing Operations  Fig. 5.2
The Motorised Probe Head

Fig. 5.3
Motorized Head

Styli and Ball Variations

Probe Body Extensions

Probe Configurations
a) point distribution on a plane

b) point distribution on a circle

Probing Point Distribution

Fig. 5.5
The Manufacturing Cell

Fig. 5.6
a) test component used by the QTC project

b) test component used by the ISS project
The Design to Manufacture Environment

Fig. 6.2
The Framework

Fig. 6.3
The Root

Fig. 6.4
The Assembly

Fig. 6.5
The Component

Fig. 6.6
The First Inspection Planning Case—Study

**Fig. 8.1**
The Bolster-Plate Case-Study

Fig. 8.2
a) Structural Bearing Case–Study

b) Wrist Motor Body Case–Study
The Inspection Plan Data Structure

Fig. 8.4
The Measuring Operation Data Structure Fig. 8.6
The Measuring Operation Types Data Structures

Fig. 8.7
The Operation Group and Probing Operation Data Structures

Fig. 8.8
The Inspection Planner Relationship Graph

Fig. 8.9
The Inspection Planner Relationship Types Fig. 8.10
The Working Plane

Fig. 9.1

a) Correct 'Z' Working Plane

b) Incorrect 'X' Working Plane
a) The Constraint

b) The One-Dimensional Method
a) The Two-Dimensional Method

b) The Three-Dimensional Method

Fig. 9.3
a) the geometrical tolerance of flatness

b) the geometrical tolerance of squareness

c) a linear dimension
The Points and Vectors Involved in Probing a Point on a Surface

Fig. 10.1
The Sequence of Moves Involved in Probing a Point on a Surface

Fig. 10.2

a) traverse to Offset Point

b) probe surface

c) retract to Offset Point

d) exit via approach direction
a) The Probing Operation

b) The Probe Movement Envelope

Fig. 10.3
a) Surface and PAD under analysis

b) The Total Probe Movement Envelope required to probe every point on the surface
a) The Probe Approach Directions for a surface

b) Component Axis System
a) TPME for a planar PAD

b) TPME for a normal PAD

TPMEs for Planar Surfaces

Fig. 10.6
a) Negative Cylindrical Surface

b) Positive Cylindrical Surface

TPMEs for Cylindrical Surfaces

Fig. 10.7
a) typical non-regular planar surface

b) exaggerated non-regular surface and the constructed rectangle used for analysis

A Planar Surface with a Non-Regular Boundary

Fig. 10.8
a) Planar PAD on a planar surface

b) Axial PAD on a cylindrical surface
a) Planar Surface

b) Cylindrical Surface

c) Detection Method

Examples of Intersecting Surfaces and a Method of Detection
a) The Probe Movements to be modelled

b) The approximate model of the Probe Movement Envelope
a) collection of surfaces representing a PME

b) matrix of cell centre-points representing each PME surface

c) PME constructed from SDSM
r = Probe Body Radius

a) Known components of the PME

b) Calculation of $l_1$ and $l_2$

Known components of the Probe Movement Envelope (PME)
a) Surface Vertices of the PME

b) Additional constructed points and vectors
a) special case where the Surface Normal is equal but opposite to the Probe Approach Direction

b) corresponding special case PME
a) Planar Surface/Coplanar PAD

b) Cylindrical Surface/Axial PAD
a) Planar surface with coplanar PAD

b) Complete PME for top cell

c) Incremental PME for subsequent cells
a) angled bore inspected with respect to machine axis system

b) angled bore inspected with respect to local coordinate system
The 6 Possible Component Orientations  

a) +X Component Orientation  
b) -X Component Orientation  
c) +Y Component Orientation  
d) -Y Component Orientation  
e) +Z Component Orientation  
f) -Z Component Orientation
The Fixturing Surfaces of a Component  

Fig. 11.2
a) orthogonal view of component

b) stable position of component when unsupported

c) side view showing component supported by simple fixture
a) Block-like Component

b) Plate-like Component

c) Rod-like Component
Stylus Tip Types

a) Sphere
b) Disc
c) Star
d) Cylinder
e) Point

Fig. 12.1
a) probing deep bores

b) probing undercuts and grooves
a) when probing a deep bore

b) when probing a misaligned bore (exaggerated)
a) Cylinder Stylus Applications

b) Point Stylus Applications
The Use of Probe Body Extension Bars

Fig. 12.5
The orientation limits of the Motorised Probe Head.
Regularly Bounded Surfaces on the Wrist Motor Body Case—Study
a) Point Template

b) Line Templates
The Plane Surface Representations

Fig. 13.3
\[ \theta = \frac{2\pi}{n} \]

\[ n=4 \quad \text{and} \quad n=5 \]

a) Closed Cylindrical Surfaces

\[ \theta = \frac{\alpha}{n} \]

\[ n=4 \quad \text{and} \quad n=5 \]

b) Open Cylindrical Surfaces

The Circle Surface Representation  Fig. 13.4
The Cylinder, Cone and Sphere Surface Representations

- **a)** The Cylinder Surface Representation

- **b)** The Cone Surface Representation

- **c)** The Sphere Surface Representation
Examples of the determination of regular boundaries

Fig. 13.6
Theoretical Method of Point Distribution

Fig. 13.7

a) surface under analysis

b) placement of "squashy balls" within the surface bounds

c) centroids of 24 "squashy balls"
a) the surface under analysis

b) the corresponding regular surface boundary

Point Distribution using the Point Density Method

Fig. 13.8
a) the superimposed point grid

b) the resultant point distribution
The Two Types of Probing Path

Fig. 14.1

a) Between an Offset Point and a Probing Point

b) Between Offset Points
Examples of Probe Paths Generated by the Simple Method

Fig. 14.2
a) The Probe Path

b) The Probe Movement Envelope

The PME Required to Model a Probe Path  Fig. 14.3
a) collection of surfaces representing a PME

b) matrix of cell centre-points representing each PME surface

c) PME constructed from SDSM
Fig. 14.5

Approach Probe Direction

Simplifications to the PME Method Applied to Probe Path Planning

a) Orthogonal View

b) Plan View

c) Orthogonal View
d) Plan View
Simplifications to the PME Method
Applied to Probe Path Planning

Fig. 14.6
a) 2d complex planar surface

b) spatial decomposition of surface

c) shortest path chosen by Heuristic Method
a) Path Derived by the Heuristic Approach

b) Path Shortened by the Trigonometric Method

Shortening the Path Found by the Heuristic Approach

Fig. 14.8
The DMIS Environment

Fig. 15.1
a) Boss Form Feature

b) Through-Hole Form Feature

c) Blind-Hole Form Feature
Form Features

Main Boss

Upper Hole

Main Hole

Section B-B

Upper Right Hole

Lower Left Hole

Tapped Hole

Section A-A

Upper Right Hole

Lower Right Hole

Fig. C.3
Relationships to be inspected

Fig. C.4
Complex Form Features

Fig. C.5
a) Plan View

b) Side View

The CSG Axis System and the Centre of Gravity

Fig. D.2
a) the original point locations

b) the repositioned points
a) Intersecting Surfaces

b) Isolated Surfaces

"Worst-Case" Surface Intersection
The results of surface intersection analysis

Fig. D.5