Optimisation of Surface Coverage Paths used by a Non-Contact Robot Painting System

Finlay Neil McPherson

Thesis submitted for the degree of

Doctor of Philosophy

To Heriot-Watt University, Edinburgh, UK

on completion of research in the

School of Engineering and Physical Science

August 2011

Abstract

This thesis proposes an efficient path planning technique for a non-contact optical "painting" system that produces surface images by moving a robot mounted laser across objects covered in photographic emulsion. In comparison to traditional 3D planning approaches (e.g. laminar slicing) the proposed algorithm dramatically reduces the overall path length by optimizing (i.e. minimizing) the amounts of movement between robot configurations required to position and orientate the laser.

To do this the pixels of the image (i.e. points on the surface of the object) are sequenced using configuration space rather than Cartesian space. This technique extracts data from a CAD model and then calculates the configuration that the five degrees of freedom system needs to assume to expose individual pixels on the surface. The system then uses a closest point analysis on all the major joints to sequence the points and create an efficient path plan for the component.

The implementation and testing of the algorithm demonstrates that sequencing points using a configuration based method tends to produce significantly shorter paths than other approaches to the sequencing problem. The path planner was tested with components ranging from simple to complex and the paths generated demonstrated both the versatility and feasibility of the approach.

Acknowledgement

I would first of all like to thank my supervisor Dr Jonathan Corney for his continued support, guidance and assistance throughout my PhD and for his trust in my capabilities. Thanks also go to Dr Raymond Sung whose suggestions and comments have been both helpful and encouraging throughout my research.

I would also like to thank Dr Theodore Lim for his help, enthusiasm and continued assistance on robotic matters and for inspiring me. I would like to thank all my colleagues and friends at the University for their encouraging words and continued support throughout my studies.

Finally, and personally, I would like to thank my parents and brother for giving me the opportunity to complete my studies and aiding me in so many ways, especially during the final months where they have proof read with increasing confusion as to the meaning of the words.

ACADEMIC REGISTRY

Research Thesis Submission

Name:	FINLAY NEIL MCPHERSON		
School/PGI:	MECHANICAL ENGINEERING / EPS		
Version: *(i.e. First, Resubmission, Final)*	Final	Degree Sought:	PhD

Declaration

In accordance with the appropriate regulations I hereby submit my thesis and I declare that:

1. the thesis embodies the results of my own work and has been composed by myself
2. where appropriate, I have made acknowledgement of the work of others and have made reference to work carried out in collaboration with other persons
3. the thesis is the correct version of the thesis for submission and is the same version as any electronic versions submitted*.
4. my thesis for the award referred to, deposited in the Heriot-Watt University Library, should be made available for loan or photocopying and be available via the Institutional Repository, subject to such conditions as the Librarian may require
5. I understand that as a student of the University I am required to abide by the Regulations of the University and to conform to its discipline.

* *Please note that it is the responsibility of the candidate to ensure that the correct version of the thesis is submitted.*

Signature of Candidate:		Date:	

Submission

Submitted By (name in capitals):	Finlay Neil McPherson
Signature of Individual Submitting:	
Date Submitted:	

For Completion in Academic Registry

Received in the Academic Registry by *(name in capitals):*			
Method of Submission *(Handed in to Academic Registry; posted through internal/external mail):*			
E-thesis Submitted			
Signature:		Date:	

Table of Contents

ABSTRACT .... **II**

ACKNOWLEDGEMENT .... **III**

LIST OF FIGURES .... **IX**

LIST OF TABLES .... **XIII**

LIST OF GRAPHS .... **XIV**

THESIS RELATED PUBLICATIONS .... **XVII**

CHAPTER 1 INTRODUCTION **1**

1.1 SURFACE COVERAGE IN OPTICAL NON-CONTACT PAINTING 1
1.2 SURFACE COVERAGE IN MACHINING 4
1.3 SURFACE COVERAGE IN CONTACT SCANNING WITH A COORDINATE MEASURING MACHINE (CMM)6
1.4 SURFACE COVERAGE FOR NON-CONTACT SCANNING 9
1.5 SURFACE COVERAGE IN CONTACT PAINTING (I.E. SPRAY PAINTING) 13
1.6 CONCLUSION 15
1.7 AIMS AND OBJECTIVES 17
1.8 THESIS OUTLINE 17

CHAPTER 2 LITERATURE REVIEW **19**

2.1 PATH PLANNING 19
2.1.1 Cartesian Path Planning 20
2.1.2 Parametric Path Planning 24
2.1.3 Configuration Space Based Planning 27
2.1.4 Path Planning Conclusion 34
2.2 REVIEW OF TECHNOLOGIES FOR PLANNING MASS CUSTOMIZATION OF PRODUCT APPEARANCE 35
2.2.1 Introduction 35
2.2.2 Mass Customization by Component Selection 36
2.2.3 Mass Customization by User Innovation 38
2.2.4 Mass Customization by User Interface 40
2.2.5 Mass Customization by Geometric and Function 41
2.2.6 Mass Customization Requirement for Optical Painting 41
2.3 SHAPE AND SURFACE APPEARANCE 43
2.3.1 Introduction to Virtual Surface Mapping 43
2.3.2 Surface Representation Overview 44
2.3.3 Overview of Texture Maps 45
2.3.4 Two-Dimensional Approach 46
2.3.5 Three-Dimensional Approach 48
2.3.6 Physical Model Surface Appearance 49
2.2.7 Photography 50

2.3.8 Conclusion *51*

CHAPTER 3 THE OPTICAL PAINTING SYSTEM **53**

3.1 HARDWARE OVERVIEW 54
3.1.1 Robot Cell Controller Setup *56*
3.2 PATH PLANNING SOFTWARE 57
3.3 PROTOTYPING SOFTWARE ARCHITECTURE 60
3.3.1 Data Collection Module *61*
3.3.2 Data Processing Module *63*
3.4 SYSTEM PERFORMANCE 64
3.5 SYSTEM CONCLUSION 65

CHAPTER 4 SURFACE POINT CLOUD GENERATION **66**

4.1 IMAGE QUALITY ASSESSMENT 66
4.2 IMAGE QUALITY: DOT PATTERNS 68
4.3 POINT EXTRACTION ALGORITHM 70
4.3.1 Q-Algorithm *70*
4.3.2 Results *73*
4.3.3 Discussion *75*
4.4 CONCLUSION 76

CHAPTER 5 POINTS SEQUENCING **77**

5.1 POINT SEQUENCING ALGORITHM 77
5.2 POINT SEQUENCING APPROACHES 78
5.3 COORDINATE SYSTEMS 83
5.4 DENAVIT-HARTENBERG CONVENTION (DH) 84
5.4.1 Generalised Transformation Matrix for DH Approach *85*
5.4.2 Direct Writing System DH Analysis *86*
5.5 INVERSE KINEMATICS 92
5.5.1 Inverse Kinematic Solution for Rotary Tilt Table *93*
5.5.2 Inverse Kinematic Solution to SCARA Robot *95*
5.6 REDUNDANCY 96
5.7 DEGENERACY 97

CHAPTER 6 IMPLEMENTATION AND RESULTS **98**

6.1 IMPLEMENTATION 98
6.2 RESULTS 100
6.2.1 Validation 1: Cylinder Test *103*
6.2.2 Validation 2: Deformed Cylinder *116*
6.2.3 Eggcup Component *127*
6.2.4 Penguin Component *134*
6.2.5 Wizard Component *141*

6.3 DISCUSSION OF RESULTS 150

CHAPTER 7 CONCLUSION 153

7.1 AIMS AND OBJECTIVES 153
7.2 LIMITATIONS 154
7.3 FURTHER WORK 155
7.4 RESEARCH CONTRIBUTION 156

REFERENCES 158

APPENDIX 1 169

TONING AND UNIFORM PATTERNS 169

List of Figures

FIGURE 1-1 EXPERIMENTAL OPTICAL PAINTING SYSTEM 2

FIGURE 1-2 COMPLEX MODEL WITH LINEAR SLICE PRODUCES SIGNIFICANTLY DIFFERENT NORMAL VECTORS........ 3

FIGURE 1-3 COMBINATION TOOL PATH OPTIMIZED TO ACHIEVE UNIFORM CUSP HEIGHT (6)5

FIGURE 1-4 CONVENTIONAL CMM MACHINE AND PROBE HEAD (10)........ 6

FIGURE 1-5 HELICAL SCAN FOR HOLLOW COMPONENTS (13) 7

FIGURE 1-6 BLADE SHARP EDGE SWEEP SCAN (13)........ 8

FIGURE 1-7 ROUND THE BLADE SCAN (13) 9

FIGURE 1-8 CARL ZEISS LINE SCANNER FOR CAPTURING SURFACES USING POINT CLOUDS (14)........ 10

FIGURE 1-9 SCHEMATIC DIAGRAM OF A TIME OF FLIGHT OPTICAL SCANNER (15)........ 11

FIGURE 2-1 A) LINEAR CARTESIAN PATH (27) B) PARAMETRIC SURFACE PATH (28) C) CONFIGURATION SPACE OPTIMIZED PATH (29)........ 19

FIGURE 2-2 A) SINGLE DIRECTION RASTER PATH B) RASTER CUTTING PATH C) OFFSET CUTTING PATH (6) 20

FIGURE 2-3 FRACTURAL TOOL PATH USING HILBERT CURVE (33)........ 21

FIGURE 2-4 A) LOW CURVATURE PATCH B) HIGH CURVATURE PATCH C) COMBINATION OF SEPARATE PATHS (43)........ 25

FIGURE 2-5 TAXONOMY OF C-SPACE REPRESENTATION SCHEMES (52)........ 27

FIGURE 2-6 MINKOWSKI SUM OPERATORS (55) 28

FIGURE 2-7 EXTENSION TO THE MINKOWSKI SUM USING ROTATION (55) 29

FIGURE 2-8 QUAD-TREE REPRESENTATION FOR CELL SUB-DIVISION (62) 31

FIGURE 2-9 CELLULAR DECOMPOSITION OF WORKSPACE WITH SINGLE OBSTACLE (65) 32

FIGURE 2-10 A) ROBOTIC REACH AND POINT OBSTRUCTIONS B) JOINT ANGLES WHERE POINT OBSTRUCTIONS EXISTS (66)........ 33

FIGURE 2-11 DEGREES OF CUSTOMIZATION (69)........ 36

FIGURE 2-12 PERSONALISED RUGBY BOOTS FROM ADIDAS 39

FIGURE 2-13 PROJECTION MAPPING ALONG VARIOUS AXES (TOP IMAGE MAPS ALONG Z-AXIS ON THE COMPONENT’S SURFACE, BOTTOM LEFT MAPS ALONG X-AXIS, BOTTOM RIGHT MAPS ALONG Y-AXIS) 46

FIGURE 2-14 VARIOUS MAPPING TECHNIQUES USED (151) *NOTE THE USE OF CGI GRAPHICS FORMAT X-Y SCREEN POSITION AND Z REPRESENTING DEPTH OF PERCEPTION. 47

FIGURE 2-15 WOOD GRAIN MAPPING USING BOUNDING BOX TECHNIQUE (97) 48

FIGURE 2-16 3D MAPPING TECHNIQUES (142) 49

FIGURE 3-1 ORIGINAL HARDWARE CONFIGURATION 53

FIGURE 3-2 NEW HARDWARE CONFIGURATION 55

FIGURE 3-3 ROBOTIC JOINTS 55

FIGURE 3-4 EXAMPLE POINT SET OF TWO THOUSAND POINTS 58

FIGURE 3-5 ORIENTATION CONTROLS THE EXPOSURE AREA FOR CONSISTENT POINTS THE ORIENTATION SHOULD REMAIN AS CLOSE TO THE NORMAL AS POSSIBLE THUS CREATING A UNIFORMLY CIRCULAR SPOT PROFILE 59

FIGURE 3-6 SYSTEM ARCHITECTURE AND DATA FLOW 60

FIGURE 3-7 GENUS OF AN OBJECT...62

FIGURE 4-1 LINEAR AND SPATIAL POINT SAMPLING APPROACHES, LINEAR CAN BE SEEN TO BE REGULAR OVER ONE FACE MODEL ‘A’ WHEREAS SPATIAL PROVIDES UNIFORM SAMPLING ACROSS MODEL ‘B’. 68

FIGURE 4-2 SPATIAL SAMPLING USING VOXELS AND SPATIAL SAMPLING USING SPATIAL CURVE (127)..69

FIGURE 4-3 PEANO CURVE 1ST TO 3RD ORDER CURVE (130) 69

FIGURE 4-4 HILBERT CURVE VARYING DEGREES FROM 1ST TO 6TH (130) 70

FIGURE 4-5 A) HILBERT CURVE B) ADAPTIVE HILBERT CURVE..71

FIGURE 4-6 ADAPTIVE QUAD SIZES DEPENDING ON SAMPLING SHOWING THE ADAPTIVE HILBERT PATH (133)...72

FIGURE 4-7 Q-ALGORITHM TOP IMAGES AND N-ALGORITHM BOTTOM IMAGES SHOWING X-Z AND X-Y PROJECTIONS...73

FIGURE 4-8 TEST CARD MAPPED ON TO EGGCUP MODEL SHOWING THE SECTION EXPOSED IN THE RESULTS 74

FIGURE 4-9 RESULTS 1. 50DPI N-ALGORITHM 2. 100DPI N-ALGORITHM 3. 50DPI Q-ALGORITHM 4. 100DPI Q- ALGORITHM...75

FIGURE 4-10 POINTILLISM BY GEORGE SEURAT *LA PARADE* AND *THE SEINE AT LE GRANDE JATTE*...76

FIGURE 5-1 4 POINTS, 2 PATHS 26% DIFFERENCE IN PATH LENGTH.......................................77

FIGURE 5-2 ARBITRARY CURVED SURFACE 77

FIGURE 5-3 SCHEMATIC ROBOT SHOWING CLOSE CARTESIAN POSITIONS REQUIRED SIGNIFICANT RE-CONFIGURING....78

FIGURE 5-4 DH CONVENTION (136)....85

FIGURE 5-5 ASSIGNMENT OF DH FRAMES TO THE ROBOTIC ARM (DRAWN USING ROBOT MODELLING & KINEMATICS (51))....87

FIGURE 5-6 ASSIGNMENT OF DH FRAMES TO THE TILT TABLE (DRAWN USING ROBOT MODELLING & KINEMATICS (51))....90

FIGURE 5-7 PUMA ROBOT WITH SIX DEGREES OF FREEDOM AND SOLUTION TREE (51) SHOWING SOLUTIONS....96

FIGURE 5-8 REPRESENTING THE LEVEL OF REDUNDANCY WITH THIS SYSTEM....97

FIGURE 6-1 NEW SYSTEM ARCHITECTURE....99

FIGURE 6-2 SCHEMATIC SETUP OF ROBOT INDICATING ROBOTIC JOINTS AND JOINT NAMES....100

FIGURE 6-3 POSITION OF POINTS EXTRACTED FROM CYLINDER AT 10DPI....103

FIGURE 6-4 LAMINA PATH SIDE VIEW....104

FIGURE 6-5 LAMINA PATH PERSPECTIVE VIEW....104

FIGURE 6-6 PERSPECTIVE VIEW OF A SINGLE JOINT PATH USING T2 FOR THE CYLINDER MODEL....108

FIGURE 6-7 PERSPECTIVE VIEW OF SINGLE JOINT PATH FOR R2 ON THE CYLINDER MODEL111

FIGURE 6-8 PERSPECTIVE VIEW OF CARTESIAN CLOSEST POINT PATH FOR CYLINDER MODEL....112

FIGURE 6-9 SIDE VIEW OF CARTESIAN CLOSEST POINT PATH ON CYLINDER MODEL....113

FIGURE 6-10 PERSPECTIVE VIEW OF THE CONFIGURATION SPACE PATH FOR THE CYLINDER....115

FIGURE 6-11 DEFORMED CYLINDER SHOWING LAMINA SLICE PATH....116

FIGURE 6-12 SIDE VIEW OF THE SEAM ON THE DEFORMED CYLINDER....117

FIGURE 6-13 DEFORMED CYLINDER SINGLE JOINT PATH FOR T2....121

FIGURE 6-14 DEFORMED COMPONENT WITH CARTESIAN CLOSEST POINT PATH....123

FIGURE 6-15 CONFIGURATION SPACE PATH FOR DEFORMED CYLINDER....125

FIGURE 6-16 EGGCUP WITH LAMINA SLICE PATH....127

FIGURE 6-17 CARTESIAN CLOSEST POINT PATH FOR EGGCUP MODEL....130

FIGURE 6-18 SURFACE CURVATURE AFFECTS DISTANCE BETWEEN POINTS....131

FIGURE 6-19 CONFIGURATION SPACE CLOSEST POINT PATH FOR EGGCUP....132

FIGURE 6-20 PENGUIN MODEL WITH LAMINA PATH 134

FIGURE 6-21 CARTESIAN CLOSEST POINT PATH ON THE PENGUIN MODEL.........................137

FIGURE 6-22 CONFIGURATION SPACE CLOSEST POINT PATH FOR PENGUIN MODEL...........140

FIGURE 6-23 WIZARD WITHOUT TEXTURE MAP..141

FIGURE 6-24 LAMINA PATH ON WIZARD MODEL...142

FIGURE 6-25 CARTESIAN CLOSEST POINT PATH FOR WIZARD 146

FIGURE 6-26 CONFIGURATION SPACE PATH FOR WIZARD...148

List of Tables

TABLE 1 SUMMARY OF SURFACE COVERAGE TECHNIQUES 15

TABLE 2 COMPARISON OF THE LED AND THE LASER SYSTEMS' PERFORMANCES 65

TABLE 3 HOMOGENEOUS TRANSFORMATIONS 84

TABLE 4 DH TABLE FOR SCARA ROBOTIC ARM 87

TABLE 5 DH TABLE FOR TILT TABLE...89

TABLE 6 TEST COMPONENTS AND DETAIL DESCRIPTORS..102

TABLE 7 MAX MOVE CALCULATION SHOWING THE SUM OF ALL THE LARGEST JOINT MOVEMENTS FOR EACH POINT TO POINT 107

TABLE 8 SUMMARY OF PATH LENGTHS AND MAX MOVES IN RADIANS OBTAINED FOR DIFFERENT SEQUENCING STRATEGIES, MAX MOVES ARE IN BRACKETS 150

List of Graphs

GRAPH 1 T2 JOINT ANGLES IN DEGREES FOR THE CYLINDER MODEL USING THE LAMINA SLICE PATH 106

GRAPH 2 R1 AND R2 JOINT ANGLES FOR THE CYLINDER MODEL USING THE LAMINA SLICE PATH 106

GRAPH 3 TOTAL DISTANCE TRAVELLED BY EACH AXIS IN RADIANS FOR THE CYLINDER MODEL USING THE LAMINA SLICE PATH 106

GRAPH 4 CUMULATIVE DISTANCE FOR SINGLE JOINT PLANNING USING T2 FOR CYLINDER MODEL 109

GRAPH 5 CUMULATIVE DISTANCE FOR SINGLE JOINT PLANNING USING R2 FOR CYLINDER (R1, R2 ARE ALL VERY SMALL) 110

GRAPH 6 CUMULATIVE DISTANCE FOR CARTESIAN CLOSEST POINT FOR CYLINDER 113

GRAPH 7 CUMULATIVE DISTANCE FOR CONFIGURATION SPACE CLOSEST POINT FOR CYLINDER MODEL 114

GRAPH 8 DEFORMED COMPONENT TILT VALUES (T1) WHILE USING THE LAMINA SLICE PATH 118

GRAPH 9 DEFORMED COMPONENT ROTATIONAL ANGLES (T2) FOR THE LAMINA SLICE PATH 118

GRAPH 10 DEFORMED COMPONENT JOINT ANGLES R1 AND R2 FOR LAMINA SLICE PATH.118

GRAPH 11 CUMULATIVE JOINTS TRAVEL R1, R2, T2 AND T1 FOR DEFORMED CYLINDER USING LAMINA SLICE PATH 119

GRAPH 12 DEFORMED CYLINDER SEQUENCED BASED ON TILT ANGLE (T1)....119

GRAPH 13 T2 JOINT ANGLES WHEN SEQUENCED USING T1 120

GRAPH 14 CUMULATIVE DISTANCE FOR EACH AXIS AFTER T1 ACTUAL SEQUENCING....120

GRAPH 15 T2 JOINT ANGLES FOR THE DEFORMED CYLINDER SINGLE JOINT T2 PATH....122

GRAPH 16 DEFORMED CYLINDER R1 AND R2 JOINT ANGLES FOR T2 PATH....122

GRAPH 17 T2 SEQUENCING SHOWING CUMULATIVE DISTANCE TRAVELLED BY EACH AXIS122

GRAPH 18 T1 AND T2 JOINT ANGLES FOR DEFORMED CYLINDER AFTER CARTESIAN CLOSEST POINT ALGORITHM 124

GRAPH 19 DISTANCE TRAVELLED BY EACH JOINT FOR DEFORMED CYLINDER AFTER CARTESIAN CLOSEST POINT ALGORITHM 124

GRAPH 20 R1 AND R2 JOINT ANGLES AFTER CONFIGURATION SPACE CLOSEST POINT SEQUENCING 126

GRAPH 21 T1 AND T2 JOINT ANGLES AFTER CONFIGURATION SPACE CLOSEST POINT SEQUENCING 126

GRAPH 22 TOTAL DISTANCE TRAVELLED BY EACH JOINT AFTER CONFIGURATION SPACE CLOSEST POINT SEQUENCING 126

GRAPH 23 EGGCUP MODEL T1 ANGLES FOR LAMINA SLICE PATH.......................................128

GRAPH 24 EGGCUP MODEL T2 VALUES FROM LAMINA PATH 128

GRAPH 25 ROBOTIC JOINTS R1 AND R2 FOR EGGCUP MODEL USING LAMINA PATH 129

GRAPH 26 TOTAL DISTANCE TRAVELLED BY EACH AXIS IN THE LAMINA SLICE APPROACH TO THE EGGCUP COMPONENT 129

GRAPH 27 CUMULATIVE JOINT TRAVEL FOR EGGCUP MODEL USING THE CARTESIAN CLOSEST POINT PATH 131

GRAPH 28 T1 AND T2 JOINT ANGLES FOR CONFIGURATION SPACE PATH ON THE EGGCUP133

GRAPH 29 CONFIGURATION SPACE PATH TOTAL DISTANCE TRAVELLED BY EACH JOINT FOR THE EGGCUP 133

GRAPH 30 T1 AND T2 JOINT ANGLES FOR THE PENGUIN USING LAMINA PATH 135

GRAPH 31 R1 AND R2 JOINT ANGLES FOR PENGUIN WITH LAMINA PATH..........................136

GRAPH 32 TOTAL JOINT TRAVEL FOR THE LAMINA PATH ON THE PENGUIN 136

GRAPH 33 JOINT TRAVEL FOR CARTESIAN CLOSEST POINT PATH ON THE PENGUIN SHOWING SIMILAR SECTIONS...138

GRAPH 34 PENGUIN T2 SOLUTIONS FOR CONFIGURATION SPACE SEQUENCING................139

GRAPH 35 JOINT ANGLE TRAVEL FOR PENGUIN MODEL IN CONFIGURATION SPACE SEQUENCING 140

GRAPH 36 T2 JOINT ANGLES FOR WIZARD WHILE USING LAMINA PATH 143

GRAPH 37 R1 AND R2 JOINT ANGLES FOR WIZARD USING THE LAMINA PATH....................143

GRAPH 38 JOINT TRAVEL FOR THE WIZARD USING THE LAMINA PATH...............................144

GRAPH 39 TOTAL TRAVEL OF EACH JOINT USING CARTESIAN CLOSEST POINT PATH FOR THE WIZARD COMPONENT...147

GRAPH 40 T2 JOINT ANGLES AFTER CONFIGURATION SPACE SEQUENCING FOR THE WIZARD 149

GRAPH 41 R1 AND R2 JOINT ANGLES FOR CONFIGURATION SPACE SEQUENCING OVER THE WIZARD...149

GRAPH 42 TOTAL JOINT TRAVEL FOR EACH AXIS IN CONFIGURATION SPACE ALGORITHM OVER THE WIZARD...149

Thesis Related Publications

1. Raymond C.W. Sung, Jonathan R. Corney, David P. Towers, Ian Black, Duncan P. Hand, Finlay McPherson, Doug E.R. Clark, Markus S. Gross (2006), "*Direct writing of digital images onto 3D surfaces*", Industrial Robot An International Journal, Vol.33, No. 1,
2. Finlay McPherson, Jonathan R. Corney, Raymond C.W. Sung, (2007), *"Path Planning for Automated Robotic Painting"*, Proceedings of ASME 2007 International Design Engineering Technical Conference and Computers and Information Conference, IDETC/CIE 2007,

Chapter 1 Introduction

Today's manufacturing industry is more competitive than ever with outsourcing to the Far East accounting for an increasingly large amount of production due to the availability of cheap labour and the global logistics networks. The main driver for this offshoring is not manufacturing technologies themselves (e.g. CNC machining, die casting, etc.), but the cost of inherently manual processes, such as complex assembly or painting tasks.

So if manually intensive steps such as hand painting could be automated, labour costs could be reduced to the point where domestic production becomes viable. Motivated by this industrial need, this thesis is concerned with the generation of efficient surface coverage paths, for a novel, non-contact painting process. The topic is one of a general class of problems that exists, in CAD/CAM where there is a requirement to visit many points on the surface of a component. Although the nature of these visits varies with the application (sometimes a cutting head and sometimes a point of light) variations of the surface coverage problem can be found in many different CAM applications such as:

- Surface Machining
- Contact Scanning
- Non-Contact Scanning
- Automated Spray Painting

After outlining the nature of the optical painting process and its particular requirements the following sections consider each of these surface coverage applications in more detail. The last section of this chapter summarises their characteristics and identifies gaps in the reported approaches to path generation that currently limit the effectiveness of the optical painting process. Having established the context of the work, the aims and objectives are presented and the structure of this thesis is outlined.

1.1 Surface Coverage in Optical Non-Contact Painting

The non-contact system that is the focus of this thesis is known as optical painting (1) and was developed within Heriot-Watt University's Manufacturing Group as a means of economically generating products which are geometrically identical but require different images on the surface. Optical painting is a non-contact process similar to photography, after a base layer of photographic emulsion is applied to the surface, a tri-colour laser is directed over an object's surface exposing "*discrete pixels, or dots*" (at 300dpi each dot must be approximately 0.085mm in diameter) on the component's surface, thus creating pictures in a similar fashion to an inkjet printer. After images have been "*exposed*" the component is "*developed*" using normal photographic chemicals (i.e. developer, fixer and stop).The laser acts as the light source and can be directed onto any part of the object's surface by the combined movement of a robot and tilt table that provides a coordinated five degrees of freedom system (Figure 1-1). The use of optical painting was conceived as part of a rapid manufacture unit, where models could be designed modelled and then physically realised (using a layered manufacturing machine) before finally being "*finished*" using the optical painting system. For the system to be successful, it must be capable of recreating a comparable surface appearance within a similar time period to conventional hand painting.

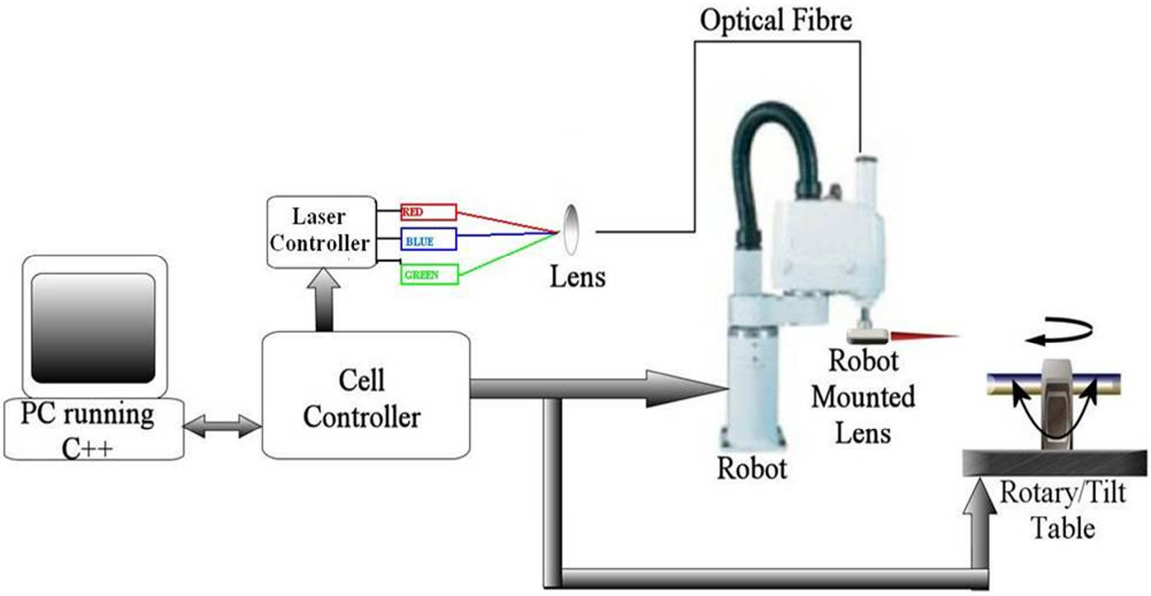


Figure 1-1 Experimental optical painting system

Because the process of painting requires the exposure of thousands of precisely positioned dots, careful planning is required to implement an effective painting path. A path may be deemed successful by visiting all the points in a given job (i.e. full coverage of a model by optical painting) however, this may not be an efficient path. An efficient path completes the job by minimising some quantitative measure such as energy or painting time. In addition to efficiency other factors must be considered to ensure successful execution, such as:

- Shape – Due to the nature of the application a variety of different geometries need to be painted, requiring a high degree of flexibility in the system.
- Lens Orientation – Optical painting relies heavily on the assumption that perfect pixels will be exposed by a beam of light normal to the surface that forms a perfect circle. However, this is not always possible, therefore the use of the best possible position to generate the most circular pixel is required.
- Registration – The use of a calibration point is required to allow the painting system to correctly position and orientate the image on the surface.

- Speed – As the painting system processes thousands of points it is important that each point is handled quickly and efficiently, consequently the mechanical performance must be considered so that slower robotic joints are used less (e.g. the tilt and rotation joints on the tilt table used in the optical painting system are slower than the robotic joints).
- Accuracy – The system requires quick and accurate positions with small smooth changes preferred to large *"Juddered"* changes (juddered in this context implies a movement made at maximum velocity over a short distance which will result in vibration because of the deceleration force occurring when the robot reaches its destination (2)), as settling time will affect the painting time and the accuracy.

The initial implementation of the optical painting path planner adopted a "*brute force*" strategy (analogous to rapid prototyping systems) that used a layered approach to generating robot instructions. The virtual model was first sliced into layers whose thickness was determined by the resolution (i.e. dpi, dots per inch) required for the final image. Each "*layer*" (or contour) is painted by individually moving all the joints of the robot required to place the lens of the laser at the correct orientation to allow a perfect circular pixel to be exposed. Although effective, this brute-force (top down lamina slice) approach caused large and sudden changes in the system's configuration (i.e. where the robot was required to make significant movements of several joints to be in position and orientation required to paint the next point in the sequence. This is illustrated in Figure 1-2 where points that are close require significant repositioning of the robot so it aligns with the surface normal).

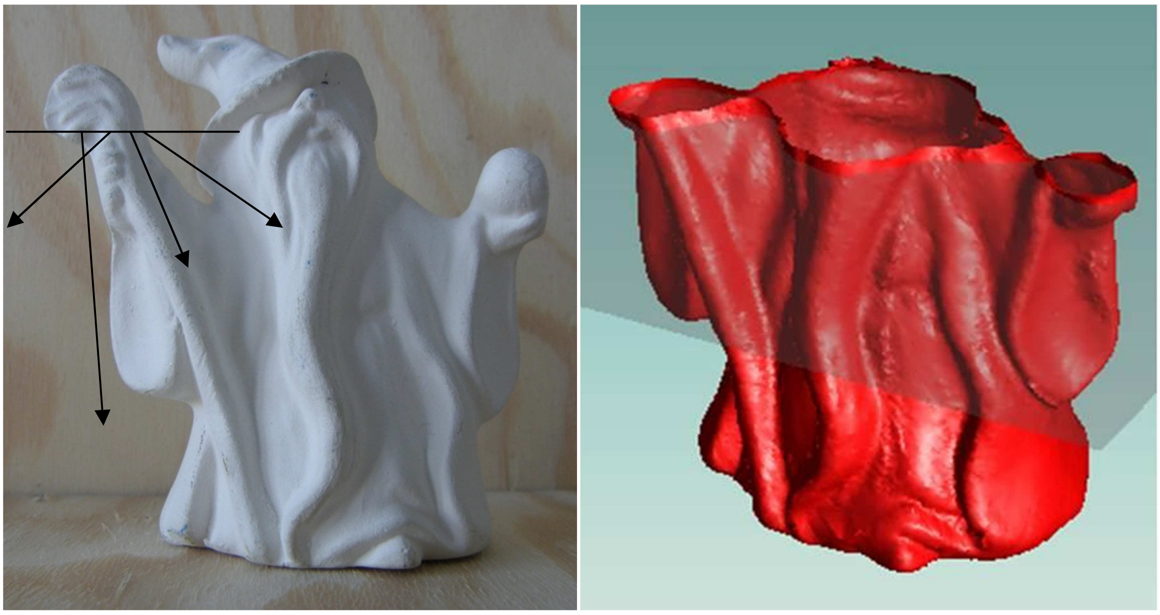


Figure 1-2 Complex model with linear slice produces significantly different normal vectors

In other words, adjacent pixels (on a slice of model) might require the robot to make large changes to its configuration. These swings produce inefficient paths whose execution times are longer than necessary. The process is fully described in Chapter 3. However, many other

CAD/CAM processes have to produce surface coverage paths and the following sections describe these and discuss if any of the path generation strategies used in these applications might be applied to the optical painting system.

1.2 Surface Coverage in Machining

Machining was initially a 3-axis manual task where an operator would fix the work piece onto the bed of the machine, either the bed or the cutter would then be moved on a two axis grid (i.e. along the x-axis or y-axis) while the cutter was lowered onto the surface (along the z-axis). This initial form of machining was highly skilled and relied on the operator controlling the movement of the work piece and cutting head simultaneously

In the early 1950s John T Parsons of the Rotary Wing Branch of the Propeller Lab at Wright Patterson Air Force Base was involved with the manufacture of rotor blades for helicopters (3). This required a much higher tolerance than previous jobs and therefore Parsons enquired if IBM would allow him to use one of their computers to generate a cutter path. Coupling this with a simple paper tape, he developed a system to control a milling machine automatically; this being one of the technologies that led to what is now known as CNC machining. Other technologies were also being developed simultaneously, however; the use of a separate system to pre-process the tape instructions allowed the machine to be manufacturing while the initial setup was being performed on the tape and this reduced the downtime between products (4).

As computational power grew, so too did the complexity of the machine tools being controlled, with the introduction of two further axes (rotation and tilt) to extend the capabilities of the original 3-axis machine (5). Coordinating the movement of these 5-axis systems meant more complex profiles could be machined (Figure 1-3), including curved B-Spline surfaces. This also allowed multiple faces to have surface operations performed on them so long as the shape of the component did not clash (i.e. collide) with the milling machine.

This growth in multi-axis machining has spurred the need for more complex path planning algorithms that machine (i.e. visit) every point on the surface allowing the complete component to be machined to fine tolerances. Machining every point on the surface of a complex component is challenging because:

- Shape – Each component will have its own unique shape, consequently individual tool paths (i.e. part programmes) must be calculated for each job.
- Contact – As complete surface coverage is required, the tool bit must be contactable to the complete surface without introducing unwanted interference (known as gouging) to completed surfaces.

- Velocity and Acceleration – Surface machining is very dependent on feed rates and tool movement which in turn determine surface finish (i.e. fast tool movement may apply stress and cause the end of the cutter to deflect resulting in excess cutting or dwell marks).
- Collision – Machining systems not only have to avoid collisions with the "job" but also have to deal with the fact that it changes shape as it is machined.

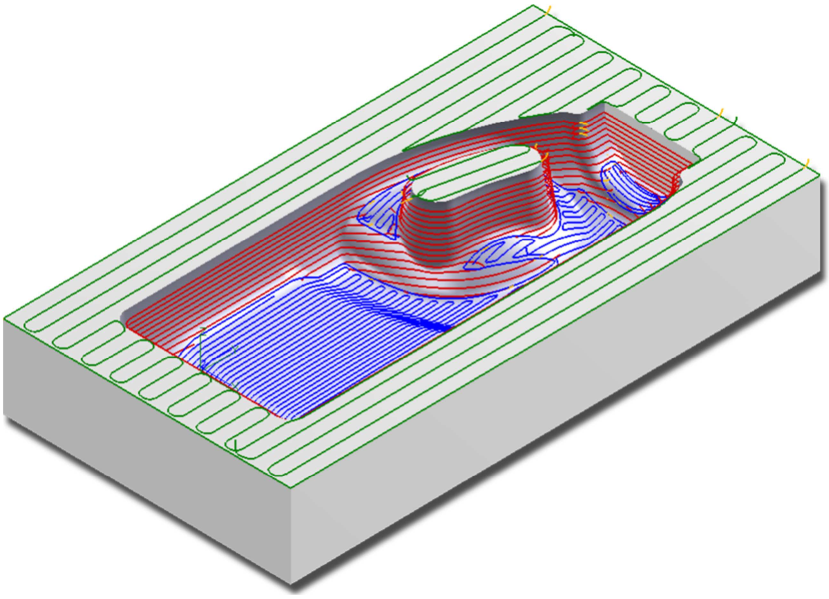


Figure 1-3 Combination tool path optimized to achieve uniform cusp height (6)

A typical strategy for generating 5-axis paths consists of three separate paths known as the facing cut, roughing cut and the finishing cut (7). The facing cut is used to generate a datum plane which tends to be a simple 2-axis path that moves at uniform speed, to establish a plane relative to the axis of the machine. The second cut, the roughing cut, is used to generate the general profile of the surface. This is normally implemented with either a down or a spiral cut. A down cut operates by identifying features that protrude, then cutting round them (in a similar fashion to layered manufacturing). A spiral cut removes material on a continuous path starting from the centre and working in an ever-increasing path downwards. During the roughing cut, the path interval tends to be in the order of one tool's thickness. The finishing cut operates with a much closer tolerance and the path interval is normally significantly overlapping the area previously cut, planning again tends to operate in down or spiral cuts.

The finishing cut is closest to the path required by the optical painting system because it makes only a small change to the overall shape but the coverage, speed and orientation of the cutter is crucial. Section 2.1.2 Chapter 2 describes how paths are generated from parametric surface descriptions to support the operations. However, the observation can be made that, even after several decades of research, the process is not entirely automated. Part programs are generated through an interactive, semi-automated process that requires users to manually identify and sequence patches to be machined accurately (8). Only recently have CAM systems been able to simulate the visual appearance of the final parts (i.e. excluding dwell marks and patterning from the movements).

1.3 Surface Coverage in Contact Scanning with a Coordinate Measuring Machine (CMM)

A coordinate-measuring machine (CMM) is a machine for dimensional measuring. It uses a mechanical touch probe positioned over a work area with a known size. The probe is accurate while working in the X-Y-Z grid area to a fraction of a mm (approximately ±0.0001mm). The probe touches the features of a component and records their relative coordinates before moving to the next "*feature*". Newer CMM machines, instead of touching the surface of a component drag the probe across it extracting data at regular intervals. This generates results quicker and more accurately (9).

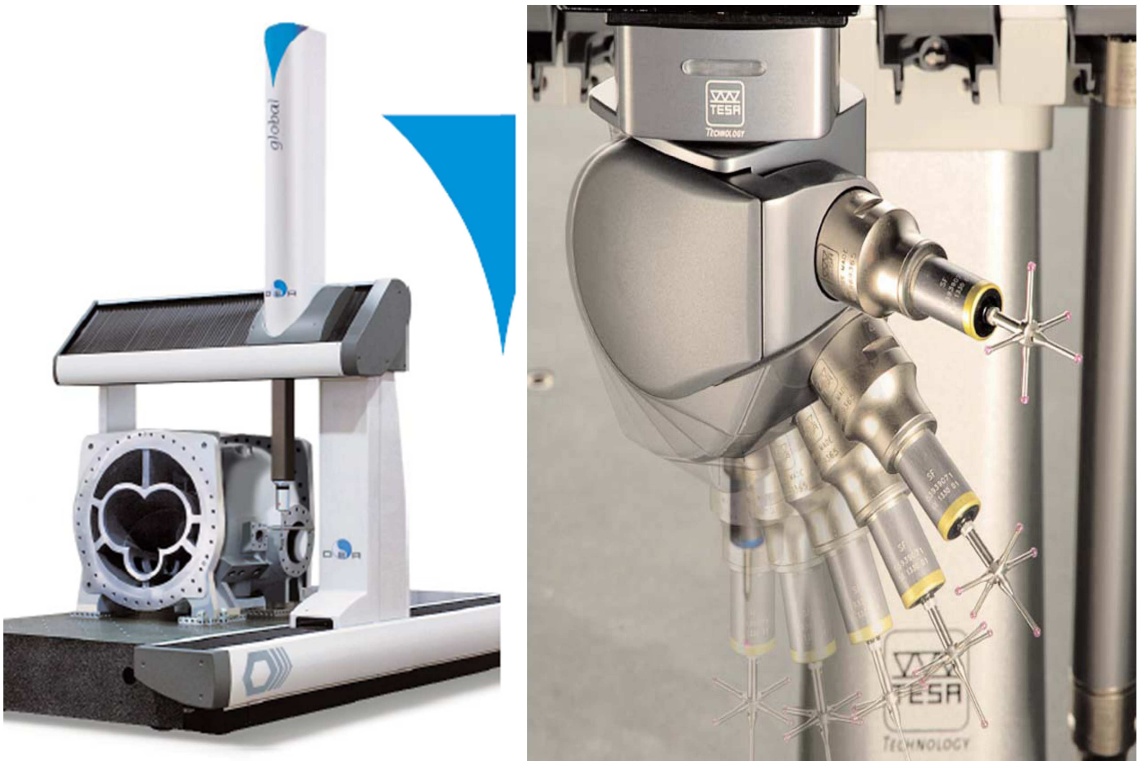


Figure 1-4 Conventional CMM machine and probe head (10)

The first CMM was developed by Ferranti in Scotland in the 1950's (11) (12) however, several machines from other manufacturers were released over the next couple of years. This suggests that many companies had realised that control technology and the demand for the accurate analysis of parts was creating a new market. One of the side benefits was the ability to reverse-engineer parts by using the probe to acquire the geometric data (e.g. point clouds) from shapes that could then be triangulated to generate 3D models.

The original application was measurement of discretized features (e.g. circular holes) but recently these systems have been increasingly employed for reverse engineering and the inspection of complex surfaces (e.g. Turbine Blades) machined by 5-axis machine tools. The complex surface inspection application is most relevant to this thesis.

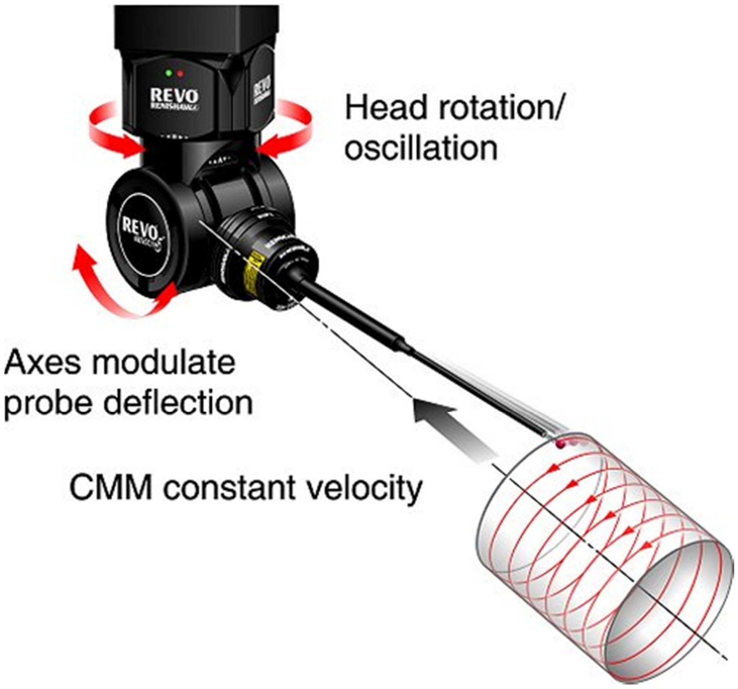


Figure 1-5 Helical scan for hollow components (13)

CMMs require the probe to visit large numbers of surface points to attain high accuracy measurements, and this can be very difficult because, as with machining, each individual component's geometry varies, creating challenges of:

- Registration – Complex geometry tends to be defined in the form of a computer model. Consequently, a process known as registration is required to calibrate (i.e. align), the machine to the orientation and location of the geometry.
- Contact Orientation – The touch probes commonly used in CMMs rely on continuous surface contact usually at an angle to the surface (as maintaining pressure normal to the surface would bend the probe).

- Contact Pressure – Regardless of its speed the probe must maintain a continuous and almost constant pressure upon the surface to gain accurate readings.
- Collision – Interference between the CMM machine and the object must be avoided.

CMMs use a variety of different scanning strategies depending on the shape of the object being scanned however, the initial step for any form of scanning is the establishment of a datum point. This is a point that can be referenced on the computer model and easily identified on the physical part, for example, a common strategy is to use the machine to identify a corner or a simple feature such as a hole. Once the virtual computer model and the physical object are correctly orientated it is possible to analyse the complete shape using an axial scan (13) where the probe moves backwards and forwards along a single axis with small path increments. This can be time consuming and the use of more targeted scanning can be used to analyse specific features. An example of this is "*Helical Scanning*" for checking the roundness of holes where the probe moves across the surface of a hole in a spiral path (Figure 1-5). Interestingly, the fine details of a CMMs surface path are achieved using feedback control rather than the explicit coordinates generated from a CAD model.

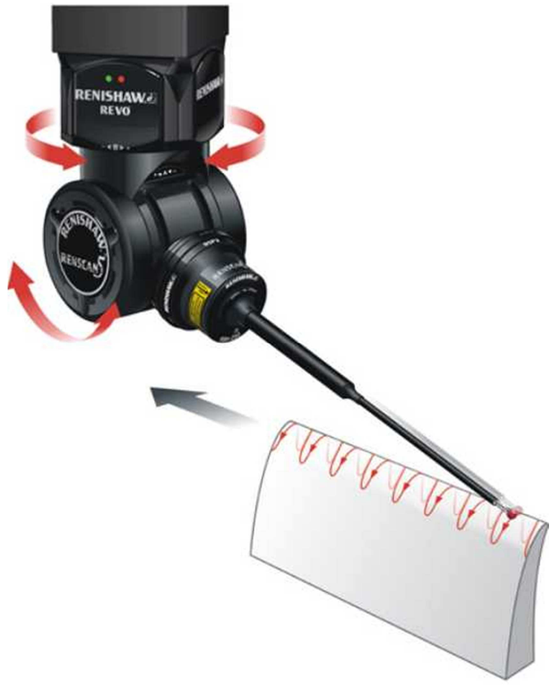


Figure 1-6 Blade sharp edge sweep scan (13)

Pre-programmed (e.g. "*Hard Wired*" or "*Canned Cycles*") inspection paths created by manual trial and error are also used for specific inspection tasks such as the "*gasket scan*" that is used to inspect a surface after manufacture to ensure flatness and high quality surfaces are maintained. Blade scanning (i.e. turbine blades) has spurred development of a collection of different scanning techniques ranging from the "*sweep scan*" (Figure 1-6) that analyses the surface curvature to "*blade sharp edge sweep scanning*" (Figure 1-7), which moves across the leading edge of the blade and identifies any anomalies on the physical model.

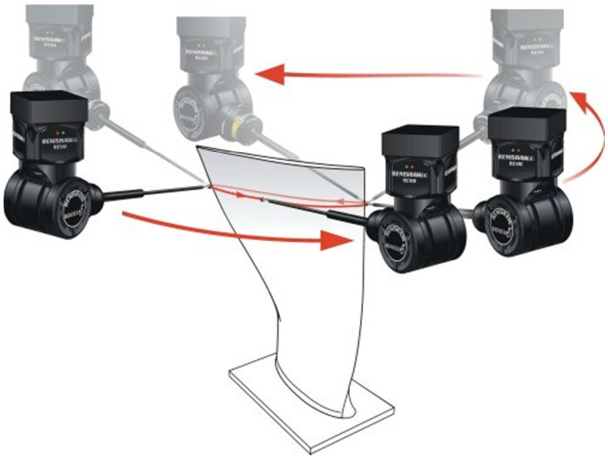


Figure 1-7 Round the blade scan (13)

Despite its sophistication, CMM inspection generally repeats manually generated "*paths*" enabled by fast feedback. The inspection of overhanging or re-entrant features requires manual programming.

1.4 Surface Coverage for Non-Contact Scanning

Many of the challenges associated with CMM programming arise from the need for physical contact; however, a family of non-contact laser scanning devices also exist. Laser scanning can be broken down into two main categories; active and passive. Neither type requires physical contact with the object being scanned. However, active scanning does require some form of electromagnetic wave to be emitted onto the surface of the model.

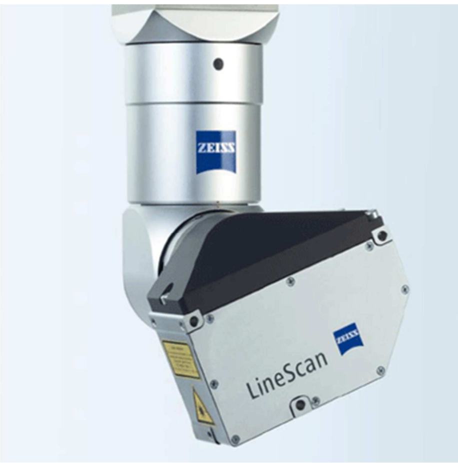


Figure 1-8 Carl Zeiss Line Scanner for Capturing Surfaces using Point Clouds (14)

Active scanning is the most common form of scanner and emits some form of wave (i.e. Ultrasound, X-Ray). Active lasers can be further classified as either, 'time of flight' or triangulation systems. The time of flight laser emits the signals and calculates the time it takes for the signal to be seen by a detector so the distance can be calculated. The triangulation technique uses a laser and a camera, so when a light is emitted by the laser the camera detects its position on the surface of the component.

This is obviously dependent on the position of the camera's viewing field as this determines the distance of the point. As the distance from the camera to the laser is known, both methods are used but for different purposes; time of flight can be used to calculate the distance to an object far away very quickly but has limited accuracy. Triangulation systems, on the other hand (such as the Minolta VIVID910) are accurate to ± 0.1mm but can only be used on objects close to the emitter. Passive scanners work differently as they do not transmit, they simply receive pre-existing information. One simple, often-used method known as Stereoscopic Imaging, which requires two cameras (each registering pictures using the knowledge of the relative camera positions) to calculate the position of objects by analyzing the difference between the two images from the cameras.

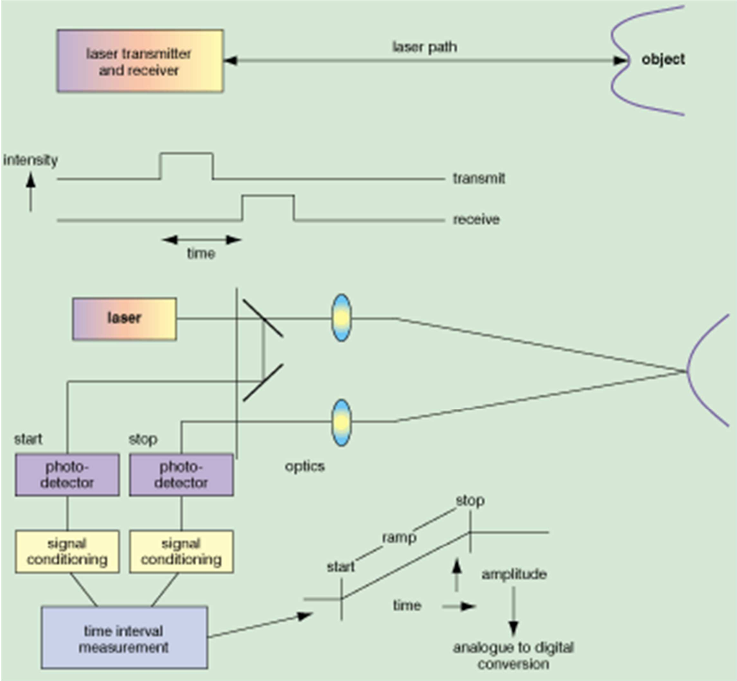


Figure 1-9 Schematic Diagram of a Time of Flight Optical Scanner (15)

The use of non-contact scanning has been exploited in a large range of applications (i.e. rock formations geology, building capture, historical applications) where contact scanning is un-feasible or inappropriate. Non-contact scanning has the following challenges:

- Shape – Non-contact scanning can easily capture the profile of models and can generate solid 3D models but only after manual processing of the results.
- Protrusions – Scanning can easily be performed working in a sphere around the object but this does not mean features will be thoroughly identified. Complex techniques can be utilized to operate within the bounding sphere to identify overhangs and other more complex features.
- Registration – The use of some form of registration is required for objects where the fixtures prevent complete, single position, scanning.
- Surface – Colour and orientation information such as the normal vector and the component information may also be required.
- Material – Optical scanners are limited to materials that are not translucent or reflective.

Path planning for non-contact scanning can be broken down into three separate categories automated, hybrid and manual. Manual scanning tends to be used on systems such as Faro arms (16) where the user controls a 5-axis scanning probe that emits structured laser light and records surface information by calculating the surface position relative to the base of the scanning unit.

Automated scanning is used in systems such as Renishaw's "Cyclone" where a multi-axis machine directs a laser scanner. The initial scan is used to calculate the limits of the component, prior to a second scan which is normally performed over a linear pattern such as a raster scan. This incrementally moves along the x-axis while scanning along the y-axis to produce an x-y-z map of the component. This two-stage approach is used to reduce the chance of collisions occurring and allows a more detailed scan to be performed with confidence. Once a Cartesian map has been generated over the surface, increasingly complex scanning paths can be manually generated to examine re-entrant surfaces or features.

Perhaps the most common approach to scanning is the use of a hybrid planning method where a combined approach is used rather than a fully automated scan or manual scan. These approaches vary greatly depending on the objective of the scan and the pre-existent information that is available to the system (i.e. if the scan is performing an inspection where a 3D model is available for comparison or if the data is simply being captured). (17) Seokbae builds an approach based on surface patches which are obtained by analysing the digital CAD model. Through using these patches and gathering information about the scanner, the system creates an optimum scanning path for the specific model with the specific scanner. This hybrid approach does not automatically control the scanner and move the scanning head over the component but produces an efficient path that minimises the chance of data anomalies which are synonymous with scanning. However, the system still requires the user to control the movement of the head though this could be automated with further work.

The path should reduce the errors that are obtained while scanning (i.e. areas where the normal and the scanner would produce an oblique angle and around the edges of the model where data can easily be corrupted) and it can also analyse the model's occluded surfaces to produce an optimum approach for handling these areas. Further work similar to this includes Xi et Shu (18) who proposed a sectioning algorithm which then looks at the surface profile and identifies the scanning window for the component. Once each section has been analysed a scanning path is created that allows an automated scanning system to be adopted. These algorithms could both be used for computer aided inspection tasks where pre-existent CAD models can be used for comparison.

It can be seen from the literature that, despite the many uses of optical scanning systems, the complete automation of the system would appear to be very difficult due to the variety of different components that the system is required to scan. Though algorithms exist for inspection operations, the automation of reverse engineering applications would appear to be a significant gap in the current literature. The optical painting system relies on pre-existent CAD data combined with aesthetic knowledge, therefore this thesis can be seen to be closer to the inspection operations than the reverse engineering applications.

1.5 Surface Coverage in Contact Painting (i.e. Spray Painting)

Contact painting has been deployed in a variety of different applications (i.e. Car components, Rust Shielding), consequently the process has been widely investigated and well documented. The use of spray painting in the car industry "*where it is a critical step*" is under constant development to generate efficient path planning algorithms that minimise the amount of overshoot generated, thus reducing the environmental and economical impact.

On a car assembly line, paint is applied in several coats to prevent rust and give a colour range. The painting process must be quick and the initial coats are usually sprayed by robots followed by hand spraying to finish. On demand manufacture, requires frequent colour changes as the parts pass the robots and these parts are geometrically different (e.g. different sizes of car model) which creates problems for the spraying process; as much as 40% of the paint can miss the target, causing waste and toxic off-spray. The process has been shown to cause significantly more environmental damage that the emissions from the car itself (19).

Imperfections during spraying; like orange peel, blistering or running can cause parts to be rejected, as can the orientation of the flakes of metal added to the paint. These flakes must be parallel to the surface of the component so that a consistent reflectivity and surface is obtained.

An atomizer is used to generate the small paint droplets that are then passed through an electric field to charge them. The droplets are then initially directed by a function of the diameter, velocity and angle of flight from release from the atomizer. The final trajectory is a function of the air velocity and the interaction with other paint droplets and the electrostatic charge that has been applied to the droplet (19).

Spray painting tends to be implemented over surfaces rather than complete models, as the development of smooth paths is difficult and an even distribution of paint is also harder to maintain over angular surfaces; therefore, automated systems tend to be used on smooth parametric surfaces (20).

The traditionally manual process of "teaching" the robot a paint spray path is slowly being replaced by automated systems, which use machine specific information to create accurate offset surfaces that are subsequently used to calculate effective paths (i.e. minimising overshoot). The issues involved with contact painting are as follows;

- Shape – Complex and angular surfaces are difficult to paint, as excess paint tends to be distributed in acute angles; this makes maintaining an even paint distribution over the component difficult.

- Excess Paint – As paint is distributed from an atomizer this normally means a conical distribution pattern which for sharp edges may prevent accurate distribution.
- Geometric Awareness – As with similar problems such as robotic assembly end effector access is important, the complete robot must be considered to prevent collisions between the robot and any work piece it is operating on. Indeed any multi-dimensional robotic arm can cause access violations (i.e. clashes between robot and job) and can damage components if the path is incorrectly configured.

Automated path planning for spray painting has been widely researched (21) (22) (23). Because the process is not a simple point to point path and requires the even distribution of paint, many different variables must be considered (e.g. painting cone, paint particle deflection, overshoot) when planning the process movements. The foundation for most papers involves the parameterization of the surface and then the use of an offset surface at the optimum painting cone or focal cone length. Once this surface is formulated, a "*coverage path*" is created to generate an even distribution of paint over the entire surface while minimizing the amount of excess paint used in overshoot areas (24).

1.6 Conclusion

Process	Axis	Point/Path /Hybrid	Automation A/M/H	Speed	Geometric Complexity	Path Planning Complexity	Part/ Patch/ Both
Optical Painting	5	Point	Automated	Medium	Accessible	3	Part
Surface Machining	5	Path	Automated	Medium	Feature	5	Both
Contact Scanning	4	Hybrid	Hybrid	Slow	Feature	1	Both
Non-Contact Scanning	3	Path	Automated	Fast	Accessible	1	Part
Contact Painting	5	Path	Automated	Fast	Simple	5	Patch
Hand Painting	/	Hybrid	Manual	Slow	Feature	2	Part

Table 1 Summary of surface coverage techniques

A quantitative comparison of path planning methods is presented in Table 1. Table 1 summarises the main characteristics of each application's approach to path planning. This summary has included hand painting as a process (though this has not been explicitly discussed in the introduction) as this is still one of the most significant surface coverage techniques used in manufacturing industry. It can be observed that when a worker performs detailed painting over a surface the bulk of the movement is performed by the smaller finger joints rather than the larger elbow and shoulder joints. Indeed a common strategy is to manipulate the orientation and location of the object with one hand, so it is positioned to be easily accessible by the fine movement of the other hand (25). Hand painting can be performed by workers of varying skill levels however; it can also be observed that skilled workers perform significantly better than un-skilled workers as they organise and prepare their work (26).

The number of axes employed is used, in Table 1, to provide a rough measure of the automation difficulty. For example non-contact scanning only uses three axes as it simply moves linearly around the part (although higher order systems exist for scanning, the bulk of optical systems only use three axes), whereas machining requires five axes to accommodate a finer control over the orientation of the machine tool. The second column identifies the path planning strategy and shows that hand painting and coordinate measuring machines utilise hybrid strategies that can be point or path based. In the case of hand painting, the strategy is very dependent on the final product and the implementation adopted. The column called Automation breaks the different technologies down into the degree of automation and it can be seen that most are fully automatic however; coordinate-measuring machines can be a hybrid strategy (as user may select the scanning strategy or can control the scanning probe manually). The analysis of the speed is based on the time required to implement the scan and does not take into account pre or post processing. The table column also assumes a single complete surface coverage, meaning contact painting may look fast but in reality, a single pass is never used in contact painting as this would not deliver a uniform thickness across the surface.

The measure of geometric complexity allows a comparison of the various degrees of surface complexity that each system can handle. As the surface complexity increases, the limits of each system's capabilities are reached, for example a mug shaped component could not easily be painted by contact systems. However, the optical painting system and the laser scanning could perform operations as long as the surface was visible. Contact painting can be seen to handle simple surfaces, as the requirement for uniform coverage limits the degree of complexity that the system can handle. Surface machining, hand painting and contact scanning require access to the model's surface and are therefore limited to coverage of the surface and features associated with the surface. The remaining approaches do not require contact with the surface and therefore can operate outside the complex hull of the component, these applications only require visual accessibility (i.e. visually accessible areas).

The penultimate column shows a numerical indication of the complexity of the path. The higher the number, the greater the difficulty, it is proposed that contact painting and surface machining produce the most complex paths as an obvious visual solution is not always possible. These applications both have further considerations that affect the path. In the case of contact painting, the key considerations are the even distribution of paint thickness and the minimization of overshoot (i.e. low variation of velocity and acceleration). Surface machining on the other hand, requires knowledge of the process of cutting so that damage to previously machined areas is prevented from occurring.

From Table 1 it can be seen that optical painting has a unique and demanding set of characteristics, as it requires complete surface coverage of the component, a fast execution time and the ability to handle different geometries which requires a novel and efficient path planning system. However, many existing planning systems demonstrate approaches that might be modified to support the optical painting system and these will be looked at and evaluated in Chapter 2 of this thesis.

This thesis will concentrate on refining and building on knowledge from pre-existent systems, combining the speed of contact painting with the geometric complexity of hand painting and the automation of surface machining. The thesis surveys most recent and relevant literature and summarises this information in the context of related topics such as surface visibility checking, robotic movement and intelligent sequencing. The thesis will then quantify the results and compare the proposed approach with the other methods discussed.

1.7 Aims and Objectives

This thesis aims to develop and implement a new form of path planning algorithm devised specifically for use with the optical painting system and with the potential to be adopted by other surface coverage systems. The algorithm will concentrate on minimizing the amount of movement required to generate a surface coverage path, thus reducing the time and energy required for painting a specific object. The algorithm will be devised by surveying the literature on path planning algorithms so that previous approaches can be extended both for use in this application and to suit the specific requirements of a mass customization system.

The objectives of this work are to develop a path planning algorithm for the optical painting system that:

1. Minimises robotic movement.
2. Can robustly handle a variety of different geometries.
3. Is potentially expandable to a variety of different surface coverage techniques.
4. Can be adapted to support different robotic configurations.

1.8 Thesis Outline

This thesis comprises seven chapters summarised as follows:

Chapter 2 presents a summary of previous research work in path planning algorithms and in particular configuration space systems. It also reviews the economic benefits of rapid production techniques concentrating on mass customization technologies. This leads to a summary of methods for personalizing components and virtual mapping techniques.

Chapter 3 addresses the specific details of the optical painting applications, highlighting the hardware and software technology used to implement the system and addresses the constraints that the current systems present.

Chapter 4 introduces surface appearance technology and discusses methods of assessing the final components. Further to this, methods of improving the surface appearance are presented and discussed.

Chapter 5 presents a configuration space path planning algorithm optimized for the optical painting system and describes the kinematics that have been used.

Chapter 6 looks at the implementation of the configuration space planning and approach and presents the results obtained.

Chapter 7 draws conclusions from the work presented within this thesis and identifies goals that have been achieved, further work is suggested and possible research directions for the future are illustrated.

Chapter 2 Literature Review

This chapter presents the three areas of knowledge that underpin this thesis: 1) the algorithms underlying current path planning approaches, 2) the integration of mass customization into the global production market, and 3) shape and surface appearance technologies.

2.1 Path Planning

The systems that use path planning are almost endlessly diverse ranging from simple human movement to integrated manufacturing units with multiple, moving axes. There is a constant drive to improve path planning with the development of GPS units to shave minutes off travelling times or provide real-time traffic information to reduce stationary time. This example reduces the execution time and increases efficiency; however, other systems may seek alternative goals. Industrial path planning integrates other factors (i.e. tool wear and finish) and aims to reduce the overall cost and time required.

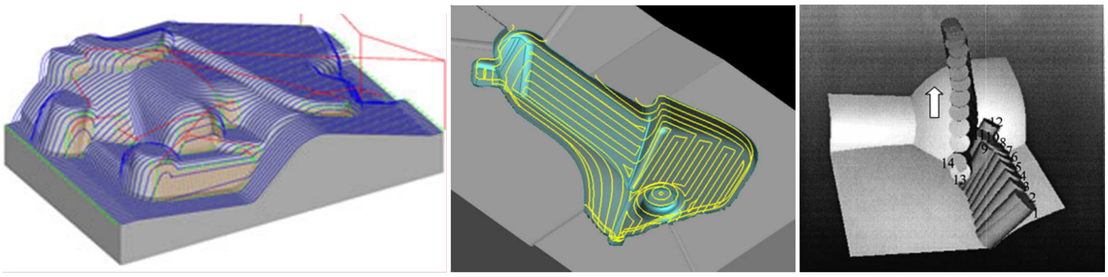


Figure 2-1 a) Linear Cartesian path (27) b) Parametric surface path (28) c) Configuration space optimized path (29)

In terms of computation, industrial path planning can be divided into three distinct categories based on the reference coordinates that they use to generate the robotic or machine tool movements.

1. The earliest work uses the Cartesian coordinate frame and represents an external view of the system, where the relative positions of work piece and machine are related using Euclidean transformations (i.e. a reference point is specified as the origin for the system and all movements are referenced from this point).

2. Parametric path planning methods have emerged with the introduction of 3D CAD on to the factory floor, as solid models represent complex surfaces with parametric descriptions. Essentially this form of planning adopts a model-centred approach.
3. The final method is based on configuration space and can be regarded as representing a robot (or machine) view of the world. Originally developed for robotic path planning it requires significant computation to support surface coverage processes. However, this approach could radically change the current method of machining as configuration space looks at the specifics of the robot (i.e. targets a solution for a specific robot rather than a generalised solution for a specific model).

The following sections review each of these areas.

2.1.1 Cartesian Path Planning

Although Cartesian path planning has been implemented in a variety of different CAM applications, this literature review has focused on the problems that are associated with surface machining, as these are very similar to those encountered in the optical painting application.

Three main path generation strategies have evolved to support 3-axis, 2.5D machining, these are; offset milling, raster and single-direction raster (30).

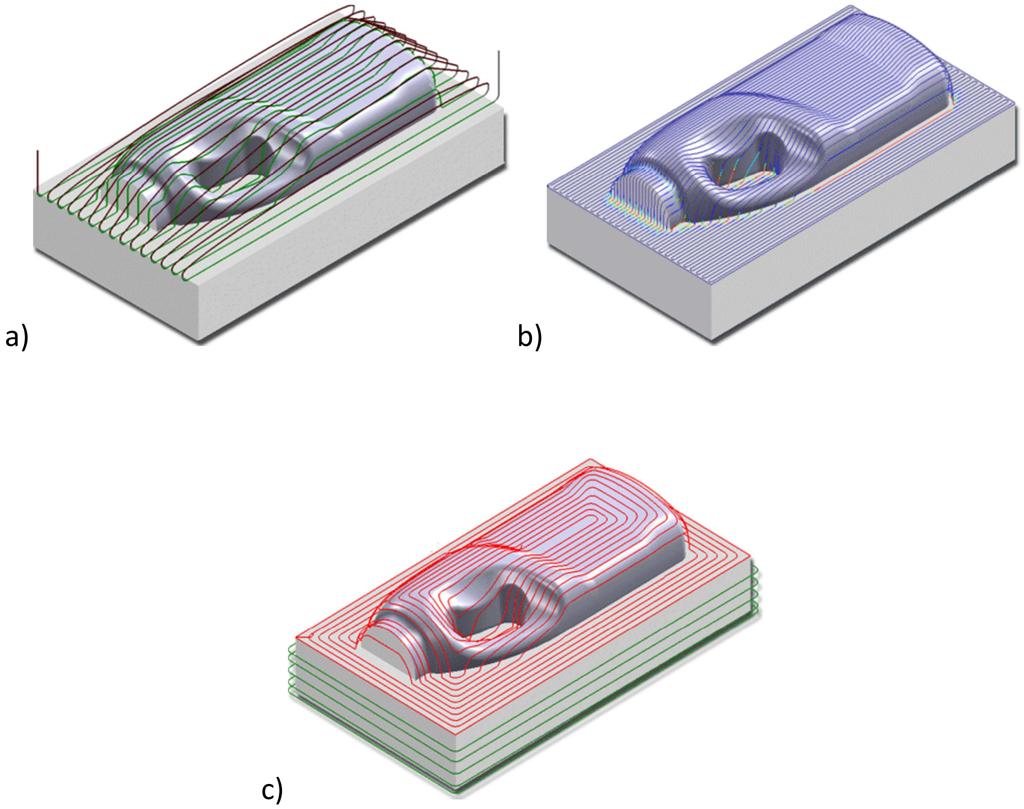


Figure 2-2 a) Single direction raster path b) Raster cutting path c) Offset cutting path (6)

Offset milling, which has a variety of different names, (e.g. window frame, reduction milling, spiral path and incremental milling (30) (31) (32)) is where the cutter starts at a given point then generates a simple 2D-shape such as a rectangle. The cutter cuts around the rectangle then generates further similar shapes within the initial one creating progressively smaller cutter paths Figure 2-2 c).

Raster and single direction raster are both very similar path planning strategies where a start point is given and linear parallel paths are generated across the model at specific intervals, usually slightly smaller than half the radius of the tool being used to machine the component. In raster machining, the linear paths are joined at each end with either a radial curve or a linear movement Figure 2-2 b.

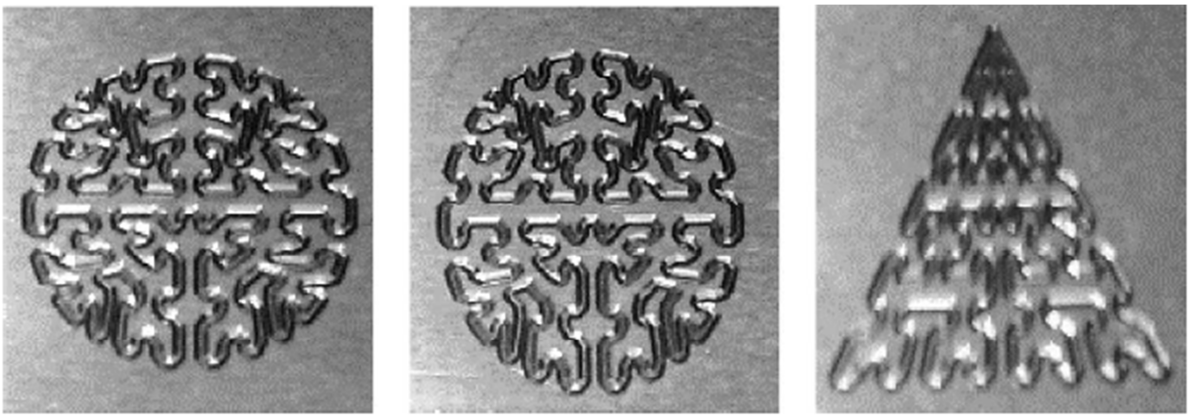


Figure 2-3 Fractural tool path using Hilbert Curve (33)

Researchers have also investigated other, non-raster, methods for generating tool paths such as Chen et al. and Anotaipaiboon et al. (33) (34) who use a classic surface coverage technique known as a '**Hilbert Curve**' to generate complete surface coverage; this can be seen in Figure 2-3. A Hilbert curve can be created with a high order to maximize surface coverage or with a lower order for less coverage or for a larger tool bit. Chen et al. goes on to propose that a further extension could be the use of a chamfered Hilbert curve so that the tool speed can be maintained. The practical results are very limited but do demonstrate adaptability to different surface shapes. However, the results do not compare machining time with conventional techniques and significant further work would be required to attain a finished planner. An increasingly common commercial approach is to generate "*layered*" cutter tool paths directly based on data obtained from the STL data (35).

Each of the three strategies shown in Figure 2-2 (Single Direction Raster, Raster, Offset) have been further optimized by investigating key steps in the planning of the tool path. For example, the study of cutter paths by Toh (30) reviews work concentrating specifically on high speed roughing strategies for hardened steel. In doing this, cutter orientation was identified as one of the main issues that affect cutting time. Where simple polygonal faceting was used, the results showed that in raster milling the orientation of the components affected the cutting time and this could be reduced by orientating the component along the longest axis for machining. Further conclusions from Toh (30) suggest that raster machining produces shorter tool paths than offset milling and that start position does not affect cutting time in offset machining, though both of these conclusions would appear logical it would be interesting if the research had expanded to look at the quality of finish obtained by each of the planning strategies.

Yongfu et al. (36) suggests the use of larger cutting tools for finishing paths, then using a cleanup cutter for sharper corners. It is suggested that cutting lengths could be reduced by as much as 70% and machining time could be reduced by 50%. However, these figures do not take into account retooling the machine; therefore the time reduction would be less. It is also suggested that the same approach could be used on complex polyhedral models. While this would be true for models with relatively low curvature, where large areas of cleanup cutting is not required, for complex models with large surface curvature changes, large areas of clean-up cutting would be required which could render the selective cutting redundant.

Although it is clear that cutter path planning algorithms have many considerations, none are quite as important as gouge prevention and consequently a considerable amount of literature exists in that very active area of research ((37) (38) (39) present reviews of the extensive literature on gouge prevention). Hwang and Chang (37) discuss a simple approach to gouge prevention when using flat mills on 3D surfaces; firstly, the model is triangulated, then the cutter paths are generated using the cutter contact (**CC**) point. The cutter location (**CL**) point (i.e. the point where the cutter will generate the correct contact point) is then generated by investigating the multiple facets that are in the shadow of the cutter. Using a z-map the contact facet can then be interrogated over either the complete facet, the edge of a facet or the vertex of the facet to generate the CL point. Though this system appears to detect possible occurrences of gouging, it does not expand into assessing the surface finish. Therefore, though a path may be free from gouging it may not meet the requirements of the client for finish quality. It is important for any path planning algorithm to fulfil the design specification and pass quality control once the task has been completed.

Most of the literature that has been covered so far focuses on 3-axis machining in the Cartesian space, very little work has been undertaken in 5-axis Cartesian space due to the complexity of the problem. However, some crossover work has been undertaken, which expands 3-axis techniques into full 5-axis surface machining. Most of these algorithms are targeted at operating over small patches of the surface of components such as pockets. Lee et at. (40) investigate pocket milling where large areas of material removal can be undertaken using conventional 3-axis techniques. Once this has been completed, an offset tool path is generated to accommodate the slanted sides of the pocket and step paths are then created so that the tool can re-orientate itself to be perpendicular to the cutting surface. This technique would work relatively successfully for simple pocket features where the base of the pocket needs to be flat. However, it would struggle when the pocket needed to be milled with a slight undercut, as the leading edge would be gouging the base of the pocket whilst cutting the undercut. The gouging would be increased if the base of the pocket had a rising curvature. Though the proposed technique has some applications it is important to note that due to the limitations, this approach could only be used on a small number of components.

The other observation that can be made from the literature is that totally automated cutter path generation appears still, to be limited to layered approaches. Conventional three and five axis approaches to part programming systems still require the human user to identify the areas being cleared by the specific tool path.

2.1.2 Parametric Path Planning

Parametric path planning has been implemented in various different applications with spray painting and machining being the two most well known. The use of parametric planning was introduced as a means of reproducing complex surfaces created using computer aided design packages where smooth curves can be modelled using mathematical equations. The mathematical equations for a complex surface can then be used to generate a two-dimensional mesh overlaying the surface of a component; the two dimensions of the mesh can then be used to define the position of a point on the surface of the component which is known as parameterization. Parameterization relies on a reference point on the surface of the model, which is used to create a grid covering the surface of the model, the grid exists only on the model's surface and when combined with the reference point can be used to convert the 2D grid position to a 3D Cartesian position (41) (42).

Since the early 1990s, a variety of different parametric algorithms has been presented, each concentrating on a specific task or application. Most of these algorithms have never been integrated into an industrial system due to the lack of a commercial platform (43). However; it has been proposed that a taxonomy or classification system may exist for the algorithms based on the approach to generating the tool path (44) (45). The three principle categories of this classification of tool path would be; iso-parametric, iso-planar and iso-scallop, each of which tries to maintain a constant value for one of the variables (for example maintaining a constant offset between the paths) within the system. The aim of any parametric tool path is to create a series of tool locations (TL) and orientations that can be used in the production of a design surface (**S**) (5).

The iso-parametric method is frequently used to assign the tool path along one of the two surface parameters and uses uniform steps along the second parameter to generate the linear transform. That is to say that the surface **S** can be defined as Equation 1 where u and v are mapping axes that have been applied to wrap the surface with a reference point defining the start position.

$$\boldsymbol{S} = \boldsymbol{S}(\boldsymbol{u}, \boldsymbol{v})$$

Equation 1 Parametric surface definition

In iso-parametric operations the tool path T(t), where t is a parameter along the tool path, tends to use a constant value for either the u or v parameter. This can be defined by Equation 2 where the tool moves along either mapped axis completely.

$$\boldsymbol{T}(\boldsymbol{u}) = \boldsymbol{S}(\boldsymbol{u}, \boldsymbol{v}^*) \; Where \; v^* \; is \; fixed$$

$$\boldsymbol{T}(\boldsymbol{v}) = \boldsymbol{S}(\boldsymbol{u}^*, \boldsymbol{v}) \; Where \; u^* is \; fixed$$

Equation 2 Parametric machining requires the machine to traverse along a single axis completely before incrementally moving along the second axis

The path interval in iso-parametric paths is based on the minimum distance between tool path intervals to provide complete coverage of a component. This system reduces the real time calculation by introducing a constant iso-parametric offset. However; since this system uses the smallest path interval, this produces excess paths and therefore introduces inefficiency to the system (46).

Iso-planar path generation sub-divides the parametric surface based on the change of gradient in the design surface (S). This allows the system to operate on similar patches rather than the complete surface where large changes may occur. This form of path planning is used predominantly within machining operations to increase the cutting rate. Where re-orientating the cutter head would be possible for undulating surfaces, it would not necessarily be efficient, iso-planar systems can maintain a fast cutter head over similar patches with small adjustments rather than large adjustments with a slower cutter head.

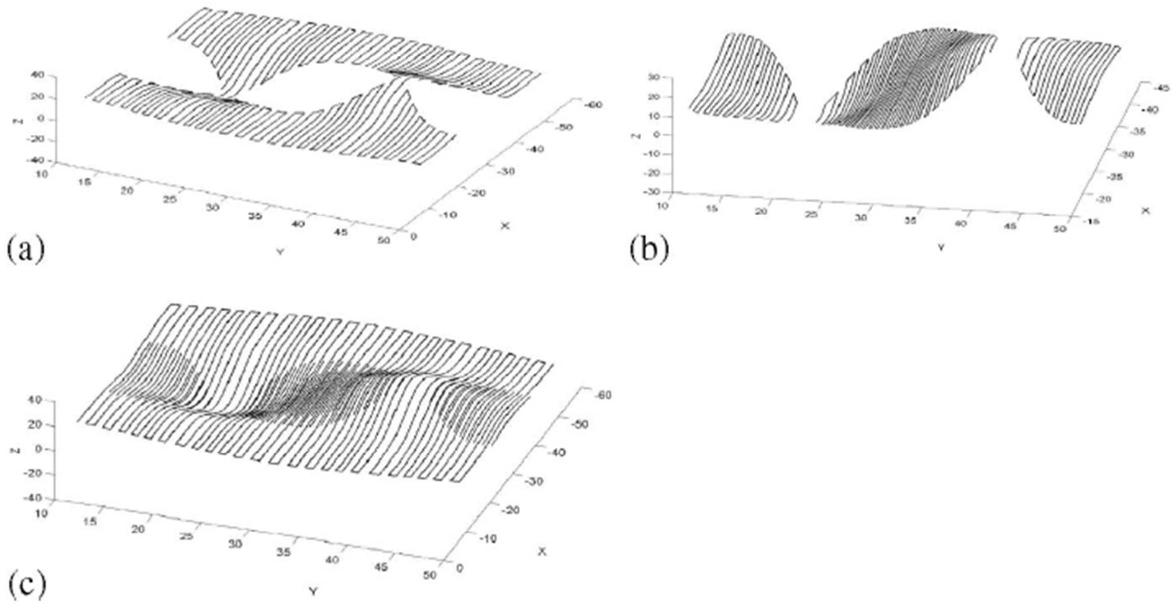


Figure 2-4 a) Low curvature patch b) High curvature patch c) Combination of separate paths (43)

Figure 2-4 shows an example of an iso-planar cutting path where the low curvature surface has a path constructed to allow a fast cutter head Figure 2-4 (a), while the high curvature surface is handled as a separate cutting path Figure 2-4 (b). This strategy can reduce cutting times when compared to iso-parametric or iso-scallop as less time is required to re-position the cutter (43) (44). Figure 2-4 (c) shows the complete surface cutting paths where the two paths have been positioned on the surface of the component. This form of planning, though it can reduce cutting times when implemented correctly, is difficult to implement as a value has to be specified defining when the model has left low curvature and entered an area of high curvature. Therefore, an iterative approach may be required to create the time curvature balance that would be required.

Iso-scallop planning has recently been introduced into commercial path planning applications as this form of planning is focused on maintaining a constant surface quality by offsetting paths based on a constant scallop height (i.e. ridge between cuts). Though the use of iso-scallop planning is relatively new, several CAM systems have quickly integrated these algorithms into the system due to the improved quality of finish that can be achieved.

The use of parametric path planning has widely been accepted by many CAM manufacturers, with most systems devising a range of different specialist algorithms combining all three approaches to provide complete surface coverage. Though many algorithms exist, it is still apparent in the literature that complete component planning is difficult to achieve with parametric planners and the majority of paths that are created are done so over surface patches rather than the complete components. This could be due to the difficult nature of building parametric meshes over complex surfaces, particularly in circumstances where the genus of the model is greater than 0 (47).

2.1.3 Configuration Space Based Planning

Configuration Space has widely been recognised since its first implementation as an effective path planning mechanism. It operates by calculating the required position of each of the joints in a kinematic chain by solving the inverse kinematic equations.

Lozano-Pérez (48) introduced configuration space in his 1980 AI Memo as a possible means to solve spatial planning problems for the positioning of a component within a constrained workspace. The characterisation of a single object in a work environment can be reduced to six dimensional points. These six dimensional points are composed of the standard positional components x,y,z with a further nine numbers used to define orientation in terms of Euler angles. Each of these three angles are defined by the rotation about the three axes x, y, z (49) (50) (51). If the work area can be reduced to a point cloud of six dimensional configurations, it is possible to separate the work area into two separate categories; free-space and occupied space. Using these categories it is possible to position or manoeuvre new objects into the work area, this process is referred to as generating configuration maps.

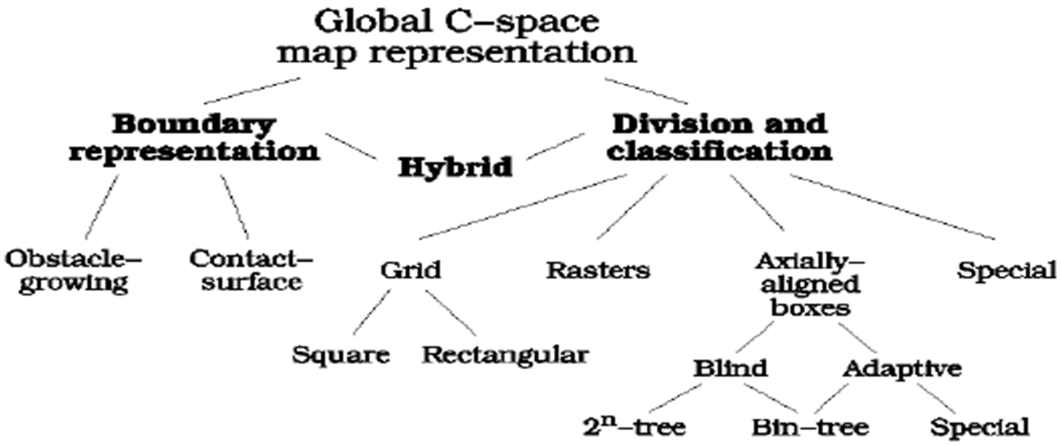


Figure 2-5 Taxonomy of C-Space representation schemes (52)

Wise and Bower (52) present a survey of configuration mapping techniques and suggest a taxonomy for defining configuration space mapping techniques by representation: Boundary, Hybrid and Division and Classification (Figure 2-5). These three main representations have different characteristics with boundary representation being the simplest and quickest method for smaller degrees of freedom as simple models can be analysed quickly. Division and classification, on the other hand, is a basic concept to understand, and significant work has been done in sub-division for other areas of research such as computer graphics. Division and classification works by breaking the work area down into smaller sizes that are more manageable. If an obstacle exists within the area, it is flagged as being occupied and is sub-divided further however, when an object does not exist within the workspace it is marked as free space and further sub-division is not required.

2.1.3.1 Boundary Representation

Lozano-Perez's initial work focused on the spatial planning problem. Using the Minkowski sum operator (48) (53) (54) it is possible to grow an obstacle to show the region surrounding it; where a moving object would meet the surface of the obstacle when it entered that region. Figure 2-6 is an example of how the Minkowski sum operator can be used for object avoidance. **Set A** is an example of a component within a workspace causing an obstruction. **Set B** shows the component that needs to navigate around the obstruction caused by **Set A**. When **Set B** in its current orientation is traced around the outline of **Set A**, a map is created showing where **Set B** would collide with **Set A**. This can be seen as the union of **Set A** and **Set B**. Further to this a second map is present at the bottom Figure 2-6 which represents **Set B** inverted and this shows the area where **Set B** would collide with the component when moving around the base of the component. Therefore it is possible to say that any area within the light grey area of the maps would result in a collision should **Set B** move into this area, it is important to note that both maps are required to identify possible collisions.

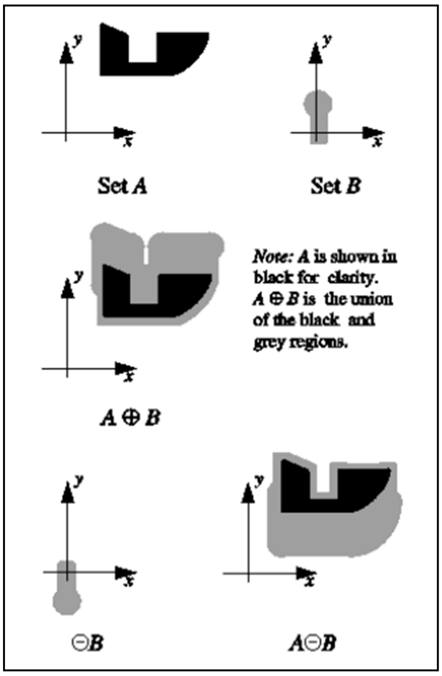
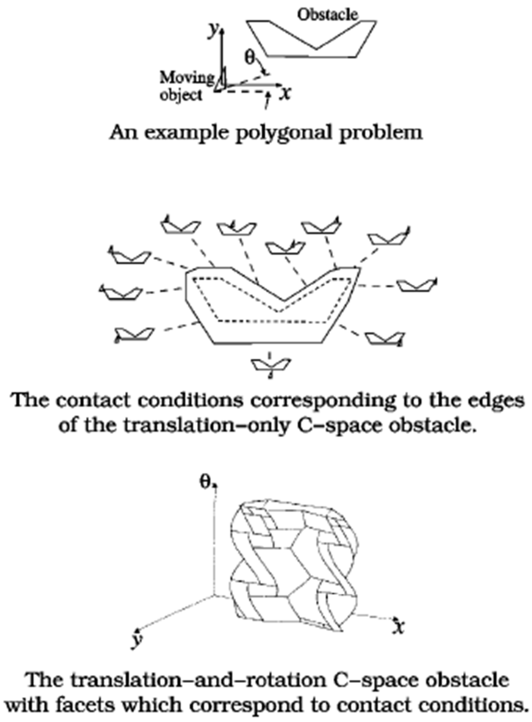

y
x
Set A
Set B
Note: A is shown in black for clarity. A ⊕ B is the union of the black and grey regions.
A ⊕ B
⊖B
A⊖B

Figure 2-6 Minkowski Sum Operators (55)

Although in principle this technique works successfully over simple polygons where the moving object operates in a translational method, to extend this to a more general 3D method, Lozano-Perez suggests the use of sliced maps. Such an approach would use a collection of maps showing safe areas for incremental rotations. Using these maps, safe locations are found for the object to move between layers. Lozano-Perez's approach could be extended to encompass full 3D objects and 3D obstacles (56) (57). This would require a large amount of processing to handle complex 3D Minkowski sums and significant amounts of memory would be required to store 3D maps to handle different slices for each rotation.



An example polygonal problem

The contact conditions corresponding to the edges of the translation–only C–space obstacle.

The translation–and–rotation C–space obstacle with facets which correspond to contact conditions.

Figure 2-7 Extension to the Minkowski Sum using rotation (55)

The work carried out by Brost (55) builds on the use of boundary representation rather than producing full 3D models of regions where collisions may occur. The construction of projection maps is used as this allows for a more intuitive method for handling rotations. Projection maps are generated by aligning the 2D obstacle on the XY plane, making sure that none of the vectors are parallel with the y-axis (this would prevent it from being visible on a projection on the X-theta plane). Then regions are generated in accordance with the different types of contact that would be achieved by rotating the colliding obstacle as in Figure 2-7. The simplest form of contact is an edge, vertex contact or vertex edge. However the algorithm also handles (edge, edge), (vertex, vertex) and discards areas where the contact is impossible due to exterior interference. Once the model has been analysed, a projection can be generated for any value along the y-axis. For convex objects there tends to only be a need for two projections, front and back, as these hopefully contain all the information required to plan a spatial path around the object.

Figure 2-7 shows an example of how Brost builds a 3D description of contacts between a moving component and an obstacle. The approach is similar to the one presented in Figure 2-6, however, rather than the component being fixed to a reference point which moves around the component resulting in the need for two maps to describe possible collisions in a single orientation; Brost approached this differently. Brost moves the entire component around the obstacle maintaining the same orientation, however the component never enters the obstacle space as the Lozano-Perez approach did (i.e. requires the second map for the lower side of the obstacle). To convert this to a 3D approach, Brost then rotated the moving component incrementally around the z-axis resulting in a 3D map with X-Y and theta as the axes. For quick interrogation of the model, Brost proposed that where the contact changed a new facet should be created on the 3D model. For example, where an edge to vertex contact changed to being an edge, edge contact a new facet boundary should be created.

It can be seen from the literature that boundary representation can be implemented successfully for simple 2D path planning and obstacle avoidance tasks. Such applications could be lawn mowing, translational robotics, pick and place tasks and other simple applications. Lozano-Perez and Brost both demonstrated different approaches for handling rotational movement. The different approaches proposed could be extended to full 3D robotics though this would require significant processing to store simple geometric shapes. (54) Although this would be unfeasible in a practical application, it has been noted by Fogel and Halperin (58) that the processing does not need to be exact in most situations, as an approximation for movement and placement algorithms would normally be sufficient.

2.1.3.2 Division and Classification

The approach of division and classification is different from boundary representation as it analyses the complete workspace rather than just the area around an obstacle. One of the first sub-division algorithms (59) operated using the Minkowski Sum (54) (56) approach presented by Lozano-Perez (48), rather than generating a graph, devised a cell structure that held information about the work area, indicating if the cell was full, empty or mixed content. Once the work area is sub-divided into cells, a path can be devised passing through only the cells that are classified as empty. When no valid route is available through entirely empty cells, then the mixed cells are sub-divided further until a valid path is discovered or a specific number of sub-divisions have occurred. This approach by Brooks and Lozano-Perez extended Lozano-Perez's earlier work on configuration space and handled rotation in a similar fashion (see earlier in this chapter). With the generation of a single sub-division that would accommodate a single angle a stack of sub-divisions were required to handle the complete angular rotation. This data could be arranged on a graph with X, Y and Theta axis and then movement around the obstruction could be devised depending on the rotation of the work piece.

Many different sub-division algorithms exist (which Wise and Bower surveyed (52)) and have been implemented in a selection of different applications particularly within the computer graphics field e.g. shadow algorithms in computer graphics (60). The simplest of all mechanisms for sub division is a conventional grid with uniform sector size. Denhe et al. (61) implemented a virtual grid mechanism where the sectors are sub-divided and handled by a specific processor (building on technology transferred from the image processing community). This reduces the burden on a single processor and can increase speed, however, it is not scalable to an industrial environment and does not distribute work evenly across the mesh of processors.

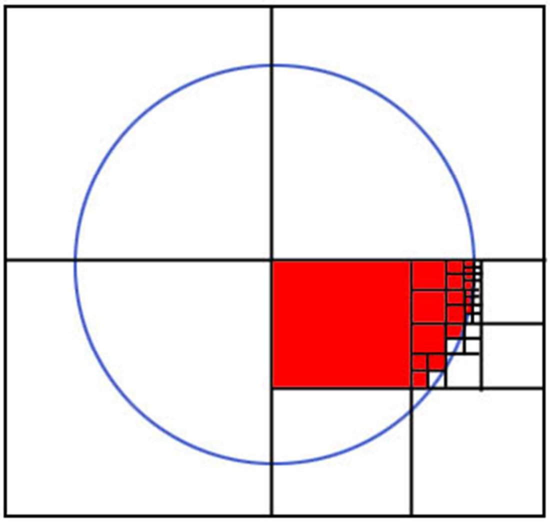


Figure 2-8 Quad-Tree representation for cell sub-division (62)

A more commonly implemented approach is to use a spatial tree structure (63) Figure 2-8, where cell size can vary depending on occupancy. This reduces the memory requirement to store information and can allow a quicker calculation of movement as larger cells can be moved through quickly. Figure 2-8 shows an example of a quad tree approach where the image has been sub-divided into quads and on the lower-right, the quad has been further sub-divided and classification has also been started showing empty and full quads. Zhu and Latombe (64) generated '*rectangloid*' cells (dimensionally dependant) similar to quadtrees when 2D, or octrees when 3D, and sub-divided the work space into progressively smaller rectangloids each classified as EMPTY, FULL or MIXED. Zhu and Latombe identified a bottleneck in the traditional process of sub-division, as both sub-division and classification were done as separate steps and the sub-division was not optimised. This meant that significantly more mixed cells were produced; the production of mixed cells meant more comparisons to the obstacles that took significant time to process. A new approach was proposed, using a union of the bounding approximation and simultaneously sub-dividing and classifying the cells, this reduced the time and increased the number of EMPTY and FULL cells created.

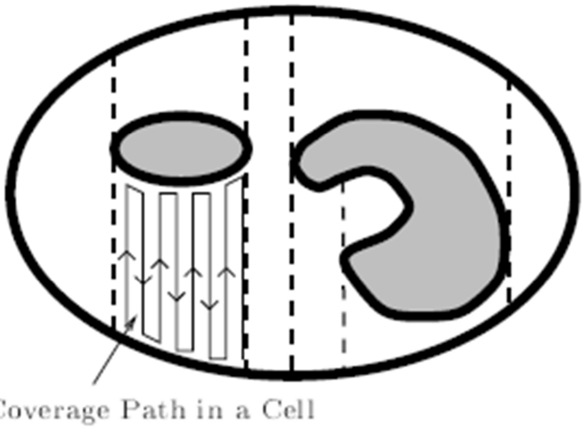


Figure 2-9 Cellular Decomposition of Workspace with single obstacle (65)

Acar et al. (65) use an exact cellular decomposition to define regions; each region is formed using a slice method. Figure 2-9 shows an example of this, a plane is swept through the entire workspace to break the area down into the regions indicated by the dotted lines. Where the plane is broken can be classified as an obstacle therefore a region boundary is created indicated by the dotted lines. Once the entire workspace has been sub-divided, each of these regions can have a path generated within it or a path that avoids the obstacles can be constructed. This method has been implemented in a 2D environment; however, it could be extended to a full 3D workspace using a 3D connectivity graph. Although the method was not specifically designed for configuration space approaches, it could potentially be extended using a Minkowski sum to create an offset boundary where collisions would definitely occur. Though (65) is not specifically designed for configuration space planning, it does provide a good understanding of approaches that have been taken for 2D inspection and mobile robotic planning. It does address a similar requirement for a surface coverage algorithm for all accessible points, though this system was devised for environments where obstacles are known about before implementation.

2.1.3.3 Hybrid Approach

A subset of techniques exists that do not conform to the traditional methods of boundary representation or division and classification. Most of these have been devised for specific tasks and are not easily extended to more general problems. One technique that has been developed for general problems approaches the topic with a fresh perspective. Rather than identifying the restrictiveness of the robot, performing a union in Cartesian space, then performing a search for available paths through configuration space Zhao et al. (66) analyse the obstacles in configuration space and build a map of occupied joint angles Figure 2-10.

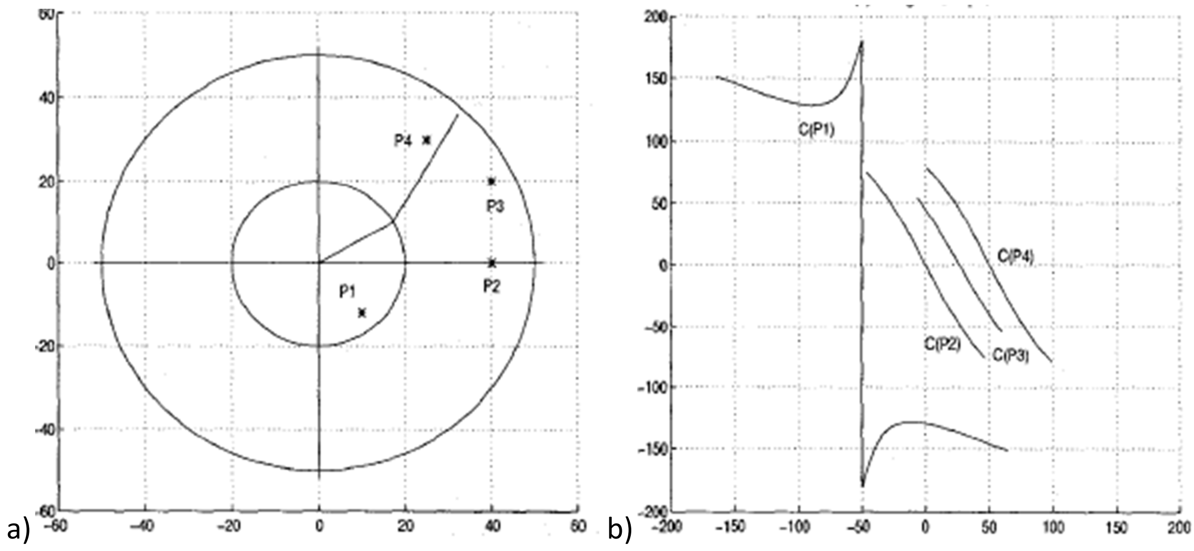


Figure 2-10 a) Robotic reach and point obstructions b) Joint angles where point obstructions exists (66)

Figure 2-10 shows a simple example of the work carried out by Zhao et al. On the left-hand side a map shows the reach of a two jointed robot where the base is located at (0, 0), various points P1, P2, P3 and P4 have been placed in the workspace representing single point obstacles. On the right-hand side of Figure 2-10 is a graph where the x-axis represents the joint rotation at the base and the y-axis represents the joint rotation of the second joint in the robotic chain (this robot is similar to a scara robot). The graph shows all the joint positions where the robot would collide with each of the points. It can be seen from the graph that P1 occurs within the reach of the first joint, therefore all solutions for the second joint would result in a collision this can be seen at approximately -50 degrees. Along with the possibility of a collision at -50 degrees, other positions for the joints could also result in a collision; these can be seen by reading the joint 1 rotation of the x-axis and the joint 2 rotation of the y-axis. In a similar fashion point P2, P3 and P4 can be seen to collide when the robot obtains a certain configuration, however, this does not occur with the first joint angle but with the second joint, therefore no single orientation for joint 1 would cause a collision.

This approach has been expanded further, but instead of looking at a point obstacle, 2D linear obstructions have been looked at. Where 2D obstacles are identified, rather than a single line being created on the graph showing joint collision, an enclosed area is created representing a group of joint angles where a collision would occur. This approach although it is initially computationally expensive would allow a visual interpretation of free space to be seen with reference to joint angles. This technique also takes into account the possibility of robotic arm collisions and generates the solution set where no collisions occur. The main drawback of this approach is that its expandability to further degrees of freedom is very limited. However, for 2D pick and place applications this approach would appear to be very good. Once the collision map has been created, constraints could be applied to the robot to prevent collisions.

2.1.4 Path Planning Conclusion

Though several different path planning approaches have been presented, it is apparent that each approach has its own unique benefits. The Cartesian system is simple to implement and is readily used in 2.5D milling operations; however, when it is extended to 5-axis machining the construction of complete surface paths is difficult. Parametric path planning is readily used in 5-axis machining as computer models can be used to generate paths; complete surface paths are problematic because of the difficulty in parameterizing arbitrary bodies containing through holes. Configuration space on the other hand requires significant pre-processing but yields efficient paths. So, given the complex nature of the optical painting system, and the need for a robust and effective path that can adapt to various robotic configurations, this thesis will concentrate on building a C-Space approach that can be tailored for each specific robot.

2.2 Review of Technologies for Planning Mass Customization of Product Appearance

2.2.1 Introduction

The optical painting process can be optimized for several different parameters, such as speed, set-up time and quality. Each of these parameters implies distinctly different approaches to tooling and components that the system will use. To determine what the technological priorities are, this section surveys the industrial context of this work.

The commercial motivations of this work arise from the trend in manufacturing towards mass customization. The use of the term mass customization has widely been acknowledged as originating from Davis (67) who in 1987 published a book called "*Future Perfect*" which describes how a market could be broken into small segments and produce individual products and services for the individual customer. The term has become a goal for many manufacturing companies as they strive to produce tailored products at a similar price to mass-produced products.

In reality, the term is used to describe a manufacturing strategy that is very simple to understand but which, if fully implemented and automated, would result in a very profitable company. Consequently many researchers have investigated the "*generic*" formula or methodology required to produce mass customization facilities for different industries. However, progress towards a general set of guidelines has been slow, with successful approaches being hard to reproduce in different domains.

For example, one of the major backers of mass customization has been the automotive industry which identified, from an early stage, the benefits of the successful implementation of this manufacturing technology.

Manufacturing technologies reflect varying social and economical factors as well as the technological advances made in production machinery. For many years, the users of mass and batch production were predominantly large-scale manufacturers with very little customer input prior to production. More recently, a combination of modular construction and Internet technologies allowed several manufacturers to provide a degree of Mass Customization. The customer selects from a limited parts family and standardised connections or specifications allow for a more individual and bespoke product (e.g. Dell Computers, BMW cars). To date, the majority of mass customization has manifested through modular products that are specified at the build. This has opened a limited market to companies who are able to integrate products into a product after the build stage an example of this would be GPS (68) units being added to cars after construction.

Svensson and Barfod (69) propose a framework, seen in Figure 2-11, where a product without flexibility from design through to distribution is classified as standardized, whereas a product that is flexible at every stage is bespoke or classified as pure customization. The framework suggests that everything in between is a level of mass customization. Svensson and Barfod go on to propose categories for each level of mass customization (Segmented, Customized Standardization and Tailored Customization). Though this is just a framework, it is interesting to note that the examples previously given (Dell and BMW) would both appear to fall into the Customized Standardization Category as the fabrication does not change, just the products used in the assembly.

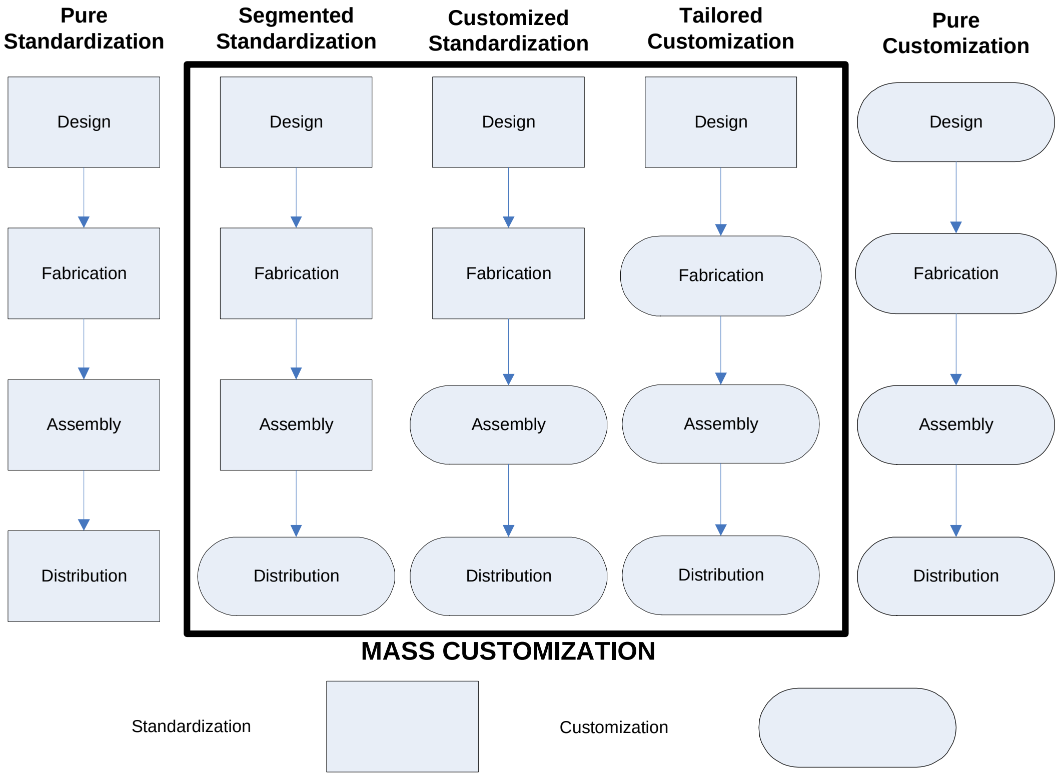


Figure 2-11 Degrees of customization (69)

The following sections review the different types of customization that are currently available, specifically highlighting the virtual world's contributions and how the market has adapted to these to provide the consumer with a more targeted approach.

2.2.2 Mass Customization by Component Selection

The simple premise behind mass customization by component selection, is that for a general product constructed from discrete components, the consumer can stipulate the modules that they require from a standard list. This idea has been around for many years with the aircraft industry using this system within their ordering mechanism. A standard fuselage is selected and then the interior can be equipped based on the consumer's requirements, however, the aircraft industry is not a mass market. In recent times, manufacturers in mass production markets have adopted a similar approach. Perhaps the most successful is Dell computers with other examples including BMW cars, Sunseeker boats and UJeans trousers. This approach requires interchangeable components and therefore specific design constraints need to be placed on interfaces between components. The problem also requires companies to consider the extent of range that they will offer the consumer as this will depict the amount of stock that the manufacturer needs to hold (70).

The case study by Kramer et al (71) presents the technologies that Dell implemented in capturing the personal computer market. It suggests that though the system is successful for Dell, this has as much to do with the conditions that afflict the PC market. The case study cites the rapid technological changes that occur within the PC market as a significant stumbling block for any new venture. The traditional PC companies adopting mass production meant that large amounts of stock were required and consequently there was very little flexibility within the system preventing the quick introduction of new technology.

The mass production market requires a large outlay for the latest products that are not guaranteed to sell at premium prices. It has even been shown that in most cases consumers are not willing to wait for new technology especially when the support network is limited. Therefore, the consumer settles for the older technology for the reduced price and improved support (72). The distribution network adopted by the conventional mass production retailers also has some intrinsic faults. The conventional approach to selling computers is an indirect distribution channel where production companies sell to a distributor and the distributor sells to retailers. This system requires stockpiling products at each level and reduces the speed with which a product can reach the market. It also increases the risk at each level, as financial outlay is required for products with a greater risk of not selling them all.

The distribution model that Dell implemented required more trust in IT systems and a greater outlay that would yield higher profits and eventually revolutionise the computer market. Using a direct sales network where the production company sold directly to the consumer, Dell removed two sections from the traditional indirect system. The direct model relies on consumers ordering directly from the production company and products are then distributed to the consumer. This system removes conventional retailers and requires customer confidence in the production company. It is important, using the direct sales model, that consumers maintain confidence in the production company, therefore the need is to deliver a service that conventional retailers cannot provide. Dell sold its products on its ability to provide the latest technology and a computer built to requirement; this was only possible due to the modular nature of the computer market with standardized interfaces and components.

The introduction of Internet sales provided an ideal platform for production companies to supply '*build to order systems*', where consumers could dictate the specification for their computer from a standardized list of modules. The order could then be sent directly to the assembly floor where it could be completed. The money for the products would already be with the company before the product was dispatched, reducing the risk for the production company. Additional benefits of using the customizable system were that it allowed the manufacturer to receive marketing knowledge and computer stock control reduced the risk of holding large numbers of components before they were required. This approach does have some drawbacks; as orders are sent directly to the assembly line, when modifications or cancellations occur, they need to be rapidly passed to the assembly line. As many of these systems are computer-based and work within a strict set of guidelines, when an alternation is required this tends to be done using human interaction. Therefore, it is important that the human quickly informs the assembly line to suspend, cancel or modify an order.

Several production companies have adopted a similar approach to Dell. However, very few have committed themselves as far as Dell. Dell have withdrawn completely from the conventional retailer market, allowing them to concentrate completely on their custom-built systems. Though companies such as Land Rover, Sunseeker and Portakabin have also initiated a build-to-order system they still maintain the use of conventional retailers and distributors. This may be due to the nature of the products, as with certain items the consumer still wishes to try before they buy.

2.2.3 Mass Customization by User Innovation

The term user innovation has been interpreted in two very different manners by the fashion industry. Major sports companies, such as Nike and Adidas, have devised user interfaces for the customization of reduced product lines. Whereas a more revolutionary approach by Threadless (73) relies on users submitting designs, that are then assessed by the consumers and the best designs are then produced. Though both systems are different in their implementation, the general idea is very similar; the consumers want to buy products with an individual identity rather than mass-produced items.

Nike have created a limited product line know as Nike ID (74) that allows consumers to customise shoes and bags with graphics or text on selected areas. This form of customization allows consumers to create a personalised product. The system is restrictive with only small areas available for customization, and uploading personal graphics is currently prohibited due to restrictions on copyright. A similar system has been implemented for Adidas (75). Though both systems are restrictive on personal graphics, they do provide a much greater flexibility for the specification of colours and sizes. An example of the extent of customization by Adidas can be seen in Figure 2-12, where a custom set of boots has been created with a user-specified trim (i.e. the tongue colour) and text (i.e. “Burns”).

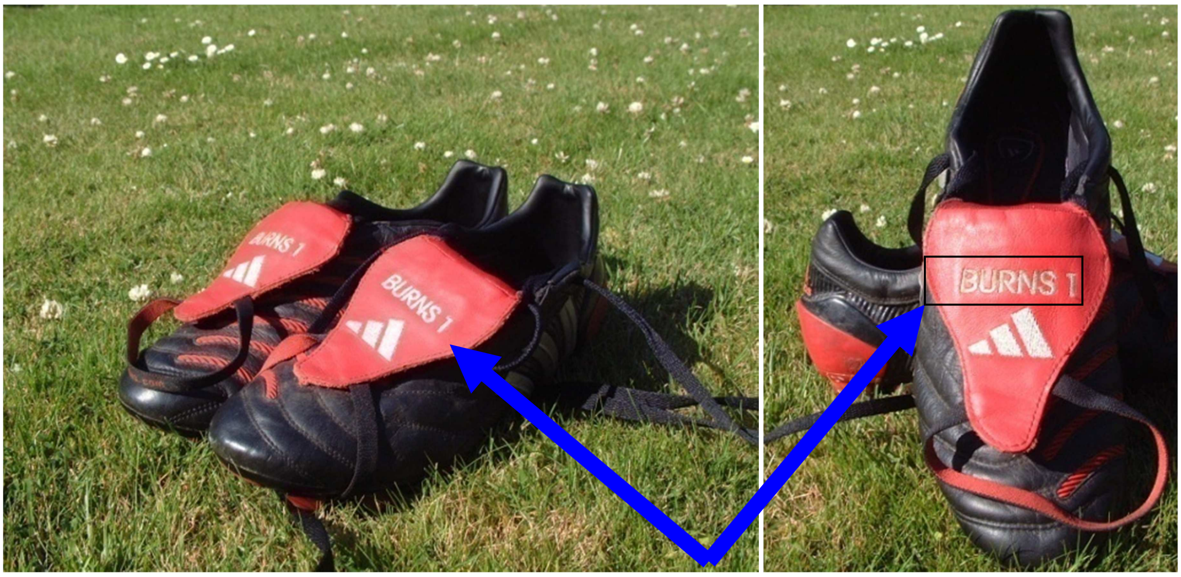


Figure 2-12 Personalised rugby boots from adidas

The models that Adidas and Nike have used are based on a negative “*cash conversion cycle*”, that is to say the money is received by the company before paying for the material to produce the product (the same as the system used in the Dell model). This reduces the risk for the manufacturer by removing the need for stockpiling an inventory. The product also attracts a price premium due to its individual nature and, as the product is delivered through a web interface, the retailer mark-up is removed.

The second form of customization is based on collective, rather than individual preferences identified by Eric von Hippel (Author of *Democratizing Innovation* (76)*)* as a natural progression for product development in the Internet age of free mass communication. Democratic customization is well illustrated by Threadless (77) a company based in Chicago; they distribute t-shirts and have grown quickly with revenue increases of 500% in one year (2006). They developed a social network for the creation and evaluation of their products. Each month online galleries allow people to evaluate designs and rate them using a scale of 1-5. At the end of each month, the highest rated designs are then turned into products with the designer receiving a small payment and vouchers for use on the site. The marketing of the products is done using the social network that has been built around the company, with each designer encouraging friends to vote for their design and so visit the site.

The strategy that Threadless has adopted is synonymous with open-source software where the consumer shapes the final product. This delegates innovation to the consumer and allows the manufacturer to produce what the market wants. Jake Nickell, the co-founder of Threadless, describes the strategy as based on common sense in giving the consumer what they want. Production is led by consumer interaction and trust in the knowledge gained from the social network allows them to forecast sales.

The Nike model and the Threadless model are both successful implementations of user interaction within a mass production market, though they both take very different strategies. The Nike model relies on the standard nature of a base component which can be easily modified. The Threadless model relies on the production of standard products specifically tailored for its customers that have been evaluated before production therefore ensuring a guaranteed market.

2.2.4 Mass Customization by User Interface

Perhaps the newest method for mass customization is in the virtual world where interfaces can be adapted for specific purposes (i.e. CAD designers require different buttons in an application to finite element analysts). Modifying user interfaces has been around for many years, the software that brought media attention to this form of customization was '*Winamp Media Player*'. '*Winamp*' (78) was released as a free application that allowed the consumer to play media files and generate playlists of their favourite music. The feature that set it apart from other software was that the developers openly released software for modifying the layout and appearance of the graphical user interface; this was to be known as '*Skinning*'. Although other "*skinable*" products pre-date '*Winamp'*, the ability to '*Skin'* an application had traditionally been a complex programming task, involving large quantities of coding and demanded lots of time to produce good results.

Since '*Winamp's*' release in 1997, the demand for customizable interfaces has grown rapidly with technology giants such as Microsoft leading the way. Microsoft Office (79) has traditionally provided a large selection of tools within its applications, however, the interface to use these tools has sometimes been deemed as confusing. In 2007, Microsoft released a new version of the Office product family featuring '*The Ribbon'* which simplified the interface by providing the most commonly used buttons on the standard view. The software also enabled users to customise the layout and the aesthetics with much more ease than previously. Additionally the creation of customized ribbons was added so that consumers could create their own working environment rather than relying on the standard views. Microsoft further enhanced consumer experience with new web technologies that enabled users to customise websites that they visit regularly, such as the BBC website (80) that uses the '*Web Parts'*. These are a set of controls that enable the consumer to position and display information they require rather than viewing a standard layout. The technology relies on the creation of a local XML file that defines the custom user interface which is accessed every time the consumer visits the website.

The use of customization in virtual environments has grown rapidly with almost all new applications allowing some form of user influence on the layout and appearance of buttons.

2.2.5 Mass Customization by Geometric and Function

The mass customization of a product's geometry and functionality is probably the hardest form of customization to perform economically with very few commercial examples having succeeded.

Perhaps the most notable firm to have come closest to this form of customization is IKEA (81). IKEA provide a unique range of furniture that consumers purchase in a variety of different shapes and sizes. Although this is similar to mass customization by component selection, the major difference is the degree of geometric variation possible through the flexible configuration of standard base units. IKEA furniture does not fully fulfil the criteria for customizable functionality, as the general function remains the same (e.g. storage) (82) (83). Other products that allow variable geometry include dental components, medical components and toys.

The market for products with customised geometry is being enabled by the emergence of new manufacturing technologies like rapid prototyping (RP) making it possible for consumers to view a mock-up of the final product before production. Rapid manufacturing has evolved quickly, as rapid prototyping technologies produce parts closer to final working components, the boundary between prototypes and fully functional parts is closing. For example, micro-milling machines originally designed to produce small prototype replicas are being refined for the construction of jewellery (84). The manufacturing processes used in RP traditionally used layered manufacturing approaches which remove the intricacies associated with surface machining operations such as milling. This allows a more diverse range of products and designs to be developed, which in turn, provides a much greater range of products for a consumer. New technologies such as micro milling are opening new avenues for prototyping as well as accelerating production times. The ability to mass produce items using rapid prototyping is not yet economical for most products due to the relatively slow nature of RP technologies (85).

2.2.6 Mass Customization Requirement for Optical Painting

It appears likely that mass customization as a manufacturing technology will continue to increase as customers demand better and more tailored products to fulfil their needs (86). Mass customization will not replace mass production or individual production, as there is still a need for goods which are identical or specifically designed for a task.

The use of mass customization will slowly bring about a change in the way that products are designed; this is already evident in the market place (87) where computer components have moved to standard connections and mobile phones have interchangeable covers. The use of modular construction and standard interconnections allows a greater variety to be developed, resulting in a more competitive market place, and allowing the customer to select the components based on their specific attributes. The diverse range of components being designed will segment the production environment where modules are constructed separately from the assembly plants of mass production.

Though the four categories that have been used to summarise mass customization are broad, (indeed certain manufacturers or products may fit into several different categories) the general overview provides a breakdown of current technologies. It is clear that the ideal manufacturing unit would encompass all facets of customization (i.e. shape, appearance and functionality). However, current technologies only allow tightly focused customization within the manufacturing market; allowing general product families to be adjusted to a consumer's needs.

Although the financial outlay of implementing any mass production system is initially very large due to the large amount of tooling required, this can be recovered across the high volumes produced. The difficulty for a mass customization production system is that the time required to create tooling reduces the volume produced and therefore the cost per unit increases. However, '*tool-less*' production methods would remove this overhead and allow manufacturers to customise the shape and appearance of products economically. But if the physical surface appearance is to be produced automatically from a digital model, then the representation of both shape and appearance must be clearly defined and understood by the system. The next section reviews the technologies of surface appearance.

2.3 Shape and Surface Appearance

Surface appearance is an important factor in the design of new products where consumers demand quality aesthetic appeal as much as quality manufacturing. The surface appearance of a product can be visualized in two ways; a virtual model displayed on a computer screen or a physical prototype. These types of visualization can be sub-divided further as flat two-dimensional or three-dimensional models. Traditionally computers have struggled to produce virtual models that could rival physical models due to the relatively low resolution of screens and the difficulty of computing the effects of lighting and other natural effects. Academic work has developed many different approaches to virtual surface mapping "*Computer Graphics Principles and Practice*" summarises much of the fundamentals of computer graphics (49) and "*GPU GEMS*" compiles the most recent papers on applications that exploit dedicated geometric hardware into an easily accessible text (88).

As the demand for greater product selection has increased, manufacturers have needed to adapt production techniques to provide many different colours or appearances for similar products. This section will first look at the virtual techniques that are used to produce photo realistic virtual models and then will address the manufacturing methods that are currently available to create physical surface appearances on components.

2.3.1 Introduction to Virtual Surface Mapping

Any system used to customize the surface of a component will need to access the digital representation of a CAD model's appearance. This section reviews the graphics technology used to create this representation. The technique of mapping an image onto a three-dimensional virtual model was pioneered by Catmull (89) and subsequently refined by Blinn and Newell (90). The techniques that they proposed in the 1970s are still in existence however, further refinements have been undertaken. It was proposed by Catmull that a pixel map could be "*placed*" (i.e. mapped) onto a surface and rendered to give greater realism for virtual three-dimensional objects. Prior to this technique, the use of surface detail polygons was used. This approach used the shading of many individual polygons to create an image and this was neither accurate nor speedy. The modelling of a field would require individual polygons to be created for each blade of grass and hence to represent wind moving grass in a field would be time consuming and un-economical for most applications.

With continued research into image mapping techniques (91) (92) a variety of different methods have emerged which can be used depending on the specific requirements, perhaps the most significant step forward is the development of tactile surfaces. A tactile surface is a three-dimensional representation of a surface and utilises more than just the two vectors in the x and y direction, previously associated with texture maps. The introduction of a z-axis allows for a feeling of depth to be included in the mapping of the surface. This increases realism and potentially (for haptic interfaces) introduces another primary sense (touch) so that the human can relate to the surface in a more realistic way than conventional computer graphics.

2.3.2 Surface Representation Overview

The representation of 3D models is achieved in several different ways. The four principle methods used in computer graphics are; constructive solid geometry (CSG), boundary representation (B-REP), spatial partitioning and 3D meshing. Each of these has a different set of procedures for the construction, manipulation and storage of 3D components (49).

Constructive solid geometries use simple primitives such as rectangles, cylinders, prisms, spheres and cones, and perform operations using Boolean operators such as union, difference and intersection to construct 3D models. The model is stored using a tree structure, therefore a clear construction history is always present within a CSG model and changes can be made quickly by modifying steps in the tree. CSG models provide a high level of accuracy due to the exact nature of the primitives. The tree structure ensures a compact and efficient data management structure, though the construction of complex curved surfaces can be very difficult (using CSG).

A boundary representation uses surface boundaries to represent 3D shapes. The storage of B-Rep models is achieved using topological and geometric data. The three principle topological data items are vertices, edges and faces. B-Rep models use the idea that faces are infinitely large and are clipped using the edges which are constructed from the vertices.

Spatial partitioning (49) (93) (94) is a method for storing 3D components that relies on a finite workspace that can be sub-divided using one of several algorithms (i.e. cellular decomposition, spatial occupancy enumerator). Spatial partitioning breaks the workspace down into smaller volumes called voxels which are stored as either occupied or free. Most algorithms also allow partial occupancy by further sub-dividing the voxel into smaller volumes till a given resolution is met. The use of spatial occupancy relies on storing information about the complete workspace and therefore can be inefficient and is also an approximation of a component. Collision detection and other rendering operations can be performed quicker using a spatial occupancy system.

The final representation uses a polygonal mesh which is comprised of small polygons linked together. The mesh can be constructed of various polygonal structures, though in practice triangles and quads are usually used. The mesh tends to be stored as a geometric and topological structure using vertices to store the position of points and a connectivity map to store the connections between vertices. Polygonal meshes require large point sets for curved surfaces to provide a good approximation and require surface normals to be explicitly stipulated. The storage of 3D models tends to require large point sets and therefore can be inefficient. Optimization algorithms exist to remove redundant point information (95).

This representation is associated with surface scanning systems where it provides a means of producing digital models from physical models; this allows surface defects to be identified and provides a more accurate representation of the mass produced components. The data that is obtained from a surface scan combines both the geometry (i.e. shape) of the object and a texture map recording its surface colour at each point.

2.3.3 Overview of Texture Maps

While working with texture maps it is important to understand the terminology that is used to describe various different aspects of an image. The main cause for misunderstanding comes from the description of the coordinate system. When referring to an object's position, the world coordinate scheme is used. This positions an object relative to an arbitrary point which can exist anywhere and is only a reference. When the object is using a texture map, a second coordinate scheme is normally used which is referred to as the "*object*" coordinates where an arbitrary point on the surface is used as the reference and all coordinates are recorded relative to this point. The main advantage of this scheme is that it allows the object to be rotated or moved without having to reposition the texture map.

While dealing with a texture map, the convention is to use the object coordinates as this allows the map to be positioned and fixed to the surface and allows the object to be rotated or displaced without adjusting the map. Alternatively, if world coordinates are used, then when the model is rotated the map would remain fixed over the projected area (in a similar fashion to a projector) so even if the model had been removed, the texture would continue to be projected over the same area.

A conventional texture-mapped model works by having a file that stores the object and the data related to the object such as coordinates and a second file that holds a two-dimensional image. The location of pixels in the image is specified in terms of two axes (u axis, v axis). The object file holds the reference point in object coordinates for the process of mapping, this can be anywhere on the surface. Once the start point has been identified, the files are combined by relating the map's u, v coordinate scheme to the object's relative x, y and z-axes in the object coordinate scheme.

Texture mapping can be broken down into two varieties 2D and 3D. Two-dimensional techniques generally use a mapping in the world coordinate system such as projection where the image is directly "*placed*" onto the surface. Three-dimensional techniques use a more realistic approach by distorting the image in such a way that it is fitted onto the surface of the model.

2.3.4 Two-Dimensional Approach

The use of a two-dimensional approach can be broken down into several simple variations. The most commonly used approach is projection-based (Figure 2-13) where a two-dimensional image is projected along an axis or a user-generated vector.

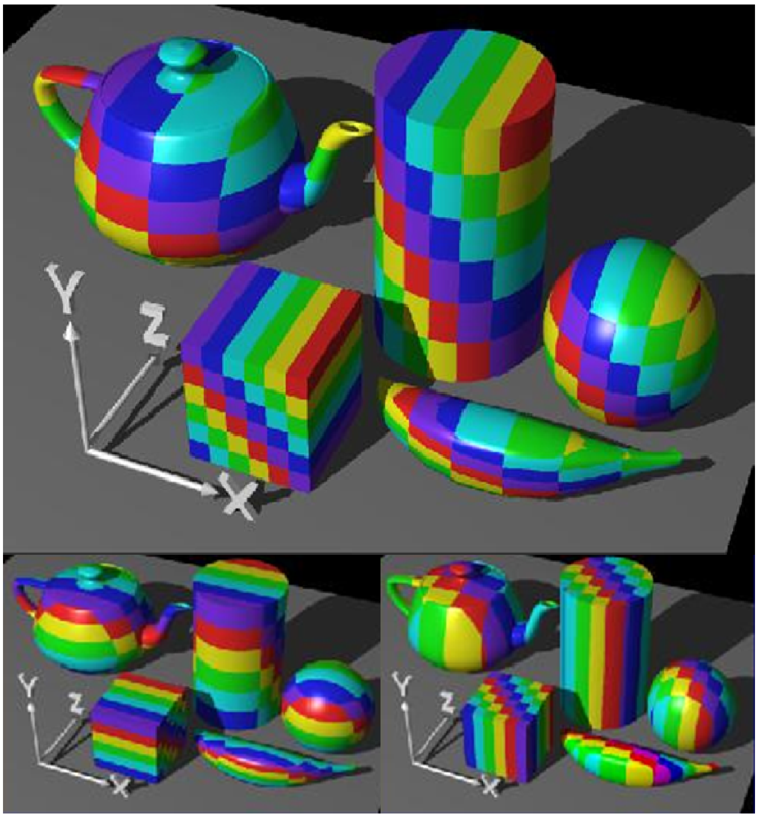


Figure 2-13 Projection mapping along various axes (Top image maps along Z-axis on the component's surface, bottom left maps along X-axis, bottom right maps along Y-axis)

Figure 2-13 shows the results of projection mapping on the surface of the component. Where the top image shows projection mapping along the z-axis; it can be seen that the colour remains constant throughout the z-axis so long as the x and y coordinates do not change. Projection mapping is not widely used because of the distortion that arises in the image from mapping a 2D image onto a 3D component where the surface is not always planar. It must also be noted that projection mapping treats a component as 2D therefore the same image appears on the back of a component. This approach also has difficulty with re-entrant surfaces, as the mapping reproduces the same colour along the entire axis of projection. However, in the context of this thesis, it may be possible to produce a simpler painting system using a digital projector for flat surfaces or over areas where a simple repeating texture may be used. Projection mapping has been used in work looking at registration, and commercial applications such as digital white-boards have used the simple mapping procedure for the calibration of the system (96) (digital white-boards project index marks on the board and request the user to touch at each of these point to calibrate the projection position on the white-board) .

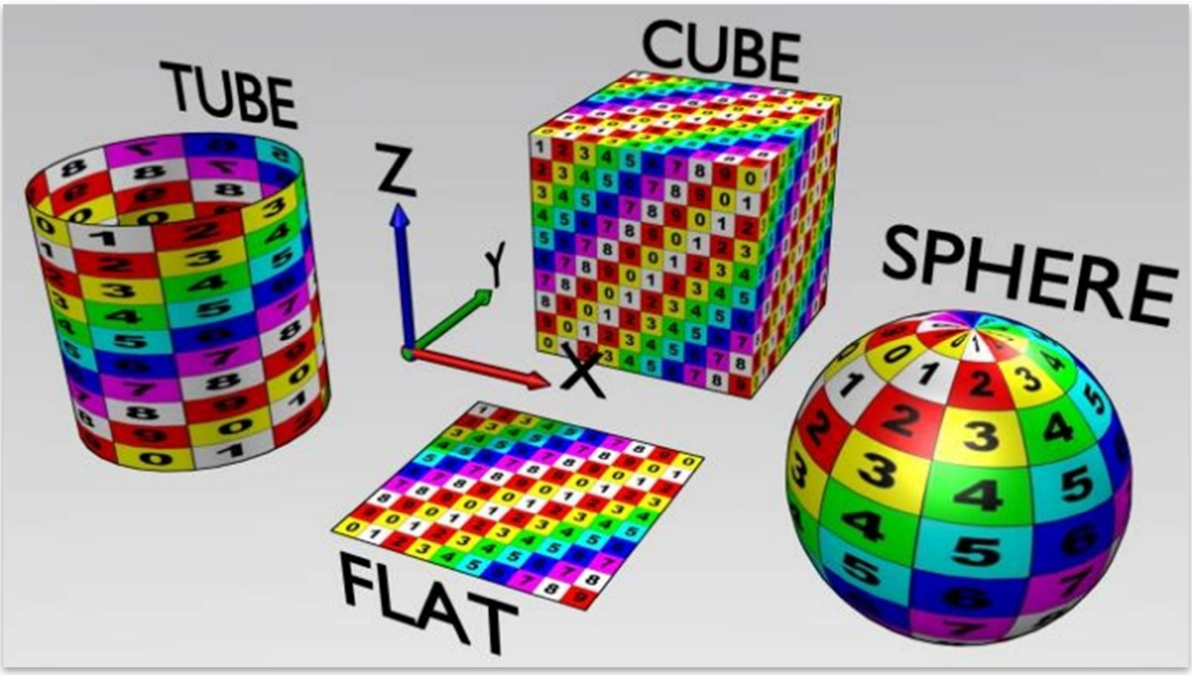


Figure 2-14 Various mapping techniques used (151) *Note the use of CGI graphics format X-Y screen position and Z representing depth of perception.

The second standard approach is a bounding volume method. This uses simple geometric shapes to surround the object. The surface of these simple geometric shapes then has the image map applied to it. The bounding volume is then mapped onto the three-dimensional shape using a transform function. One of the simplest shapes to use for texture mapping using bounding volumes is a cylinder. To use this form of mapping the coordinates are converted into radial coordinates r, theta and height. For the mapping of the image, interest is in the theta and height coordinates. To find the colour on the texture map, the theta value is converted into an x-coordinate and the height is converted into a y-coordinate.

Another shape that is commonly used in the volume approach is the sphere, as it has properties that allow for the complete surface of an object to be covered by the texture map. The use of a spherical bounding volume adopts the latitude and longitude in a similar fashion to grid references. To find the colour on the texture map the x-coordinate is mapped against the latitude and the y-coordinate is mapped to the longitude. This distorts an image as the positions near the top of the sphere are compressed more and areas near the middle are expanded. Therefore careful preparation of the texture map is required for an accurate mapping process (in map-making, elliptical maps are sometimes used to represent the earth to help to prevent scaling up countries in size near the north and south poles). The use of a box map is a different type of mapping as this still utilises a bounding volume in a similar fashion to spherical and cylindrical maps however, the use of multiple texture maps or the same texture map on several faces is possible.

The images in Figure 2-14 show a collection of mapping techniques including cylindrical, spherical and bounding box. Figure 2-15 demonstrates the application of bounding box mapping onto a complex shape (a teapot has been mapped using a wood grain texture). The use of bounding techniques is prevalent in instances where materials need to be applied to surfaces as simple repetitive patterns. However, the use of bounding techniques is not normal with photographic quality images when they need to be mapped as they often distort. A more common approach is to use projection mapping, as the images tend not to distort as much.

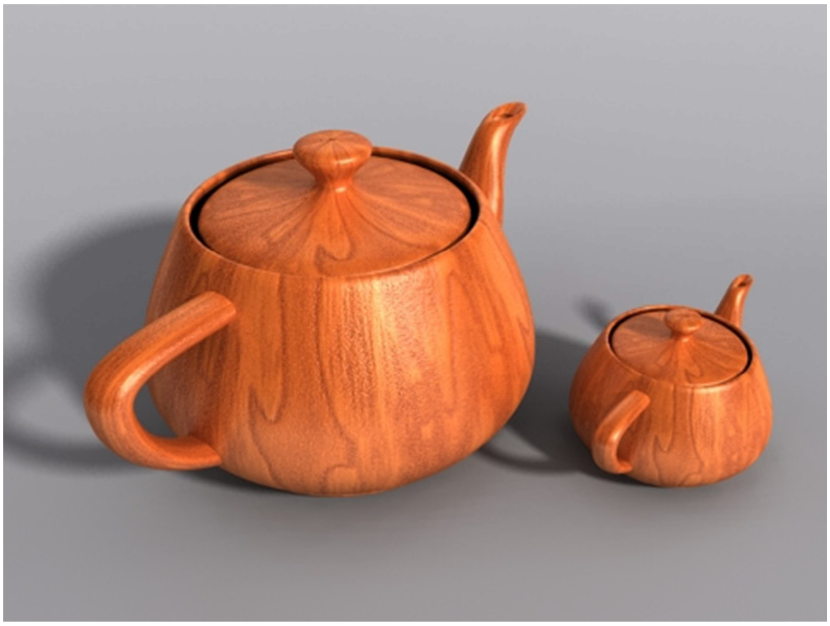


Figure 2-15 Wood grain mapping using bounding box technique (97)

2.3.5 Three-Dimensional Approach

In a three-dimensional texture-mapping scenario, each point determines its colour without the requirement for an intermediate texture file. The use of a mapping function allows the computation of the colour directly using the x, y and z coordinates. The majority of the mapping functions do not explicitly define the x, y and z coordinates, but use a function to compute the value based on the solution to the function and hence these are known as procedural textures. Figure 2-16 demonstrates 3D mapping techniques where the image uses a ramp function where the range of the function is 0 and 1 to evaluate the coordinates.

This form of mapping could be used in the calibration of a 3D photographic cell using a function controlled either by the six variables in Cartesian space (i.e. x, y, z, nx, ny, nz) or could also evaluate a function controlled by the configuration space information (i.e. Rotary joints 1, 2, 3, 4, 5 and Displacement joint 1).

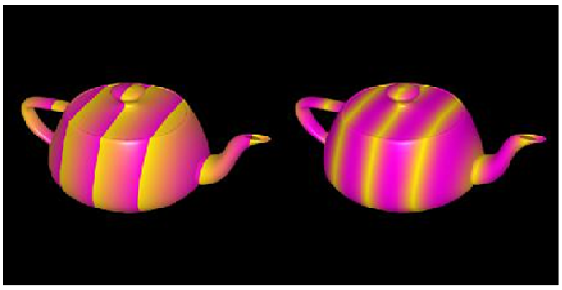


Figure 2-16 3D Mapping Techniques (142)

2.3.6 Physical Model Surface Appearance

This section reviews the processes available for physically creating images on 3D objects. Lithography has been around since the early 1800s and is achieved by transferring an image using a chemical process. Originally created by Bavarian author Alois Senefelder (98) who used a piece of limestone and an oil-based image to burn the same image onto the surface of the stone. This was the first example of image transfer from one medium to another. Though the process has changed significantly, the basics remain constant; alter one surface with the application of an adherent second surface. Other methods to alter surface appearance have been around for significantly longer, such as painting where pigment is used to alter the colour of light that is reflected by selectively absorbing colour.

Modern day physical image creation tends to be broken down into two separate categories; paint application or image transfer, each of these techniques is widely used on its own or combined in the modern toy market.

Paint can be applied to several different surfaces and comes in a variety of types (i.e. matt, gloss), it has many different applications with aesthetics only being a small subset along with other applications such as protection and stubbing (roughening a surface before finishing). Paint can be applied using several different methods. Within a mass production unit, two principle methods are used; dip coating (99) and spray painting (however, hand painting is also widely used in manufacturing environments). Dip coating tends to be performed on components that will be of a single colour and is often used on metal components requiring a thick layer to protect the base surface from exposure to the elements. Spray painting has been used in both manual and automated processes where a thin layer of paint is applied to the surface of a component. Depth of colour is obtained by applying multiple layers of paint. The use of automated spray painting is a complex process often performed within a specially constructed painting environment so that overshoot (i.e. paint that does not adhere to the surface) can be collected, minimising the environmental impact.

Transfers are a much cleaner and efficient production process than painting; they are normally constructed on 2D sheets that act as a substrate (100). The methods for applying a transfer vary but it is normally done using either heat or water; with heat being the preferred approach in the manufacturing industry. Many problems are associated with transfers being used in a manufacturing application, perhaps the most notable difficulty is that a 2D image must adhere to a 3D surface. However, the economic benefits of using transfers make the process very popular with certain manufacturers.

2.2.7 Photography

Traditional photography uses light sensitive paper to produce images on a 2D canvas. Though this system has been used for many years, improvements have been aimed at quicker turnaround times from film to paper rather than the fundamental process. The paper that is used is manufactured to a high standard and has a uniform coverage of a light-sensitive chemical (similar to "*liquid light*") (101) (102) (103) (104). Though this process is being superseded by digital photography, similar constraints exist for both digital and optical systems (102):

- Photography tends to be restricted to 2D canvases
- Paper can only be purchased in specific sizes which the manufacturer determines
- The medium is restricted to paper

In recent years, high quality emulsions have emerged which exhibit similar properties to photosensitive papers, one of the most common is known as "*silver gelatine*" which can be applied to different mediums and acts in a similar fashion to the paper.

Emulsions use the same exposure process as traditional papers where a dark room environment is required. The emulsion is first applied to the surface where the image will be exposed and then it is allowed to dry. Once the surface is dry, images can be projected onto the surface using either a projector or other image generator. The surface is then bathed in developer till the image starts to appear, at which point it is placed into a stop bath to prevent the chemical reaction over exposing the image. Once the reaction has stopped, the surface is placed into a fix bath which chemically adheres the image to the surface.

Though emulsions currently only exist for monochrome photography (i.e. black and white) a variety of techniques exist that allow colour to be introduced to an image. Perhaps the most commonly used approach is toning (104). Toning is the process of changing the colour of a photographic image and it can be done using a variety of methods, however chemical toning is perhaps the most common. Though toning tends to introduce a single colour, the selection of colour can dramatically change the appearance of a picture. Other variables, such as when the toning is done, have similarly dramatic effects (see appendix 1 for toning and uniform patterns) (103).

2.3.8 Conclusion

Current methods for physical image reproduction on 3D surfaces are either labour intensive or inflexible. This is in sharp contrast to virtual methods where a variety of different surface rendering algorithms exist.

The use of texture maps has increased and a large number of file formats have adopted a two-file approach for two-dimensional texture mapping. Also, as the complexity of models has increased, so has the computational power required to render models. The development of a two-file approach allows the manipulation of a three-dimensional model without the constant wait while the complete model and surface are rendered and re-rendered. The latest generation of computer graphics cards specifically handle the rendering of images and graphics to reduce the load on the computer processor. Although this is an exciting area of research time constraints have prevented it from being looked into in great detail.

The use of VRML (Virtual Reality Modelling Language) files on the Internet; has increased the publicity of the two-file approach due to the flexibility of the VRML file format, which can now implement interactive graphics on a model surface. The use of sounds and movies on the surface of a model can demonstrate the effect of projecting a moving image onto a model's surface, therefore identifying distortion that will be present when this application is realised in the real world rather than the virtual world.

The majority of texture mapping is still carried out using two-dimensional approaches. Where the application is the computer games industry the use of the volume techniques such as bounding boxes is prevalent particularly in '*first person shooting games*.' Projection techniques are widely used in the home (or DIY) texture-mapping environment, where images are projected onto an object then printed onto heat-sensitive paper and applied to a real world object, for example mug printing. The format adopted for the optical painting system is VRML 3D because of a simple and well-documented structure and wide spread use on the Internet in graphical applications.

Chapter 3 The Optical Painting System

The optical painting system was developed as part of a research project within Heriot-Watt University's Digital Tools Group, who aim to produce software tools that improve productivity at different stages of the product life-cycle. The system was motivated by the need for the mass customization of rapidly produced parts (i.e. personalized cases for consumer products). This market is currently growing as rapid prototyping and manufacturing technologies converge. Despite advances in materials and accuracy, most painting of models is still done by hand.

The optical painting system aims to emulate the flexibility of layered manufacturing systems by providing a technology that is;

1. Highly automated (so requires little operator experience).
2. Produces production quality images on objects with complex shapes.

With these aims in mind an experimental rig was created (using a 4-axis robot to manipulate high-powered LEDs and some custom software) to investigate the feasibility of an optical painting approach. The results of the test showed that potentially it would be possible to generate images on the surface of complex geometries by exposing a photographic emulsion applied to an object's surface (105).

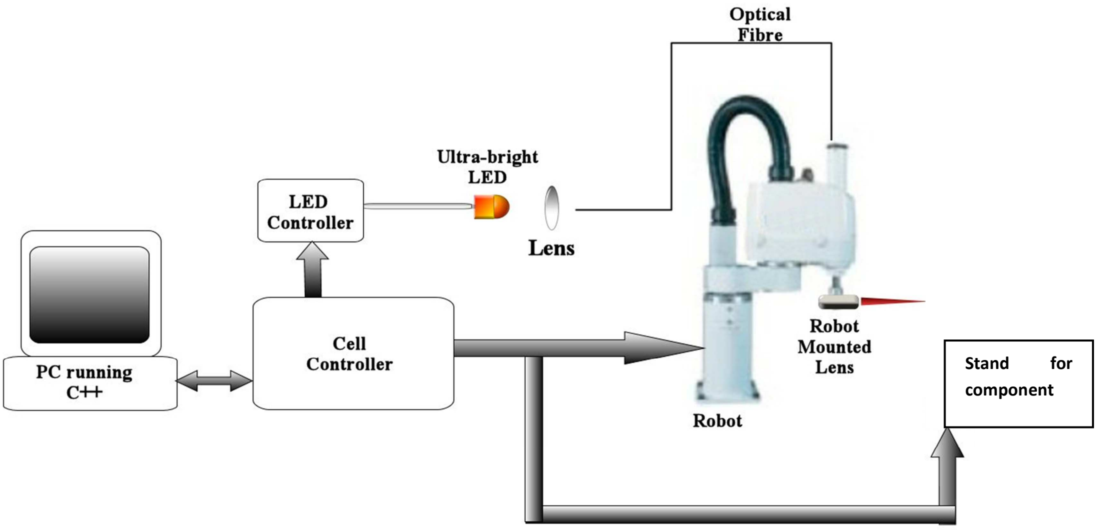


Figure 3-1 Original hardware configuration

However, the feasibility study also showed that further development would be required for both the hardware and the software. For example, the results indicated that using the LEDs as a light source would require exposure times of up to 5 seconds to fully expose a single pixel or point (i.e. turn completely black). Consequently, total painting times were excessive but it also appeared likely that they could be greatly reduced by using a source with a greater energy level (i.e. a laser). The work also suggested that the point extraction algorithm required refinement so that the complete model could be processed, as the system initially only dealt with points that were physically next to each other (i.e. existing on a single loop of the surface). The original hardware configuration can be seen in Figure 3-1 where a 4-axis robot and LED are being used to provide the movement and the light source.

In 2003, funding was obtained from the Scottish Manufacturing Institute (SMI) to develop the optical painting system. Early on in the project, modifications to the hardware were successfully completed allowing a further two axes to be added to the system providing the capability to rotate and tilt the component being *"painted"*. The high power LEDs were replaced with a tri laser source and the system was tested. This setup, shown schematically in Figure 3-2, has been used throughout this research, initially with the original software and subsequently using the improved algorithms developed by this work. The next section details the hardware, software and algorithms associated with this system.

3.1 Hardware Overview

The system shown in Figure 3-2 uses three coloured lasers (red, blue, green) combined into a single beam which enters multi-modal optical fibre to deliver the beam to a focusing lens situated at the end of a robotic arm. This arrangement both anticipates the future use of more sophisticated colour photography and also allows the exposing light to be optimized for the monochrome emulsion which reacts faster to certain wave lengths. The robot is an Epson SCARA E2C (Selective Compliance Assembly Robot) arm with four degrees of freedom (three rotational and one prismatic) though this robot had a limited range of movement (compared to say a 6-axis machine) it offered much better speed and accuracy. Furthermore, the controller could be easily interfaced to external systems and software. Each joint is controlled via the robot controller which also incorporates a pulse generating board for the operation of the lasers and the additional two axes of the tilt table. This setup offers the following advantages:

- Compact and robust optical delivery
- Incremental control of light beam projection
- Incremental control over each independent colour
- Coordination with the movement of the robot and tilt table

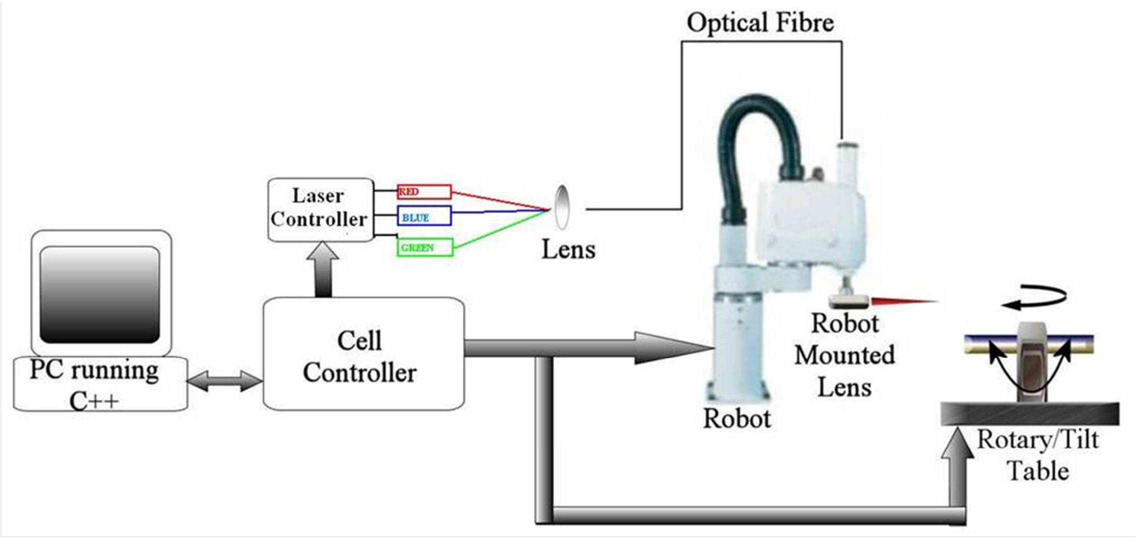


Figure 3-2 New hardware configuration

The robotic cell controller has an Intel Celeron processor (running at 850 MHz) with 128 Mb of RAM and can be used to control the robot directly, or process instructions from a PC. This latter configuration allowed software development to be done in C++ using an API for Epson's SPEL robot programming language (106) (107).

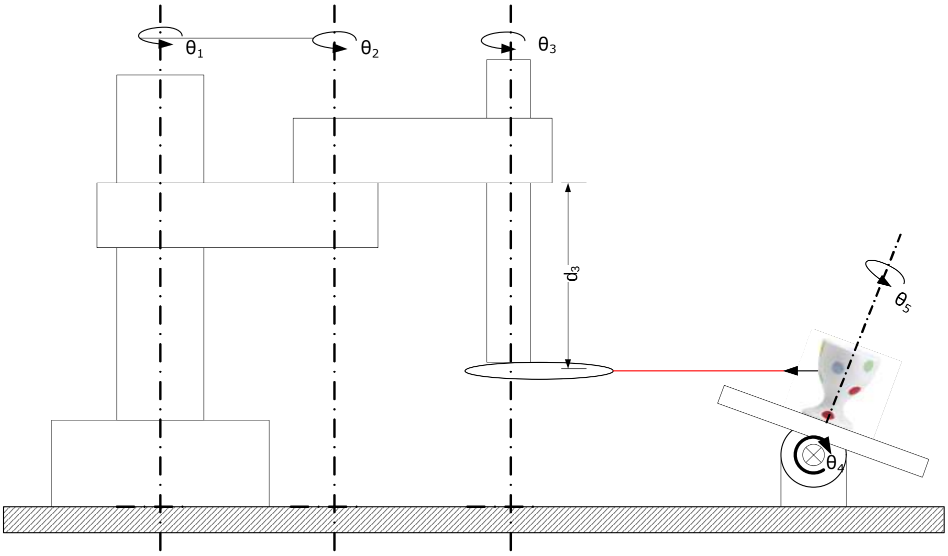


Figure 3-3 Robotic joints

Because the SPEL API allows software development on a PC (Intel Pentium 3 processor with 1 GB of Ram) using Microsoft Visual Studio 2003 (108) it could be interfaced with other graphics and modelling packages. The lens used to target the laser was focused at 6cm, so for optimum performance the robot position had to be offset 6cm along a surface normal. However where a distance of 6cm cannot be achieved due say, to obstruction on the object's surface, it is still possible to expose a point on the surface from a distance other than 6cm, although this will vary the intensity of the beam delivered creating a less consistent appearance. Later, in Chapter 4 image quality assessment will be reviewed and different ways to interpret image quality will be discussed.

3.1.1 Robot Cell Controller Setup

The robotic cell controller coordinates all the movement of the robot and tilt table along with sending the instructions for the laser to be pulsed. It is therefore important to understand the specific configuration of the unit and the order in which it processes instructions. Although the joint angles associated with the start and end points of a path will be fixed, there are several different approaches to controlling the manner in which the robot moves from one configuration to another (51). The different types of movement are distinguished by the way in which changes to the position of individual joints are allowed to influence the end effector.

The principle types of movement are;

1. Point to Point: where there is no attempt to control the movement of the end effector between start and end points. In other words, each joint simply moves to its new position independently of the other joints; there are three distinct forms of Point to Point movement.
    a. Time coordinated: Point to Point, where the speed of the joint's movement is varied to ensure that all the final positions are reached at the same time. In this scenario, the overall time of each move is dictated by the time for the slowest joint (i.e. lowest velocity or furthest distance).
    b. Un-controlled: Point to Point, where each joint moves as fast as it can to its final location, again overall movement time will be determined by the slowest joint.
    c. Sequential: Point to Point, where each joint is moved individually in a sequence. Overall time will be the sum of all the individual movements.
2. Controlled Path: where the movement of individual joints is modified to ensure the behaviour of the end effector meets some predefined criteria. There are two principle forms of controlled path.
    a. Coordinated Path: Controlled path, where the end effector moves between start and end points along a defined route (e.g. arc or straight line).
    b. Velocity: Controlled path where the end effector moves along a pre-defined route at a defined or controlled speed.

In this work, there was no need to control the movement of the end effector between points since:

- The image was created from discrete points rather than continuous arcs or lines so the path between pixels was of no importance.

- There was little danger of collisions because of the offset, non-contact nature of the task.

As a result of these circumstances, Point to Point control was assumed. Given that the robot is exposing hundreds of very small end effector movements (typically of the order less than 0.001 radians of movement to create 150-300dpi images), there was no need for coordination of the individual joint movements. Furthermore, constraints of the SPEL API forced the implementation to adopt a sequential Point to Point approach.

The system was constrained to a sequential Point to Point approach because the two tilt table axes had to be driven by explicit SPEL commands to move the joints. In other words, SPEL had no utility to extend the time coordinated Point to Point movement (it was able to execute for the basic 4-axis SCARA robot) to incorporate the additional axes of the tilt table.

While more sophisticated robot-control languages might provide this facility, it has not been required for the physical experiments conducted in this investigation. The theoretical investigation into the path planning algorithm detailed in Chapters 5 and 6 reports changes in overall joint movement (the sum of all joint travel) and the relative changes of the distance moved by individual joints. So although the dominant axis of individual movement might change with each Point to Point move, a comparison could still be made. This is discussed further in Chapter 6.

The next section looks at the specific details of the point extraction and current path planning software.

3.2 Path Planning Software

The path planning software's function is to derive a sequence of painting configurations which instruct the robot to move to a series of positions and orientations and pulse the tri-colour laser for a calculated amount of time in order to expose pixel by pixel an image on the object's surface. Because thousands of points need to be generated, it is essential that the sequence of instructions can be automated e.g. generated directly from a digital model of the shape (e.g. VRML 2 model). For example, the system aims to produce images of around 300dpi, implying a theoretical maximum of 90,000 dots per square inch.

Consequently, the points must be organised in an efficient manner so that the painting process can be performed as quickly as possible. Although this problem appears to be under constrained (as most of the surface positions are visible from large sections of the workspace and the kinematics of the six degree of freedom cell allows a large number of potential solutions), certain physical restraints exist which limit the possibilities to a more manageable number (i.e. the tilt table works between -60° and +60°). The principle constraints on the system are:

1. Orientation: Although a point may be visible from many different configurations, the maximum amount of energy will be produced when the beam is aligned along the normal axis with the surface located at the focal point, this will also produce the best image quality (i.e. sharpest, smallest pixels).
2. Energy of movement: The robot and tilt table can be configured in numerous ways to deliver a light ray to points. It is therefore important to sequence the points in a manner that efficiently organises the points and minimises the energy and time expended moving from point to point.

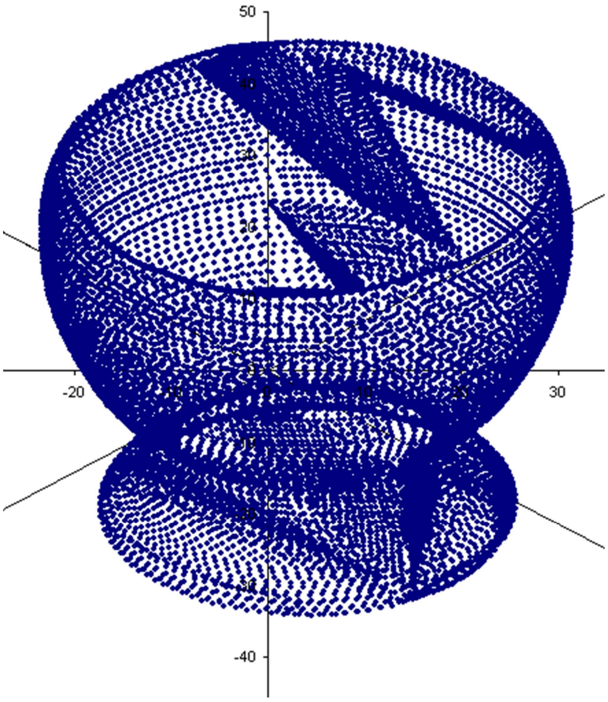


Figure 3-4 Example point set of two thousand points

It is this second requirement that provides the motivation for the work on path planning reported later in this thesis.

3.2.1 Orientation

The orientation of the object relative to the laser is particularly important as this controls the shape of the "*pixel*". Figure 3-5 illustrates the different types of spot that could be obtained when the beam orientation deviates from normal to the surface. As the system is photosensitive, the control over spot geometry is crucial to prevent over exposure of certain areas e.g. from dots overlapping. The system uses a constant offset distance and currently only paints dots that can be exposed along the surface normal vector at each position. This approach allows the system to maintain a constant dot shape and size. In the discussion in Chapter 7 the possibility of extending the system to use variable dot size and adjusting the approach vector to handle occluded surface areas is discussed.

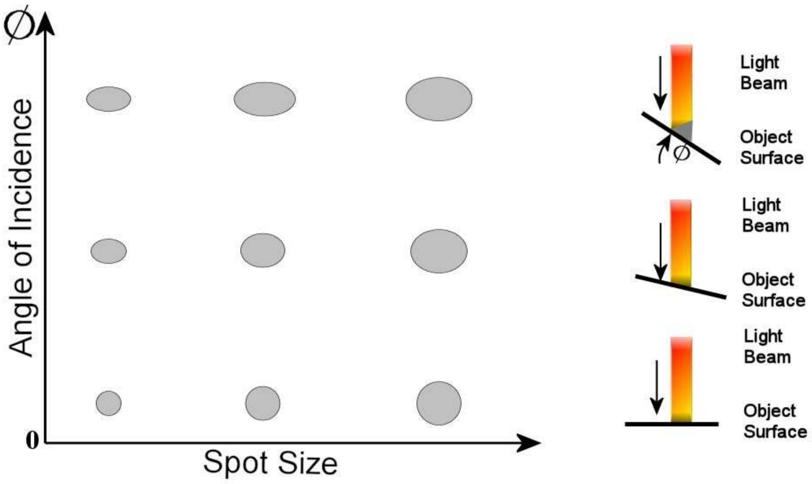


Figure 3-5 Orientation controls the exposure area for consistent points the orientation should remain as close to the normal as possible thus creating a uniformly circular spot profile

3.2.2 Energy of Movement

The study of robot speed (109) (110) has been an area of academic research since robot invention. The complex nature of the problem is easily understood. For example, the motion of a robot can be defined as the time for acceleration, time for motion, time for deceleration and settling time. Each part of the motion must be considered individually, when looking at robotic tasks; quick decelerations can cause longer settling times and long decelerations increases operation time, consequently a balance must be obtained for each stage to ensure the best performance from the robot. Indeed even the type of actuator affects robotic speed; electrical arms have greater speed, and are quicker to configure, hydraulic operated robots can operate with larger loads and give a smoother velocity profile but require significantly more time to calibrate. Engelberger (111) suggests that a "*crossover*" point exists between hydraulic and electric robots. Although this point varies with intended use and commercial factors, for smaller scale tasks in clean environmental conditions (i.e. indoors in a dust free environment), most of the evidence (112) suggests that electric robots would be the better choice for this experimental rig.

The Epson robot uses high torque AC servo motors as its drive system with a maximum speed on the X-Y plane of 2900mm/sec and in the Z plane 1100mm/sec; the rotational axis of the manipulator can perform at a maximum of 2600degs/sec. The low weight of the manipulator allows the use of an electric robot, which also simplifies the calibration process. This low weight also ensures that the settling time is minimal.

3.3 Prototyping Software Architecture

The software available at the start of this work was broken down into two separate modules; the initial module loads the 3D model, extracts point and textural information (i.e. surface image map) and is known as the “*data collection*” module. The second component instructs the robotic cell controller to execute instructions and is known as the data processing module. A schematic of the system can be seen in Figure 3-6. The data collection module is designed to construct 3D models from various file types and then extract a set of points that lie on the surface of the model; the resolution of these points can be defined by the user. The data processing module is designed to read the set of points, offset these points along the normal vector and then move and orientate the robot to these positions before instructing the laser to send pulses of light down the optic fibre.

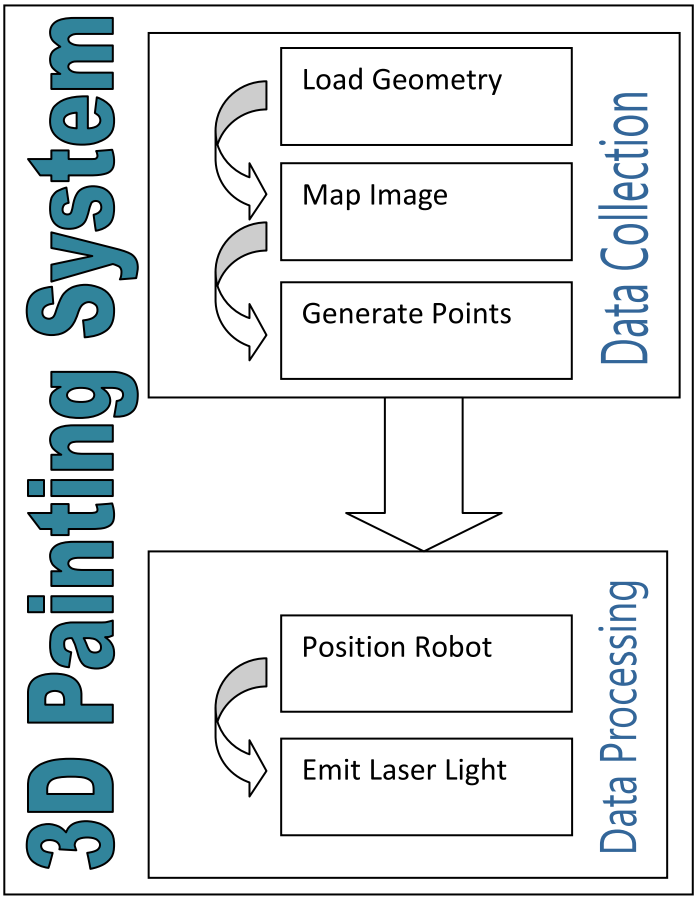


Figure 3-6 System architecture and data flow

Note that this architecture has no path planning step; the robot simply visited the points in the sequence generated by the data collection module.

3.3.1 Data Collection Module

The first module was developed using the Hoops modelling kernel which provides a rich set of graphics and interface tools for use in a CAD environment. The Hoops API allows integration with a variety of different modular programming components. The original code used a combination of Hoops (113) and Trolltech's QT (114) to develop a user interface and a graphical viewer for models. Both these packages were used as they provided cross-platform support for both Windows and Linux systems.

Initial work was done using the Internet orientated 3D modelling language VRML 2; a widely used format for creating 3D images with web browsers. The structure of VRML 2 files follows a simple scheme where initially points are defined then a connection map is applied to construct solids. Following this, a further connection map is used to link an external image file to the surface of the model. This information is extracted from the VRML 2 file using a lexical scanner know as FLEX (Fast Lexical Analyzer Generator) which produces a point set and connection map that Hoops can then process to create solid models with surface texture maps.

The data extraction module was originally coded by Dr M. Gross (115) using the O-Algorithm (described in the next section) and then later re-implemented with the more versatile N-Algorithm (developed by the author). Each of these algorithms produced a 10D array of data holding the Cartesian coordinates, normal vector information and textural description in the form of RGB and greyscale value.

3.3.1.1O-Algorithm

The O-algorithm uses a "raster" approach for extracting points at evenly spaced positions around the circumference of the model. The algorithm simplifies the slicing of the model by moving the model space coordinate frame across the model in incremental steps to define a loop of triangles around the circumference of the model on the X axis (i.e. Y=0). The triangles are extracted to create the loop based on the criteria that all triangles are constituted of points with two positive and one negative or two negative and one positive value for the Y coordinate which must exist on the loop where Y=0. Once the triangles have been extracted, a neighbouring triangle segment is created housing all the triangles connected along the Y=0 axis. The user is next asked to specify a start point on the segment where point extraction should begin, either the complete model can be used or an end point can be stipulated for extracting points on a patch.

The extraction algorithm works by calculating the distance from the start point to the finishing point or the complete circumference then sub-dividing the model into layers based on a resolution that the user specifies. Once the model has been sliced, the triangles constituting each layer are calculated by examining the connections that exist between the start triangle and all subsequent triangles in a similar fashion to creating the Y=0 segment. Where the next triangle has two points above or below the slice height, it forms part of the loop, when the loop's next triangle is the start point, a complete segment circumference has been completed. Once the group of segment circumferences are calculated, representing the complete area being sampled, point extractions occur on each segment at the given resolution. Though the algorithm extracts points quickly there are some inherent problems; where the start point for a given horizontal slice is at or near the boundary of the next triangle in the sequence, a larger gap than expected may appear at the end of the slice depending on the size of the start triangle as the start triangle is classed as already being processed. The algorithm also relies on a next triangle sequencing approach which means that only connected triangles on a layer will be processed.

The O-algorithm provides a quick and efficient method for extracting points from geometric models using a raster points generating approach. It handles simple geometric models completely, though it does not provided coverage for models that contain holes or re-entrant features. In other words the algorithm handles objects that are fully connected on a single loop without for example, handles, Figure 3-7. This was a significant limitation so a new algorithm known as the N-algorithm was developed that could handle geometries that are more complex.

3.3.1.2 N-Algorithm

The N-Algorithm (116) also uses a raster approach for extracting points but extracts information across the entire component and not just a single surface loop.

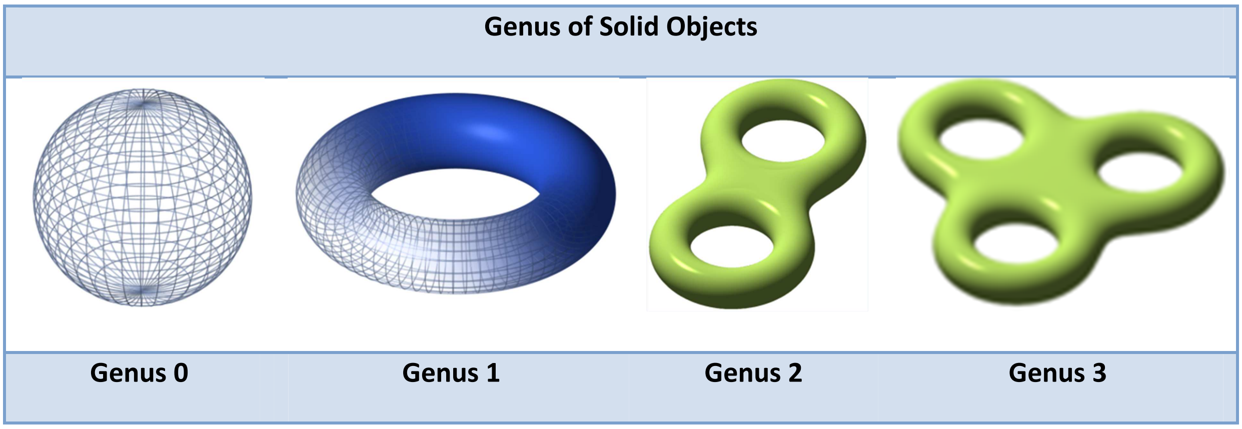


Figure 3-7 Genus of an object

The algorithm operates in a similar fashion to the O-Algorithm in formulating a loop segment along the Y=0 path which is then used to sub-divide the model on a two-axis plane. However, when sub-dividing layers instead of traversing the surface until the start triangle is found, the N-Algorithm extracts all triangles that exist at a given height using the same assumption of two points above/below the slicing height. A connection map is then formulated for the triangles which may yield multiple loops rather than a single-looped segment; this allows point extraction to occur over the complete model at any given height. This system reduces the number of segments required as complete layers can be analysed rather than separate loops, it also prevents geometric anomalies from significantly affecting results (i.e. holes in the surface prevented the O-Algorithm from processing the surface).

The N-Algorithm provides a significantly more stable solution as the algorithm processes all the triangles on the layer, no matter if mesh anomalies occur or not, the algorithm continues without crashing. The O-Algorithms suffered from a lack of stability, when a loop of triangles was detected, points were extracted around the loop but if the mesh was damaged or had an incorrect connectivity map then it would crash i.e. if all the triangles in the loop did not face in the same direction or were not all connected in the connectivity map. The N-algorithm however, identified all the triangles in the loop and where an abnormality occurred in the mesh the algorithm processed all the remaining triangles before moving on to the next layer without crashing. Mesh abnormalities are a common problem especially with scanned data, so consequently a healing algorithm tends to be used prior to any operations however, it does not always remove all the problems within the scanned mesh.

Both the O and the N algorithm use a raster approach to point scanning where a start and a finish position are specified by the user. The algorithms have produced consistent results with uniform linear point coverage and point separation for models' profiles; this has been assessed by visual inspection. Throughout the rest of this thesis, the experiments will be conducted using the N-algorithm.

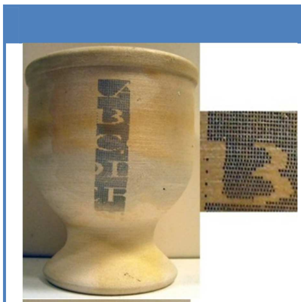
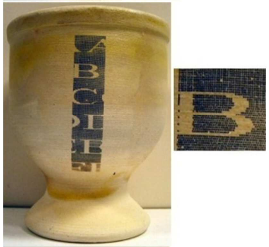
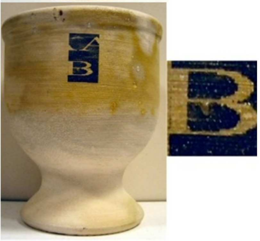
3.3.2 Data Processing Module

The "*data processing*" module is used to communicate with the robot. The communication software uses a simple user interface to open the point file and display processing information, robot instructions use simple SPEL commands (e.g. move, pause and stop) so that users can track the system while it is processing points. The software links Windows C++ programming to the industrial software for operating the cell controller using the COM (Component Object Model) interface. Using this, text strings are sent to the industrial robot using the SPEL programming language and are interpreted as instructions.

The cell controller has complete control over the different hardware modules, and therefore instructions must be directed to the independent hardware components. Each hardware component is named and is initially calibrated by the cell controller before painting can commence. The calibration process for the robot involves moving the robot to certain positions manually to allow it to calibrate its sensors within the joints. The tilt table was designed with optical sensors to allow the table to calibrate by sensing its position.

3.4 System Performance

Table 2 summaries the performance of both the LED and the Laser system for the painting of several test patterns. It is clear from the table that the new system using the tri-colour laser setup is capable of exposing points much quicker than the LED system. However, each of the test patterns only processes a small segment of the eggcup and so the time to process the entire model would still be significant. An important observation is that the time required to expose each point with the laser system is less than a second. Therefore, the system is constrained not by the exposure time, but by the time that the robot takes to position itself correctly to illuminate a point.

	LED System	Laser System
	100dpi 5000 dots 2.5 hours 1 sec exposure time per dot	100dpi 5000 dots 26 minutes 1 milli-sec exposure time per dot
	130dpi 8000 dots 4 hours 1 sec exposure time per dot	130dpi 8000 dots 49 minutes 0.2 milli-sec exposure time per dot
	300dpi 14893 dots 7 hours 0.8 sec exposure time per dot	300dpi 14893 dots 2 hours 0.2 milli-sec exposure time per dot

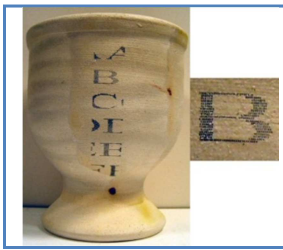
	130dpi 3747 dots 2 hours 1 sec exposure time per dot	130dpi 3747 dots 20 minutes 1 milli-sec exposure time per dot

Table 2 Comparison of the LED and the Laser systems' performances

3.5 System Conclusion

This chapter has described how the hardware and software of the painting system were improved at the start of this work. Although these changes improved the speed and flexibility of the system tests, Table 2 showed that the speed was still too slow to compete with manual processes. Given that the exposure time is only milliseconds, further improvement must come from elsewhere. There are two choices:

- More sophistication in the point generation algorithm
- More sophistication in the sequencing of the points

The next chapter investigates how point generation may improve the system's performance and reviews methods of evaluating image quality. Chapters 5 and 6 look at the sequencing of the points and investigate the potential of path planning to improve the system's performance.

Chapter 4 Surface Point Cloud Generation

The previous chapter identified two approaches to improving the speed and quality of the 3D painting system. This chapter reports investigations into the first of these options, namely the improved generation of the surface point cloud.

Essentially each of these points represents a pixel in the final image so there are two considerations:

1. How the nature of the point pixel distribution affects the image's final quality.
2. The computational techniques available for the generation of points on a surface.

Both these issues are investigated in this chapter.

4.1 Image Quality Assessment

The quality of an image on a surface is difficult to quantify as many variables affect the outcome. Within academia, the construction of a theory of image quality has not been completed for physical models though several algorithms have been proposed for virtual images (117) (118) (i.e. pictures on a computer screen).

The traditional perception of digital image quality relies on the simple premise that higher resolution and higher bits per pixel (i.e. the numbers of colours or shades) produce better images. This assumption has meant that manufacturers have strived to produce printers and scanners capable of significantly higher resolutions than are required for conventional printing or scanning applications. The 3D painting system uses an industrial robot with an accuracy (+/- 0.008mm) and allows the possibility of increasing the resolution of prints in excess of 400dpi (i.e. the traditional method for measuring image resolution in 2D). This high resolution would come at a cost, therefore, a balance must be sought where image quality is maximised to produce the most realistic components with consideration for the time required to manufacture it.

One obvious question (119) (120) (121) is; "*Where is the trade off between resolution and greyscale capabilities?*" Resolution is a measure of the number of dots that exist within a given area, normally measured in inches. Greyscale is a measure of the number of steps that exist between black and white, historically 256 in modern day computers (as this represents 8 bits or 1 byte of data), however other numbers have been used such as 16 steps and 4096 steps. (119) Farrell demonstrates, using practical examples, that a trade off does exist where an image at 300dpi with 8 bits greyscale is regarded as being similar to an image of 600dpi with 4 bits greyscale. Farrell then goes on to propose that dpi, though it is the conventional method for measuring quality, is more a measure of sharpness and that the bits per pixel (i.e. greyscale or the number of intermediate steps) is a better measure of the quality of each pixel.

Given Farrell's work, one could hypothesise that a reduction in the resolution and an increase in the greyscale for the '*Direct Writing*' application considered here might significantly reduce the number of points required to "*paint*" (i.e. expose) an image. Even a small reduction in the number of points could have a significant impact; every 50% reduction in the dpi resolution implies that the number of points falls by a factor of four. For example, at 300dpi we need to paint 90,000 points per square inch; for 600 dpi we need to paint 360,000 points per square inch, which in turn should reduce the time by a factor of four also.

Any point number reduction also decreases the chance of overlapping pixels which becomes an issue at high resolutions. The effect of overlapping pixels is that areas become over exposed and produce dark patches where pixels merge together producing unclear and poor quality images. A similar problem to overlapping pixels can be found in inkjet printing where ink is deposited too closely together in images or where the media being used is of a low quality. When ink is deposited from multiple colour heads and the media is not capable of absorbing the quantity of fluid in the deposition area, the surrounding area sees a bleeding effect, causing ink to spread and colours to combine in an incorrect fashion (122). This is a promising area, and although time constraints have not allowed a thorough investigation here, it would be an interesting area for further work to be carried out.

The perceptual nature of the problem (i.e. several people can look at an image and have different opinions of its quality) has made the construction of a single metric for measuring the quality of images difficult, with several different algorithms being proposed focusing on individual variables. The assessment of image quality through a metric has similarities to challenges found in other applications (123), particularly in signal processing where monitoring, benchmarking and optimization all require to be evaluated against a reference scale. The painting system may not require as comprehensive a solution as signal processing but does still requires some form of evaluation to judge the level of success or failure. It is interesting to note that though work has been carried out on 2D image processing, there appears to be virtually no work carried out on 3D applications. This could be indicative of the complex nature of handling multiple perspective views and more irregular surfaces.

Time has not allowed the systematic investigation of 3D image quality. However, a qualitative assessment has been made by visual inspection i.e. looking at one component, comparing it to a previously exposed component and evaluating which appears to be a more accurate reproduction.

4.2 Image Quality: Dot Patterns

The point sampling system that has been used in the optical painting system is similar to traditional inkjet printing in that it produces linear points at regular intervals. The system operates using a user specified start point which samples points on the circumference of the model at a specified frequency i.e. based on the dpi that the user requests. Once a complete circumference has been sampled, the system incrementally moves down the model and begins sampling the model on the new level. The primary benefit of this form of sampling in a 2D environment, such as printing, is that one axis only needs to be moved at anytime. However in the 3D painting system where points are offset from the surface, two points that are close on the model, may create distant locations off the model if they have a different normal (e.g. Figure 4-1 A points lying on the other side of a sharp edge).

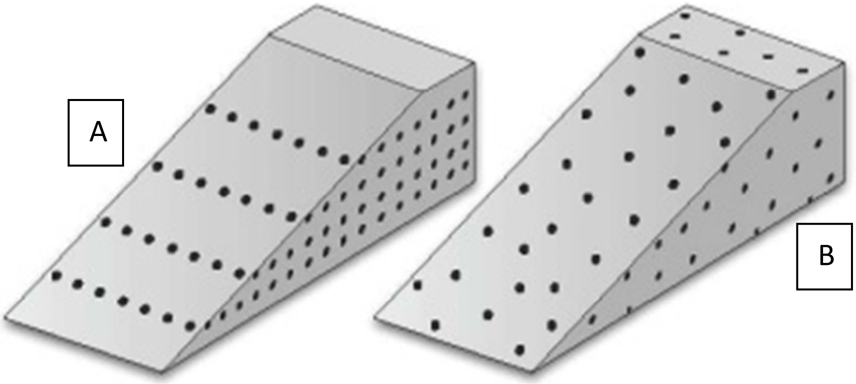


Figure 4-1 Linear and spatial point sampling approaches, linear can be seen to be regular over one face model 'A' whereas spatial provides uniform sampling across model 'B'.

Spatial point sampling algorithms have been used in a variety of different problems (e.g. to avoid aliasing artefacts that may be caused by linear sampling and to improve surface calculation efficiency (124)). The problem is illustrated by Figure 4-1 where the linear sampling approach provides regular sampling over individual faces, whereas the spatial sampling provides uniform sampling over the entire model. It is therefore possible that using paths generated from spatial sampling may produce more uniform coverage, as the points can be controlled based on a point density function rather than a user-specified linear sampling rate. A variety of different spatial sampling approaches exist, though two different approaches will be looked at here, namely; stratified point sampling (125) and spatial curve sampling (126), both these algorithms sample points in a spatial fashion rather than in a linear manner.

Stratified point sampling requires the 3D model to be voxelised and registered within an octree structure. Once this has been completed, each voxel is analysed to calculate which triangles exist within its boundary. The triangles are then sub-divided so that sub-triangles (those that cross the voxel boundary) fit within the voxel. An exponential distance function then assesses the distance from the centre of the triangle to the centre of the voxel and the triangles are prioritised based on this distance. The triangle with the highest priority is then used to generate the sample point.

Figure 4-2 shows an example of a model's surface being sampled by an octree arrangement of voxels.

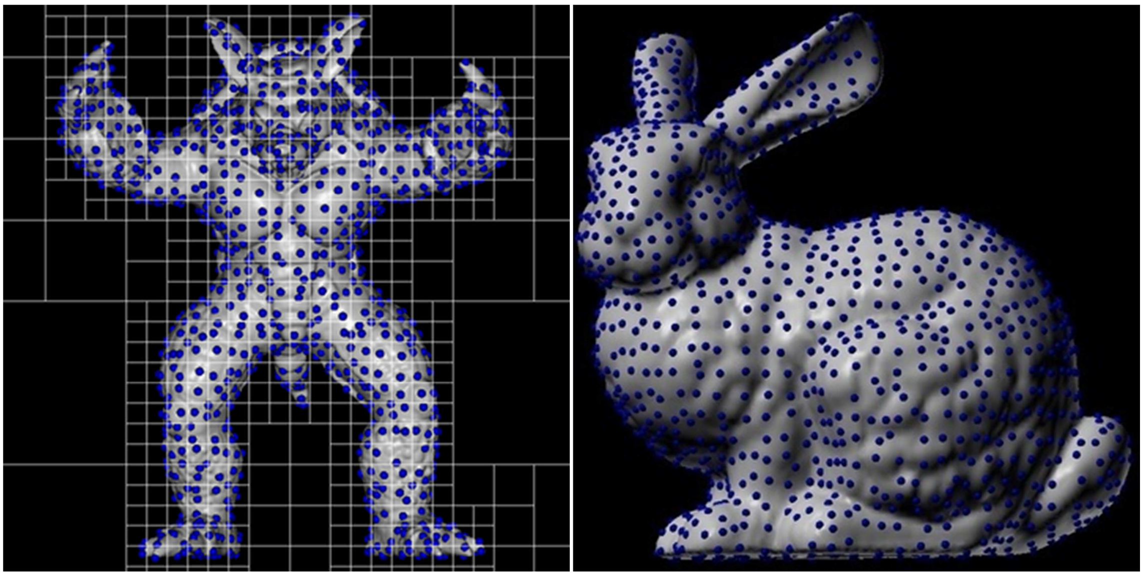


Figure 4-2 Spatial sampling using voxels and spatial sampling using spatial curve (127)

Spatial curve sampling works in a very different manner to voxel subdivision and starts by flattening (i.e. unfolding) the 3D mesh to form a 2D network of triangles. Then a space-filling curve is used to create a single path that covers the entire model. A variety of different space filling curves exist using different methods to achieve complete coverage. The two best known space filling curves (128) (129) are the Hilbert curve (Figure 4-4) and the Peano curve (Figure 4-3) which both operate with a unit square to provide complete coverage. Once the spatial curve has created a path over the 2D mesh, sampling can take place along the path either; using a uniform distance between samples or at control points (i.e. where the path changes direction).

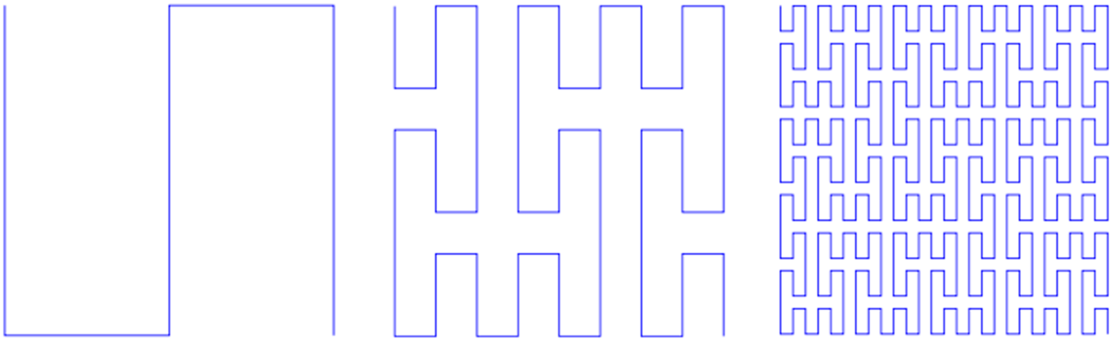


Figure 4-3 Peano curve 1st to 3rd order curve (130)

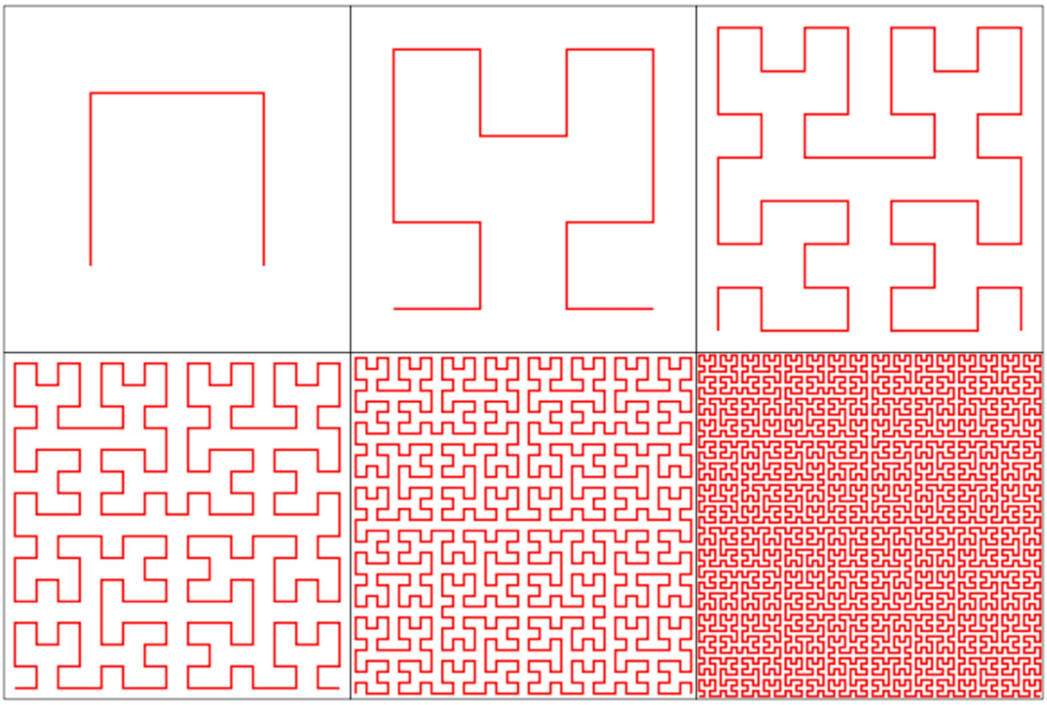


Figure 4-4 Hilbert curve varying degrees from 1st to 6th (130)

Although both the spatial and stratified sampling approaches can provide an even point distribution across the surface of the model, it is also important to note that linear sampling can provide an even distribution on many cylindrical components. Perhaps the most important question is; when is each of the different approaches most applicable? Though this thesis does not directly deal with this question, the following section briefly reports an investigation into the potential for a spatial curve sampling procedure as an alternative to the linear approach.

4.3 Point Extraction Algorithm

To assess the potential improvements that could be obtained by using a more sophisticated point generation algorithm, this section reports the results obtained when a spatial curve was used to generate the surface points rather than the linear point extraction approach (i.e. the N-algorithm).

4.3.1 Q-Algorithm

In association with Jonathan Quinn at Cardiff University's Department of Computer Science (131), a uniform surface coverage of the eggcup model was generated using an adaptive Hilbert curve (132) (124).

The Hilbert curve is a space-filling curve that uses a recursive pattern to provide complete surface coverage. The construction of a Hilbert curve requires the surface mesh to be manually “cut”, (normally in an area of low interest) to allow the 3D mesh to be reduced to a 2D surface. Once a 2D surface has been created, it can then be sub-divided using a quad tree depending on the required point density and after being reduced to the required quad density, a space-filling curve is then used to visit each of the quads. Figure 4-5 a) shows a path across the surface of an eggcup created using a traditional Hilbert curve. One of the problems that can be seen on this model is that certain areas do not appear to have a uniform point distribution, this can be caused by distortions created when mapping from a 3D mesh to a 2D surface. To overcome this distortion, a second approach called an adaptive Hilbert curve was investigated.

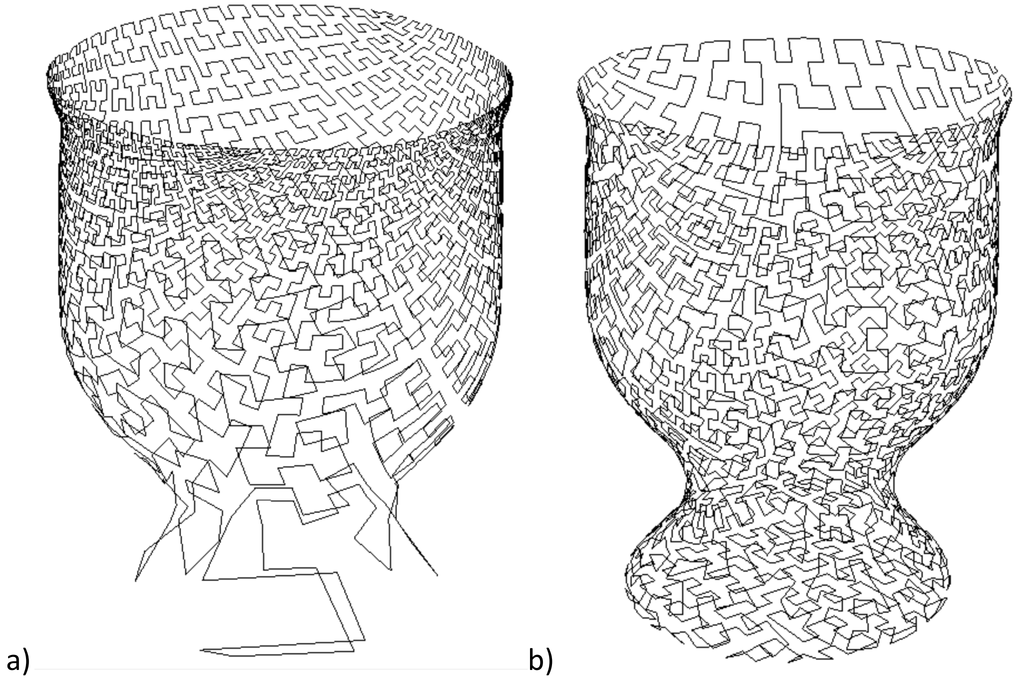


Figure 4-5 a) Hilbert curve b) adaptive Hilbert curve

The adaptive Hilbert curve works in a similar fashion to the traditional Hilbert curve but when sub-dividing the quads, rather than using a constant number of sub-divisions across the entire model, the sub-division limit is varied by an analysis of the 3D geometry and resulting 2D surface. While the traditional flattening process could reduce large areas on the 3D mesh to a small area on the 2D surface, the adaptive Hilbert curve sub-divides these smaller 2D areas further to provide a uniform point density on the 3D model.

Figure 4-6 shows an example of a 2D surface being adaptively sub-divided then shows the Hilbert curve that would be generated for this sub-division.

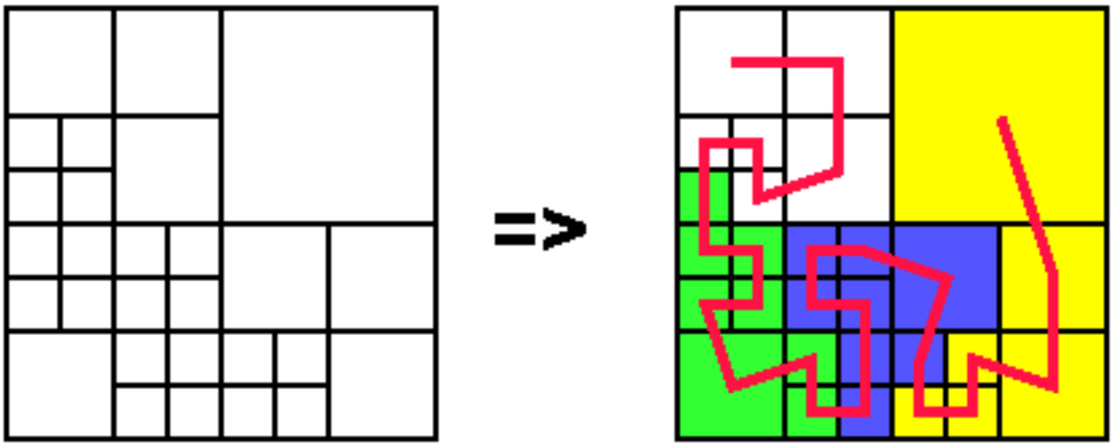


Figure 4-6 Adaptive quad sizes depending on sampling showing the adaptive Hilbert path (133)

The benefit of the adaptive approach rather than the traditional Hilbert curve is clear. Throughout the rest of this chapter, the adaptive Hilbert curve has been used and is referred to as the Q-algorithm.

4.3.2 Results

Using the Q-algorithm various point samples were obtained with a similar number of points as the N-algorithm across the surface of an eggcup component. The lowest sample was obtained at 10dpi which equated to approximately 1600 points. This has been visualised in Figure 4-7 where it is possible to see that the Q-algorithm creates a more complex path to follow with a more varied distribution of points. The N-algorithm can be seen to create linear paths on the z-axis and concentric circles on the x-y plane. Both strategies produce similar profiles for the component, with the N-algorithm creating a more sharply defined top edge with its linear construction.

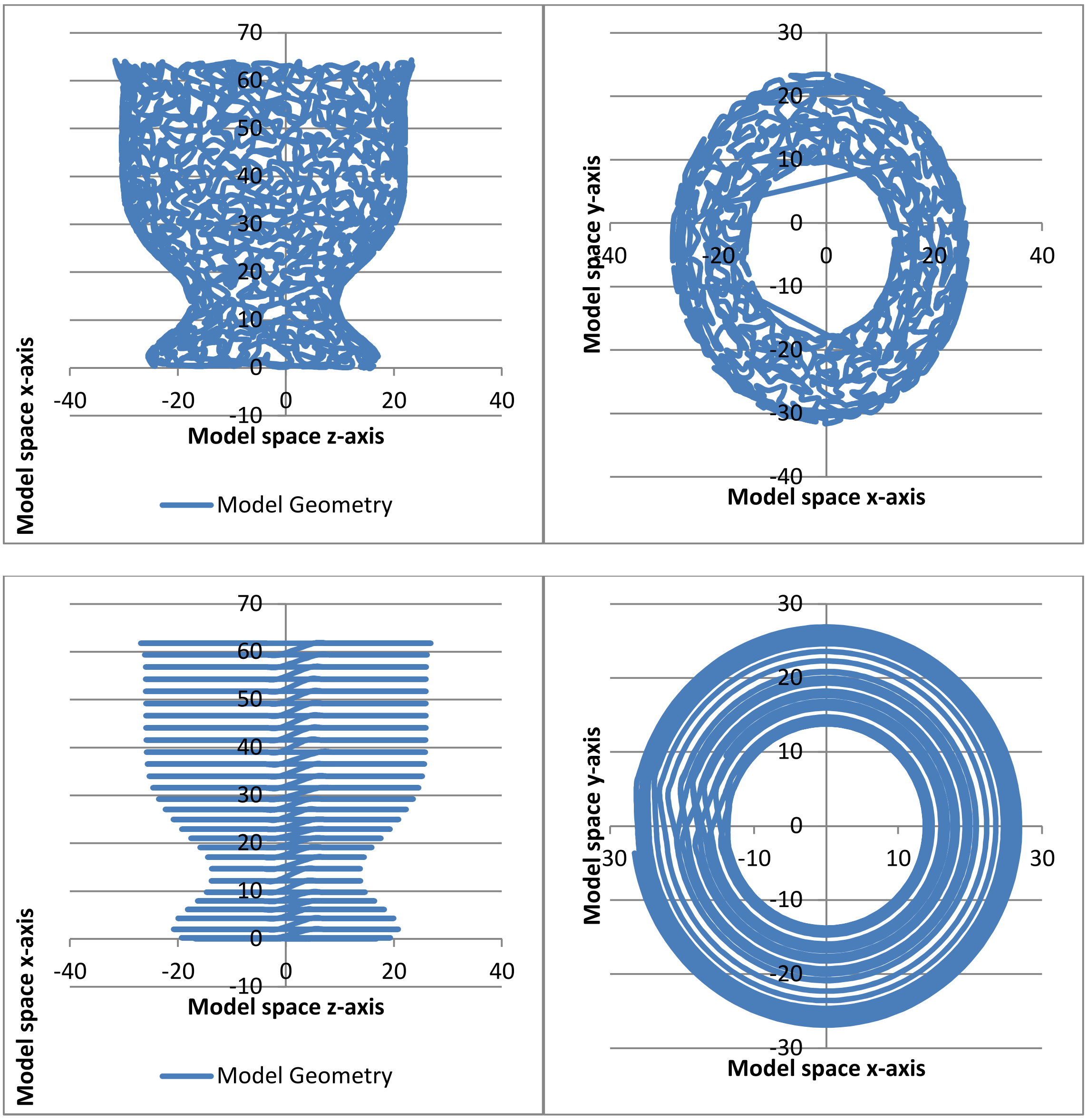


Figure 4-7 Q-algorithm top images and N-algorithm bottom images showing x-z and x-y projections

The surface path allows the image created by the laser to be visualised and the resulting image quality assessed. To quantify the capabilities, a “test card” pattern was mapped on the surface of the eggcup model.

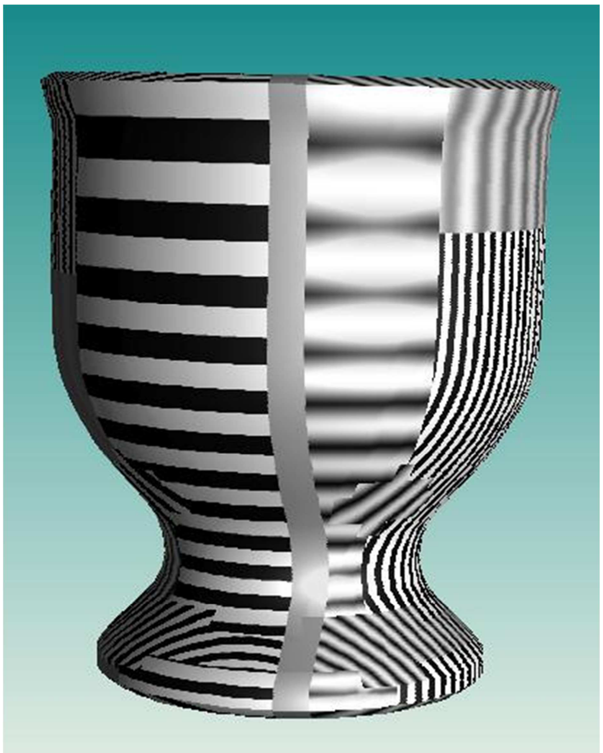


Figure 4-8 Test card mapped on to eggcup model showing the section exposed in the results

Using the optical painting system, the pattern was reproduced physically at resolution of 50 and 100dpi. The results of these experiments can be seen in Figure 4-9 where the four results show the images obtained. The first two images show the results for points generated by the N-algorithm and the final two images show the results for the Q-algorithm. The 50dpi N-algorithm image can be seen to be rather pixelated with clear raster lines evident. It is possible to see that the N-algorithm produces a block effect with small white areas appearing intermittently. The larger white area appearing on the lower band is an artefact of the photosensitive paint bubbling while being exposed. The equivalent image using the Q-algorithm produces a grainy effect still exhibiting a distinguishable image, though with more white areas appearing in the black band. The Q-algorithm appears to produce much softer edges but also looks like it has a lower point density (which is not the case) this would be what is expected as the points are sampled uniformly not with the regularity of the linear approach.

The N-algorithm 100dpi image is a crisp image with a clearly defined edge. However, the lower portion of the component produces a stepping effect where the surface gradient changes significantly (circled in Figure 4-9). This defect is associated with the seam that the N-algorithm creates in the mapping of the image. The 100dpi Q-algorithm is a much denser image than the 50dpi, though unlike the N-algorithm, gaps can still be seen in the pattern though the same stepping artefacts cannot be seen in the image (due to the local control of the sampling process being limited to the vertical bands). It is also possible to see that the pattern appears to be denser on the lower region than on the upper region (i.e. where the surface gradient increases).

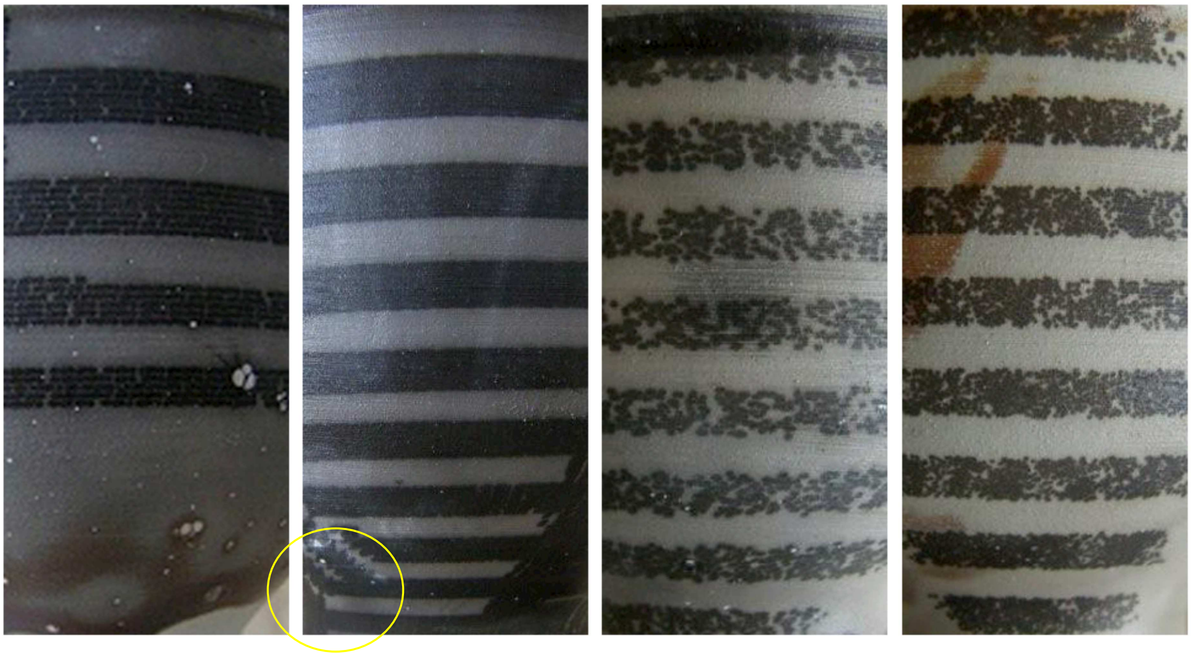


Figure 4-9 Results 1. 50dpi N-algorithm 2. 100dpi N-algorithm 3. 50dpi Q-algorithm 4. 100dpi Q- algorithm

4.3.3 Discussion

The test card used was a conventional 2D lens test image which uses vertical lines with varying widths, whose separation is based on a logarithmic scale. The results that have been obtained for the different point extraction algorithms show an interesting pattern where at low densities the softer Q-algorithm generates images that do not have a "*pixelated*" computer look. This is in stark contrast to the N-algorithm where the block nature of the points gives a binary feel to the results. However, the results obtained at higher resolutions show a lower number of points are required when the N-algorithm is used to produce complete coverage, than the Q-algorithm. This may be due to the uniform pattern producing a better exposure for fringe areas (i.e. areas adjacent to nominal pixel locations are exposed by light leakage causing areas between lines to become black).

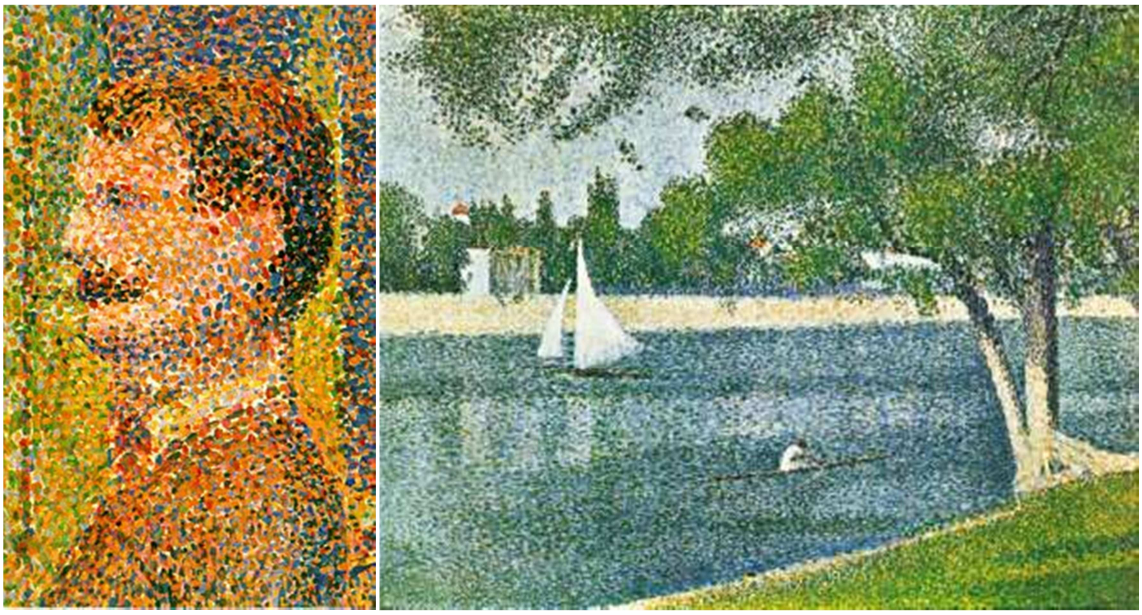


Figure 4-10 Pointillism by George Seurat *La Parade* and *The Seine at Le Grande Jatte*

To reach a definitive conclusion a number of different patterns on various surfaces would have to be tested. However, these limited experiments suggest that point distribution is not significant at high densities because of "blurring" between neighbouring pixels. It remains an open question as to whether the Q-algorithm would produce better results when applied to the reproduction of images in the style of pointillism. The construction of images using spatial points has been used for many years by artists such as George Seurat, Vincent Van Gogh and Chuck Close (Figure 4-10) who use a limited number of colours and various sizes of points to create the impression of continuous tones.

However, although this is an interesting research avenue, it did not address the original objective of this work, namely faster paths. Consequently, the focus of the work returns to more effective path planning by looking at sequencing the points generated by the N-algorithm more efficiently.

4.4 Conclusion

This chapter reports work done in exploring issues of image quality and point sampling algorithms. Although an interesting avenue to explore, time prevented further investigation and many of the topics remain open questions that are discussed in Chapter 7 under further work.

Chapter 5 Points Sequencing

This chapter identifies four possible strategies for improving the efficiency of the path planning process and develops forward and inverse kinematic equations for the system.

5.1 Point Sequencing Algorithm

To create an image, the dot of laser light must expose all the pixels (i.e. points) that make up the picture on the surface of the model. Although the number of points to be visited is fixed by the image quality required (i.e. dpi), the path that is travelled is not. For example, Figure 5-1 shows a simple pattern of four dots at the corners of a unit square and two alternative paths whose lengths vary by 26%.

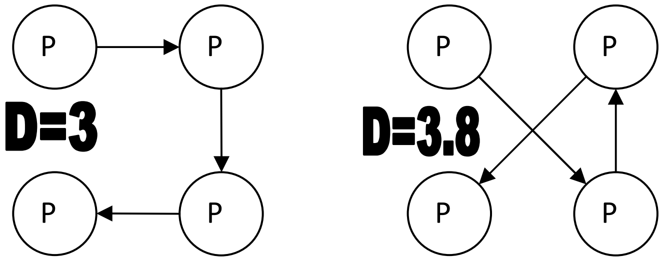


Figure 5-1 4 Points, 2 paths 26% difference in path

So clearly the order in which points are sequenced will have a significant effect on the overall process time. In the case of the unit square, point sequencing algorithms (i.e. path planning algorithms) that simply select the next closest point would avoid the longer route. However, the points being dealt with are not always on a plane, they can lie on any arbitrary curved surface (Figure 5-2).

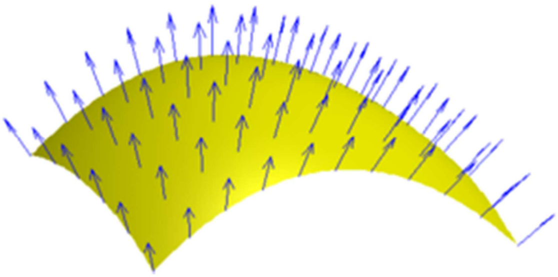


Figure 5-2 Arbitrary curved surface

Although, the calculation of the distance between points on an arbitrary surface is a simple operation, if a straight-line distance in Cartesian space is used, the distance over the surface is a much more involved procedure often involving iterative processes. However, there is another dimension to the problem, in the optical painting process, the dot of light does not simply have to visit all the points but also (if possible) align the beam with the surface normal at that location. Consider the robot shown schematically in Figure 5-3:

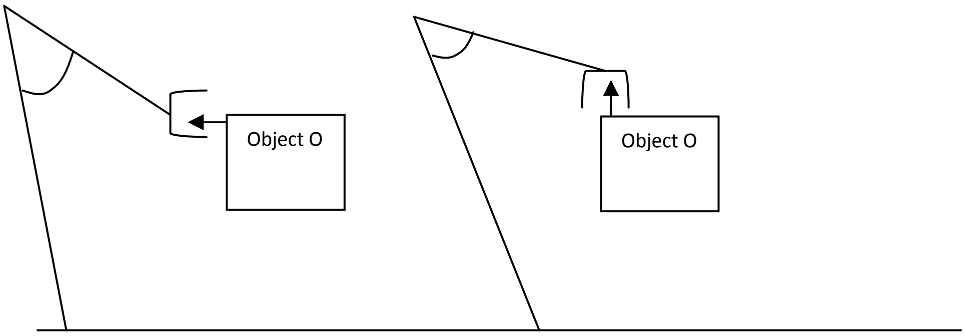


Figure 5-3 Schematic robot showing close Cartesian positions required significant re-configuring

Although two points can be close on a surface, a robot trying to maintain an orientation normal to the surface will be require to alter its configuration. Because of this, there will frequently be shapes where it is not the distance in Cartesian space that is important, but rather the distance in configuration space. This observation suggests that the sequencing of points based on the closest point in configuration space should be investigated. The dimensionality of configuration space is determined by the number of degrees of freedom of the robot. Consequently, the calculation of geometric properties can be a computationally intensive process involving six or more dimensions.

However, concerns about the amount of computation required also motivated the investigation of sequencing of points in order to minimise the movement of a single joint. The hypothesis underlying such single joint optimization was that for rotational or turned components (e.g. cylinders and vases) the majority of axes played only a small role compared to the tilt and rotation of the table. To investigate this theory, paths were generated that moved to the next closest point in terms of the given joint's position.

5.2 Point Sequencing Approaches

The previous discussion identified four distinctly different approaches to the path planning:

1. "Lamina slice" reproduces the sequence inherent in the point generation process using the N-algorithm.
2. "Cartesian closest point" based sequencing where the next closest point in (x,y,z) space is selected (regardless of surface geometry).
3. "Single joint optimization" where the next closest value for an individual axis is selected.
4. "Configuration space closest point" where the next closest point in a multi-dimensional space is selected.

Each of these processes can be described mathematically:

5.2.1 Lamina Slice

Where P is a set of points generated from a 3D model, by in this case, the lamina slicing.

$$P = (P_1, P_2, P_3, P_4, P_5, \dots, P_n)$$

Equation 3

The total distance travelled by any single joint of the N-jointed robot can be expressed as:

$$D_J = \sum_{i=1}^{i=N} \left(P_{(j,i+1)} - P_{(j,i)}\right)$$

Equation 4

Where $P_{(j,i)}$ is the value of a single joint j, at point i, and D_J is the total distance travelled through by axis j.

The total movement of the N-jointed robot can be written as:

$$D_t = \sum_{\mathrm{j}=0}^{\mathrm{j}=\mathrm{N}} D_j$$

Equation 5

Where D_t is the sum of the movements for each of the robotic joints D_j.

5.2.2 Cartesian Closest Point

Where P is the set of points generated from the 3D model by the lamina slice procedure and CP is a sequenced set of points, created from re-sequencing P so the next point in the list is always the nearest one in Cartesian space:

$$CP = (CP_1, CP_2, CP_3, \dots., CP_n)$$

Equation 6

The Cartesian closest point algorithm (Algorithm 1) reads the set of points P and finds the nearest Cartesian point in the list of unvisited points and adds it to the sequenced set CP.

```
Read $P_0$ and Set as $P_{Current}$                // Get start point from unvisited list
Remove $P_0$ From Points set P                     // Remove from list of unvisited points
Add $P_{Current}$ to $CP_0$                        // Add point to visited list
Set $D_{min} = 1000$                               // Set Distance between point large
Repeat
    For n = first to last (of point set P)
        Calculate Distance $D_{current}$ $P_n$ to $P_{Current}$
                                                   // calculate the distance between
                                                   //points in Cartesian space

        If $D_{current} < D_{min}$                 //if distance is shorter than shortest
            Set $D_{min} = D_{current}$            // Set Shortest distance equal current
            Set $P_n$ as $P_{next}$                // Set next point to be visited as current
        Else
        Go to next n                               //else go to next unvisited point
    End for Loop.                                  //Stop loop when all unvisited points checked
    Remove $P_{next}$ from Point set P             //Removed next point from unvisited point set
    Add $P_{next}$ to CP Point                     // Add point to visited list
    Set $P_{Current}$= $P_{next}$                  // Set point as current point
Return to start of repeat as long as P still has points in it.
                                                   // Repeat as long as there are points in
                                                   //unvisited point set
```

Algorithm 1 Pseudo code defining the Cartesian Closet Point Algorithm

The total distance travelled by any single joint of the robot can be expressed as:

$$D_J = \sum_{i=1}^{i=N} \left(CP_{(j,i+1)} - CP_{(j,i)} \right)$$

Equation 7

Where $CP_{(j,i)}$ is the value of a single joint j, at point i, and D_J is the total distance travelled through by axis j.

The total movement of all the robot's joints can be written as:

$$D_t = \sum_{j=0}^{j=n} D_j$$

Equation 8

Where D_t is the sum of the movements for each of the robotic joints D_j. In the case where two points are the same distance apart, the first one encountered in the list generated by the lamina slice procedure will be selected.

5.2.3 Single Joint Sequencing

Where P remains a set of points generated from the 3D model and SJP is a sequenced set of points, based on a single joint sequencing approach.

$$SJP = (SJP_1, SJP_2, SJP_3, \ldots.., SJP_n)$$

Equation 9

The single joint sequencing algorithm (Algorithm 2) reads the set of points P and sequences them to minimise the movement of a given joint J. This is done by orientating to the smallest value that a specific joint must obtain then incrementally increasing to the largest value.

```
Set J = A                                    //Where A is the axis to sequence
Read $P_0$                                   //Read first point in unvisited list
Set $P_0$ as $P_{Current}$                   //Set to current point
Read $P_{Current,J}$                         //Get joint position for joint J

Repeat
    For n = first to last                    //For the unvisited points list
        If $P_{n,j} < P_{Current,j}$         //If joint angle is less than present joint angle
            Set $P_{Current} = P_n$          //Make point the current point
        Else
        N++                                  //If not go to next point
    End for loop                             //End loop once all points are tested in P
    Add $P_{Current}$ to SJP Point set       //Add current point to SJP
    Remove $P_{Current}$from P               //Remove $P_{Current}$ from Point Set P
    Set $P_{Current}$                        //Set $P_{Current}$ as first value in point set P
Return to start of repeat as long as P still has points in it.
                                             //Repeat loop as long as P has points
```

Algorithm 2 Single Joint Sequencing Algorithm

The sequenced points SJP are then used to calculate the travel of each joint J. The total distance travelled by joint J moving through N points can be expressed as:

$$D_J = \sum_{i=1}^{i=N} (SJP_{(j,i+1)} - SJP_{(j,i)})$$

Equation 10

Where $SJP_{(j,i)}$ is the position of joint j, at point i, and D_J is the total distance travelled through by axis j.

The total movement of an N-degree of freedom robot can be written as:

$$D_t = \sum_{j=0}^{j=n} D_j$$

Equation 11

Where D_t is the cumulative movement (e.g. rotations) for each of the robotic joints D_j.

5.2.4 Configuration Space Sequencing

Where P remains a set of points generated from the 3D model and CSP is the point set P re-sequenced by selecting the next closest point in configuration space.

$$CSP = (CSP_1, CSP_2, CSP_3, \ldots., CSP_n)$$

Equation 12

The configuration space algorithm (Algorithm 3) reads the set of points P and sequences the results based on finding the smallest value for the n dimensional closest point algorithm and places the results in the sequenced point set CSP. When two joints are the same distance apart the first location encountered in the list is used.

```
Read $P_0$ and Set as $P_{Current}$                 // Get start point from unvisited list
Remove $P_0$ From Points set $P$                  // Remove from list of unvisited points
Add $P_{Current}$ to $CSP_0$                       // Add point to visited list
Set $D_{min} = 1000$                              // Set Distance between point large
Repeat
        For n = first to last (of point set P)
                Calculate Distance $D_{current}$ $P_n$ to $P_{Current}$
                                                  // calculate the distance between
                                                  //points in Configuration Space

                If $D_{current} < D_{min}$        //if distance is shorter than shortest
                        Set $D_{min} = D_{current}$   // Set Shortest distance equal current
                        Set $P_n$ as $P_{next}$       // Set next point to be visited as current
                Else
                Go to next n                      //else go to next unvisited point
        End for Loop.                             //Stop loop when all unvisited points checked
        Remove $P_{next}$ from Point set P        //Removed next point from unvisited point set
        Add $P_{next}$ to CSP Point               // Add point to visited list
        Set $P_{Current}$= $P_{next}$             // Set point as current point
Return to start of repeat as long as P still has points in it.
                                                  // Repeat as long as there are points in
                                                  //unvisited point set
```

Algorithm 3 Configuration Space Algorithm

The sequenced points CSP are then used to calculate the travel of each joint J. The total distance travelled by joint J can be expressed as:

$$D_J = \sum_{i=1}^{i=n} \left(CSP_{(j,i+1)} - CSP_{(j,i)} \right)$$

Equation 13

Where $CSP_{j(i)}$ is the value of a single joint j, at point i, and D_J is the total distance travelled by axis j.

The total movement of the robot can be written as:

$$D_t = \sum_{i=0}^{i=N} D_j$$

Equation 14

Where D_t is the sum of the total movements of the robotic joints D_j.

Each of the four path planning approaches described here requires the joint angles to be calculated for the given position and orientation. The next section presents the derivation of the forward and inverse kinematic equations used to model the system.

5.3 Coordinate Systems

The first step in the derivation of kinematic equations is the location of the frames of reference. The combination of robot and tilt table creates two kinematic chains that share a common origin. The two primary coordinate frames that need to be specified are the robotic work area origin (which will also be the global coordinate frame) and the model space coordinate frame used to specify the geometry of the model.

The origin for both the robot and the table has been placed at the base of the robotic arm system. The model space coordinates have been placed at the centre of the robotic tilt table. As this is a circular table, the centre is easily found and therefore registration of models on the table can easily be achieved. The points extraction software was designed to allow the user to stipulate an origin position so this location can be specified.

When constructing robot kinematics, a large part of the work is concerned with the assignment of coordinate frames so that when a link moves, the location of points on that link remain constant within its own coordinate frame (50).

The most common approach to locating coordinate frames is known as the *"Denavit-Hartenberg Convention"* (also known as DH) (134) which uses four parameters to stipulate the relationship between each joint. The method uses individual homogeneous transformations to represent each parameter of the transformation from one coordinates frame (Table 3) to another. To describe these transformations a series of matrices are defined. Using this system, six parameters are required; three describe the linear translations and three describe the Euler angles (135). This thesis uses the DH method to construct the forward kinematic transform for the robotic system.

Transformation Type	Matrix Representation
Lateral Movement along the X,Y or Z Axis Where Px, Py and Pz represent X,Y,Z	$\begin{bmatrix} 1 & 0 & 0 & Px \\ 0 & 1 & 0 & Py \\ 0 & 0 & 1 & Pz \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Rotational Movement About X Axis by $\boldsymbol{\theta}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Rotational Movement About Y Axis by $\boldsymbol{\theta}$	$\begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Rotational Movement About Z Axis by $\boldsymbol{\theta}$	$\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Table 3 Homogeneous Transformations

5.4 Denavit-Hartenberg Convention (DH)

To analyse the motion of a kinematic model, each link must be examined and assigned four parameters which can be used to define a generalized transformation matrix. The DH convention stipulates three rules (51) for the construction of the intermediate coordinate frames between the global coordinate system (Robotic Workspace) and the end effectors.

I. The z-vector, Z_i, of a link frame F_i is always on a joint axis.
II. The x-vector, X_i, of a link frame F_i lies along the common perpendicular to axis Z_{i-1} and Z_i and is orientated from Z_{i-1} to Z_i.
III. The origin of link frame F_i is located at the intersection of the common perpendicular to axis Z_{i-1} and Z_i and the joint axis Z_i.

When these rules are implemented, it is possible to place intermediate coordinates systems throughout the system (Figure 5-4) and so define the four parameters required to define a transformation between coordinate systems. The four parameters that are required to construct a DH model are d_i, a_i, θ_i and α_i where d_i and a_i are geometric length and θ_i and α_i are angles.

I. d_i, called the link offset, is the algebraic distance along axis z_{i-1} to the point where the common perpendicular to axis z_i is located.

II. a_i, called the link length, is the length of the common perpendicular to axis z_{i-1} and z_i; that is, the parameter a_i, is equal to the shortest distance between consecutive joint axis z_{i-1} and z_i.

III. θ_i, called the link angle, is the angle around z_{i-1} that the common perpendicular makes with vector x_{i-1}.

IV. $\alpha_{i,}$ called the link twist, is the angle around xi that vector z_i makes with the vector z_{i-1}.

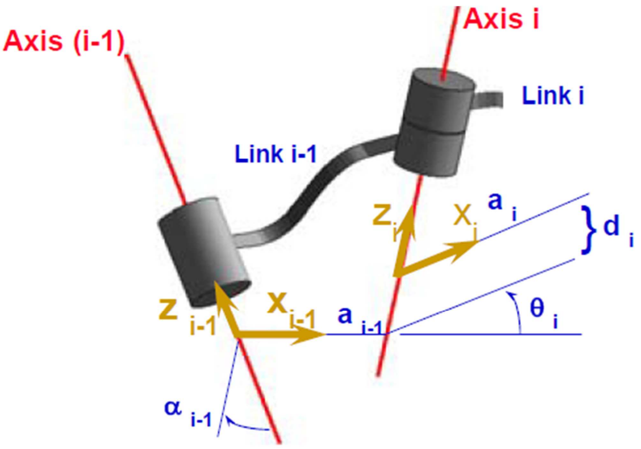


Figure 5-4 DH Convention (136)

Using the DH method imposes rules that greatly reduce the chance of errors in the analysis, by formalizing the procedure.

5.4.1 Generalised Transformation Matrix for DH Approach

As mentioned earlier, the DH approach uses a generalised transformation matrix which combined four of the standard transformation matrices. Though this may not seem like a timesaving option, it limits the options for selecting coordinate frames and therefore reduces the problem from being a six variable problem to a four variable problem. The DH transformation matrix limits the problem to rotation and linear movements along the z and x axes and because all axes are perpendicular, the system can accommodate any movements along the y-axis by a combination of rotations and linear movements.

$$T_{DH} = ROT_{Z,\theta_i}, TRANS_{z,d_i}, TRANS_{x,a_i}, ROT_{x,\alpha_i}$$

Equation 15 Constituent homogeneous transforms of the DH approach

$$T_{DH} = \begin{bmatrix} Cos\theta_i & -Sin\theta_i & 0 & 0 \\ Sin\theta_i & Cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\cdot \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & Cos\alpha_i & -Sin\alpha_i & 0 \\ 0 & Sin\alpha_i & Cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equation 16 Multiplication of transforms

$$T_{DH} = \begin{bmatrix} Cos\theta_i & -Sin\theta_i Cos\alpha_i & Sin\theta_i Sin\alpha_i & a_i Cos\theta_i \\ Sin\theta_i & Cos\theta_i Cos\alpha_i & -Cos\theta_i Sin\alpha_i & a_i Sin\theta_i \\ 0 & Sin\alpha_i & Cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equation 17 Generalised DH matrix

5.4.2 Direct Writing System DH Analysis

The direct writing system (1) (115) constitutes two separate kinematic chains whose end points must be aligned (i.e. equal position an orientation).

The first system to be modelled is the conventional SCARA robot (50) (51). This is comprised of four separate links, which entails four local coordinate systems and a global system. Figure 5-5 illustrates the location of each frame as follows:

1. The global system (i.e. the origin) is shown with a red axis.
2. Each linear movement along the z-axis (d_i) is purple.
3. Linear movements along the x-axis (a_i) are highlighted with green.
4. The orientation of the coordinate frames can be seen in the convention single arrowheads representing the x-axis, dual heads represent the y-axis and triple heads represent the z-axis.

i Link Number	d_i Link Offset	a_i Link Length	θ_i Link Angle	α_i Link Twist	Joint Type
1	H1	L1	θ_1	0	Rotational
2	0	L2	θ_2	0	Rotational
3	H2	0	θ_3	0	Rotational
4	dH	0	0	0	Prismatic

Table 4 DH Table for SCARA Robotic Arm

Where the values for H1, H2, L1 and L2 are constant and dictated by the geometry of the robot; the values dh, $\theta_{1,}$ θ_2 and θ_3 are variables depending on the joint configuration. The value dh represents the only prismatic joint and reciprocates along the z-axis of the reference coordinate system.

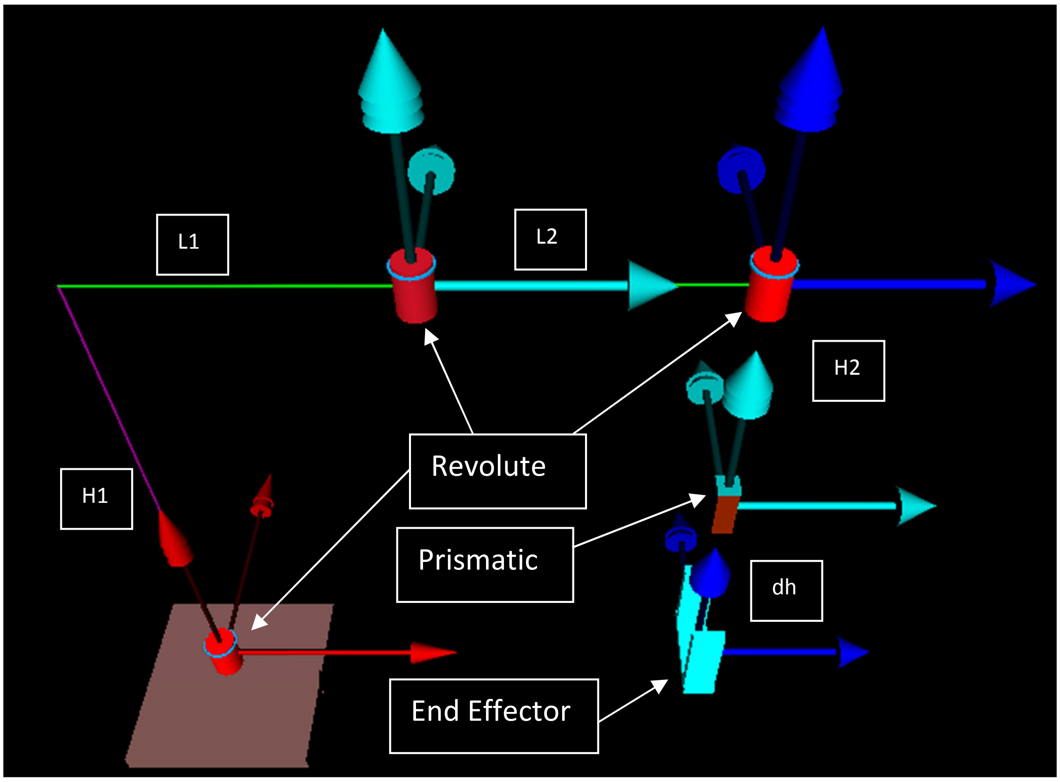


Figure 5-5 Assignment of DH frames to the Robotic Arm (Drawn using Robot Modelling & Kinematics (51))

$$T_{Complete} = T_{Link\ 1}.\ T_{Link\ 2}.\ T_{Link\ 3}.\ T_{Link\ 4}$$

Equation 18 DH components for constructing the robotic arm

$$T_{DH} = \begin{bmatrix} Cos\theta_i & -Sin\theta_i Cos\alpha_i & Sin\theta_i Sin\alpha_i & a_i Cos\theta_i \\ Sin\theta_i & Cos\theta_i Cos\alpha_i & -Cos\theta_i Sin\alpha_i & a_i Sin\theta_i \\ 0 & Sin\alpha_i & Cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equation 19 Standard DH matrix

$$T_{Link\ 1} = \begin{bmatrix} Cos\theta_1 & -Sin\theta_1 .1 & Sin\theta_1 .0 & L1.Cos\theta_1 \\ Sin\theta_1 & Cos\theta_1 .1 & -Cos\theta_1 .0 & L1.Sin\theta_1 \\ 0 & 0 & 1 & H1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equation 20 DH matrix for link 1 of the robotic arm

$$T_{Link\ 2} = \begin{bmatrix} Cos\theta_2 & -Sin\theta_2 .1 & Sin\theta_2 .0 & L2.Cos\theta_2 \\ Sin\theta_2 & Cos\theta_2 .1 & -Cos\theta_2 .0 & L2.Sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equation 21 DH matrix for link 2 of the robotic arm

$$T_{Link\ 3} = \begin{bmatrix} Cos\theta_3 & -Sin\theta_3 .1 & Sin\theta_3 .0 & 0.Cos\theta_3 \\ Sin\theta_3 & Cos\theta_3 .1 & -Cos\theta_3 .0 & 0.Sin\theta_3 \\ 0 & 0 & 1 & H2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equation 22 DH matrix for link 3 of the robotic arm

$$T_{Link\ 4} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & dH \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equation 23 DH matrix for link 4 of the robotic arm

Equation 18 represents the cumulative effect of all the links shown in Figure 5-5 and Equation 19 shows the general form of the DH matrix. The values for the robot (Table 4) have then been substituted into this equation to create each of the link matrices (Equation 20 through to Equation 23). Each of the link matrices have then been multiplied together to create the transformation matrix from the base of the robot through to the end effector Equation 24, this equation has then been simplified into Equation 25.

$$T_{Complete} = \begin{bmatrix} Cos\theta_1 & -Sin\theta_1.1 & Sin\theta_1.0 & L1.Cos\theta_1 \\ Sin\theta_1 & Cos\theta_1.1 & -Cos\theta_1.0 & L1.Sin\theta_1 \\ 0 & 0 & 1 & H1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$$\begin{bmatrix} Cos\theta_2 & -Sin\theta_2.1 & Sin\theta_2.0 & L2.Cos\theta_2 \\ Sin\theta_2 & Cos\theta_2.1 & -Cos\theta_2.0 & L2.Sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$$\begin{bmatrix} Cos\theta_3 & -Sin\theta_3.1 & Sin\theta_3.0 & 0.Cos\theta_3 \\ Sin\theta_3 & Cos\theta_3.1 & -Cos\theta_3.0 & 0.Sin\theta_3 \\ 0 & 0 & 1 & H2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & dH \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equation 24 Concatenation of the constituent links for the robotic arm

$$T_{Complete} = \begin{bmatrix} Cos(\theta_1+\theta_2+\theta_3) & -Sin(\theta_1+\theta_2+\theta_3) & 0 & L2.Cos(\theta_1+\theta_2)+L1.Cos(\theta_1) \\ Sin(\theta_1+\theta_2+\theta_3) & Cos(\theta_1+\theta_2+\theta_3) & 0 & L2.Sin(\theta_1+\theta_2)+L1.Sin(\theta_1) \\ 0 & 0 & 1 & dH+H2+H1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equation 25 Simplified matrix for the robotic arm

Using the same conventions for labelling it is also possible to model the tilt table. The model for the table can be constructed using prismatic joints with a fixed length and then accommodate the two rotational links; an alternative would be to use a conventional homogenous transform for the linear movements. TL1, TH1, TL2 and TH2 are all fixed constants.

i Link Number	d_i Link Offset	a_i Link Length	θ_i Link Angle	α_i Link Twist	Joint Type
1	0	TL1	0	0	Prismatic
2	TH1	0	π/2	π/2	Prismatic
3	TL2	0	θ_4	3 π /2	Rotational
4	TH2	0	(3 π /2)+θ_5	0	Rotational

Table 5 DH Table for Tilt Table

$$T_{TiltTable} = T_{TTL\,1}.\,T_{TTL\,2}.\,T_{TTL\,3}.\,T_{TTL\,4}$$

Equation 26 DH components for constructing the tilt table

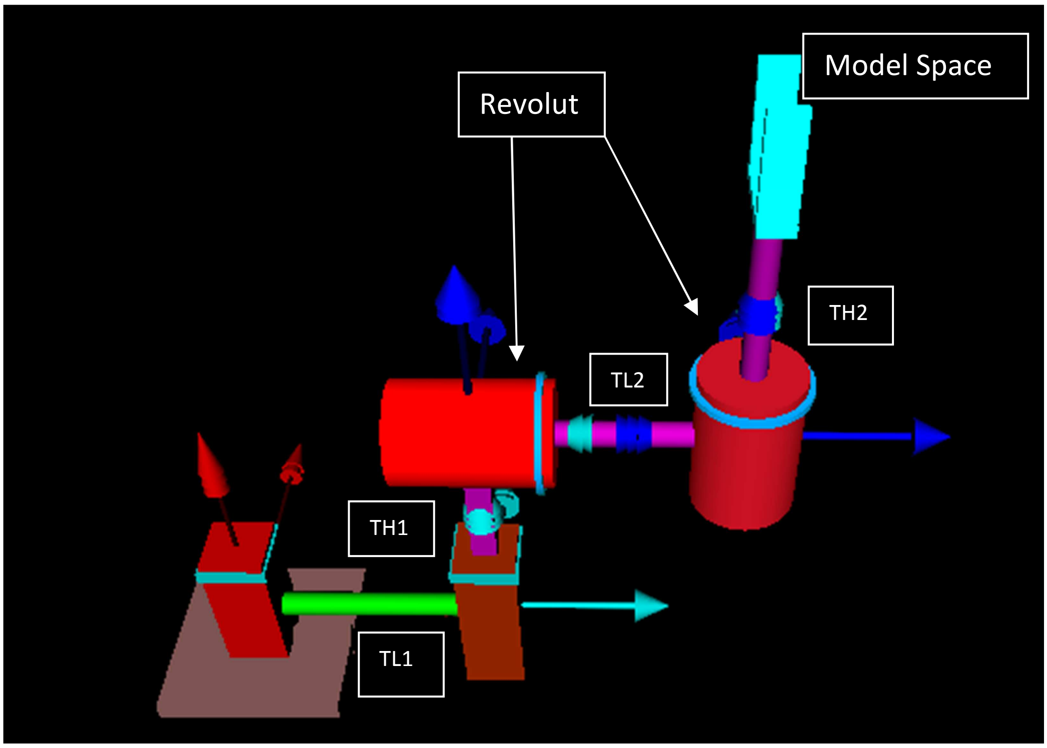


Figure 5-6 Assignment of DH Frames to the Tilt Table (Drawn using Robot Modelling & Kinematics (51))

$$T_{TTL\,1} = \begin{bmatrix} 1 & 0 & 0 & \mathrm{TL1.1} \\ 0 & 1 & 0 & \mathrm{TL1.0} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equation 27 DH matrix for link 1 of the tilt table

$$T_{TTL\,2} = \begin{bmatrix} 0 & -Sin\left(\frac{\pi}{2}\right).0 & Sin\left(\frac{\pi}{2}\right)Sin\left(\frac{\pi}{2}\right) & 0 \\ Sin\left(\frac{\pi}{2}\right) & 0 & -0.Sin\left(\frac{\pi}{2}\right) & 0.Sin\left(\frac{\pi}{2}\right) \\ 0 & Sin\left(\frac{\pi}{2}\right) & 0 & TH1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equation 28 DH matrix for link 2 of the tilt table

$$T_{TTL\,3} = \begin{bmatrix} Cos\theta_4 & -Sin\theta_4.0 & Sin\theta_4.(-1) & 0.Cos\theta_4 \\ Sin\theta_4 & Cos\theta_4.0 & -Cos\theta_4.(-1) & 0.Sin\theta_4 \\ 0 & -1 & 0 & TL2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equation 29 DH matrix for link 3 of the tilt table

$$T_{TTL\,4} = \begin{bmatrix} Cos\left(\left(\frac{3\,\pi}{2}\right)+\theta_5\right) & -Sin\left(\left(\frac{3\,\pi}{2}\right)+\theta_5\right).1 & Sin\left(\left(\frac{3\,\pi}{2}\right)+\theta_5\right).0 & 0.Cos\left(\left(\frac{3\,\pi}{2}\right)+\theta_5\right) \\ Sin\left(\left(\frac{3\,\pi}{2}\right)+\theta_5\right) & Cos\left(\left(\frac{3\,\pi}{2}\right)+\theta_5\right).1 & -Cos\left(\left(\frac{3\,\pi}{2}\right)+\theta_5\right).0 & 0.Sin\left(\left(\frac{3\,\pi}{2}\right)+\theta_5\right) \\ 0 & 0 & 1 & TH2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equation 30 DH matrix for link 4 of the tilt table

Equation 26 shows the connection of each of the links from Figure 5-6. The standard DH matrix Equation 19 has then been substituted for each of the links to create Equation 27 through to Equation 30. The complete transformation has then been calculated from the base of the robot through to the centre of the tilt table to create Equation 31; this has then been simplified to a single matrix Equation 32.

$$T_{TiltTable} = \begin{bmatrix} 1 & 0 & 0 & \mathrm{TL1.1} \\ 0 & 1 & 0 & \mathrm{TL1.0} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 0 & -Sin\left(\frac{\pi}{2}\right).0 & Sin\left(\frac{\pi}{2}\right)Sin\left(\frac{\pi}{2}\right) & 0 \\ Sin\left(\frac{\pi}{2}\right) & 0 & -0.Sin\left(\frac{\pi}{2}\right) & 0.Sin\left(\frac{\pi}{2}\right) \\ 0 & Sin\left(\frac{\pi}{2}\right) & 0 & TH1 \\ 0 & 0 & 0 & 1 \end{bmatrix} .$$

$$\begin{bmatrix} Cos\theta_4 & -Sin\theta_4.0 & Sin\theta_4.(-1) & 0.Cos\theta_4 \\ Sin\theta_4 & Cos\theta_4.0 & -Cos\theta_4.(-1) & 0.Sin\theta_4 \\ 0 & -1 & 0 & TL2 \\ 0 & 0 & 0 & 1 \end{bmatrix} .$$

$$\begin{bmatrix} Cos\left(\left(\frac{3\pi}{2}\right)+\theta_5\right) & -Sin\left(\left(\frac{3\pi}{2}\right)+\theta_5\right).1 & Sin\left(\left(\frac{3\pi}{2}\right)+\theta_5\right).0 & 0.Cos\left(\left(\frac{3\pi}{2}\right)+\theta_5\right) \\ Sin\left(\left(\frac{3\pi}{2}\right)+\theta_5\right) & Cos\left(\left(\frac{3\pi}{2}\right)+\theta_5\right).1 & -Cos\left(\left(\frac{3\pi}{2}\right)+\theta_5\right).0 & 0.Sin\left(\left(\frac{3\pi}{2}\right)+\theta_5\right) \\ 0 & 0 & 1 & TH2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equation 31 Concatenation of the constituent links for the tilt table

$$T_{TiltTable} = \begin{bmatrix} Cos(\theta_5) & -Sin(\theta_5) & 0 & TL2+TL1 \\ \dfrac{1}{2\big(Sin(\theta_4+\theta_5)+Sin(-\theta_4+\theta_5)\big)} & \dfrac{1}{2\big(Cos(-\theta_4+\theta_5)+Cos(\theta_4+\theta_5)\big)} & -Sin(\theta_4) & -Sin(\theta_4).TH1 \\ \dfrac{1}{2\big(Cos(-\theta_4+\theta_5)-Cos(\theta_4+\theta_5)\big)} & \dfrac{1}{2\big(Sin(\theta_4+\theta_5)-Sin(-\theta_4+\theta_5)\big)} & Cos(\theta_4) & Cos(\theta_4).TH2+TH1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equation 32 Matrix for the tilt table

The transformation matrix for both the robot and the tilt table components of the system (Equation 32 & Equation 25) are constructed from the same point. It is therefore possible to develop a set of equations to calculate the angles at which each of the joints must be positioned to expose a given point in model space. This is also known as calculating the inverse kinematic model.

5.5 Inverse Kinematics

The forward kinematic model of any system can simply be seen as a collection of transformation matrices combined to form a single transform function (known as a T_{matrix}) from any given base coordinate system to another coordinate system. T-matrixes have standard forms (Equation 33) where nine elements define rotation (i.e. three Euler Vectors) three elements define linear movements (i.e. x, y and z) and the remaining elements are components of an identity matrix. To calculate the inverse of a transformation matrix means to calculate the position of the final coordinate frames with respect to the joint angles when information is known about the Cartesian position of the end effector. This task is often much more complex than calculating the forward kinematics as several possible solutions may be available, requiring logical analysis to choose the most appropriate. The problem is further complicated in this system as two kinematic chains (tilt table and robot) are being used to describe the robotic position, therefore the calculation of the joint angles must be done in the correct order.

$$T = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Where\ n = normal\ vector, s = slide\ vector, a = approach\ vector\ and$$

$$p = \ position\ vector$$

Equation 33 T-Matrix Standard Form vector (51)

The direct writing system requires the solution to be calculated for two separate chains, this is done by initially calculating the rotational component of the tilt table and then the tilt. Once these values have been found it is possible to calculate the position of the point within the global coordinate frame enabling the robot to position itself to expose this point.

5.5.1 Inverse Kinematic Solution for Rotary Tilt Table

The point extraction algorithm as previously described generates an array of data containing a normal vector as well as a model space position for points on the surface of the model. Using this normal vector, it is possible to calculate the values that will be required for the tilt and the rotation of the table.

$$T_{TiltTable} = \begin{bmatrix} Cos(\theta_5) & -Sin(\theta_5) & 0 & TL2 + TL1 \\ \frac{1}{2\left(Sin(\theta_4+\theta_5)+Sin(-\theta_4+\theta_5)\right)} & \frac{1}{2\left(Cos(-\theta_4+\theta_5)+Cos(\theta_4+\theta_5)\right)} & -Sin(\theta_4) & -Sin(\theta_4).TH1 \\ \frac{1}{2\left(Cos(-\theta_4+\theta_5)-Cos(\theta_4+\theta_5)\right)} & \frac{1}{2\left(Sin(\theta_4+\theta_5)-Sin(-\theta_4+\theta_5)\right)} & Cos(\theta_4) & Cos(\theta_4).TH2 + TH1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equation 34 Kinematic chain for the tilt table is equal to the T-matrix

$$\therefore n_x = Cos(\theta_5)$$

Equation 35 Matrix comparison to find tilt table rotational value

$$\theta_5 = \cos^{-1} n_x$$

Equation 36 Direct calculation of the tilt table rotational value

By inspection, it can be seen that the angle of table rotation is the angle that exists between the normal vector and the x-axis; this can be calculated directly from the tilt table transformation matrix. Equation 35 shows the value and Equation 36 shows the calculation. However, when larger components are aligned in this fashion the position of the end effector of the robot can be outside the reach of the robot therefore, components are re-aligned so that their normal extends along the y-axis. This is often referred to as positioning the end effector in the sweet spot of the robot. The sweet spot is normally an area that is closer than the maximum reach of the robot but often the robot can move smoother through this zone as the robot is more balanced in this area.

$$\theta_5 = \cos^{-1} n_x + \pi/2$$

Equation 37 Rotation to robotic sweet spot

$$\therefore n_y = \frac{1}{2\left(Sin(\theta_4+\theta_5)+Sin(-\theta_4+\theta_5)\right)}$$

Equation 38 Calculation of tilt value by matrix comparison

$$= \frac{1}{2(\sin\theta_4.\cos\theta_5 + \sin\theta_5.\cos\theta_4 + \sin\theta_5.\cos\theta_4 - \sin\theta_4.\cos\theta_5)}$$

Equation 39 Expanding the equation for simplification

$$n_y = \frac{1}{4(\sin\theta_5.\cos\theta_4)}$$

Equation 40 Simplified comparison

$$\frac{1}{4.n_y} = (\sin\theta_5.\cos\theta_4)$$

Equation 41 Re-organising the equation

$$\frac{1}{4.n_y.\sin(\cos^{-1}n_x + \pi/2)} = \cos\theta_4$$

Equation 42 Re-organised equation

$$\theta_4 = \cos^{-1}\left(\frac{1}{4.n_y.\sin(\cos^{-1}n_x + \pi/2)}\right)$$

Equation 43 Calculation of tilt value

Once the values for rotation and tilt have been calculated, the transformation matrix can be used with a point in model space to generate the position of a point within the global workspace which the robot would recognise. It must be noted here that multiple solutions are available for the tilt and then for subsequent joint angles. This will be discussed later in this thesis. The calculation of the model space position moved into the global workspace is done by substituting the values for θ_4, θ_5 into the transformation matrix and then multiplying the transformation matrix by the model space position information Equation 44.

$$T_{TiltTable}x\begin{bmatrix}1 & 0 & 0 & Xposition\\ 0 & 1 & 0 & Yposition\\ 0 & 0 & 1 & Zposition\\ 0 & 0 & 0 & 1\end{bmatrix} = \begin{bmatrix}1 & 0 & 0 & Xglobal\ position\\ 0 & 1 & 0 & Yglobal\ position\\ 0 & 0 & 1 & Zglobal\ position\\ 0 & 0 & 0 & 1\end{bmatrix}$$

Equation 44 Conversion from local coordinates to global coordinates

Where $T_{TiltTable}$ uses values calculated for joint angles and X, Y, Z position is in model space. X, Y, Z global position is in the base reference coordinate system i.e. robot space.

5.5.2 Inverse Kinematic Solution to SCARA Robot

Having calculated the position of the points in the global workspace, the calculation of the required joint angles for the SCARA robot may now be done. Using simultaneous equations the robotic joint angles θ_1, θ_2 can be calculated for any point that is within the robot's reach this is done by taking the global position of the point and solving for Equation 46 and Equation 47. At this stage a computer and maths package (*maple*) was used to solve the simultaneous inequalities, this generates two solutions for each point (if faced with a choice the system is programmed to take the right-hand solution) unless it is at the maximum reach of the robot where only one solution exists:

$$\begin{bmatrix} Cos(\theta_1+\theta_2+\theta_3) & -Sin(\theta_1+\theta_2+\theta_3) & 0 & L2.Cos(\theta_1+\theta_2)+L1.Cos(\theta_1) \\ Sin(\theta_1+\theta_2+\theta_3) & Cos(\theta_1+\theta_2+\theta_3) & 0 & L2.Sin(\theta_1+\theta_2)+L1.Sin(\theta_1) \\ 0 & 0 & 1 & dH+H2+H1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} n_x & s_x & a_x & Xglobal\,position \\ n_y & s_y & a_y & Yglobal\,position \\ n_z & s_z & a_z & Zglobal\,position \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equation 45 Robotic arm matrix equated to global position matrix

$$\therefore Xglobal\,position = L2.Cos(\theta_1+\theta_2)+L1.Cos(\theta_1)$$

Equation 46 Simultaneous calculation of robotic joints

$$\therefore Yglobal\,position = L2.Sin(\theta_1+\theta_2)+L1.Sin(\theta_1)$$

Equation 47 Simultaneous calculation of robotic joints

Finally the calculation of the prismatic joint can be done again using the global position and the calculation of the final joint angle θ_3 for the end effector of the robot can be done by evaluating the normal vector again.

$$\therefore Zglobal\,position = dH+H2+H1$$

Where H1 and H2 are fixed based on the robot height see Figure 5-5

Equation 48 Calculation of end effector z-height

$$\therefore n_x = Cos(\theta_1+\theta_2+\theta_3)$$

$$\cos^{-1} n_x = \theta_1+\theta_2+\theta_3$$

$$\cos^{-1} n_x - \theta_1+\theta_2 = \theta_3$$

Equation 49 Calculation of end effector rotation

5.6 Redundancy

When the robotic arm can reach a given model-space position with more than one configuration the system is said to have a redundancy. The most common case of redundancy occurs within the robotic arm where most positions can be attained with either a left or a right bend on the links resulting in two solutions. Redundancies tend to occur when a system has a greater number of joints than there are constraints on the system (i.e. When a 3 DOF robot is tasked with positioning an object on a line (a one DOF task)) (51).

When solving the inverse kinematics of the tilt table, one such redundancy was obtained when two possible solutions were obtained for the angle of tilt. Each solution was valid however, only one solution would be appropriate for this task. Therefore when analysing the results it is important to impose constraints that exist within the mechanical system to provided viable solutions. In the case of the tilt table, a constraint maybe applied to the system so that all solutions must fall within the mechanical arc of the system (i.e. between $-\frac{\pi}{3} \ll \theta_4 \ll -\frac{\pi}{3}$) (in other words, forbid the model from rotating under the table). A second redundancy that occurred was the well-known robotic arm left and right solutions (aka the lefty and righty). These can cause damage to the workspace when not treated appropriately, as the arm can rapidly oscillate from the left-arm approach to the right. Therefore, it is important to stipulate which solution the system should select; throughout this work, the right-hand solution has been chosen as a convention.

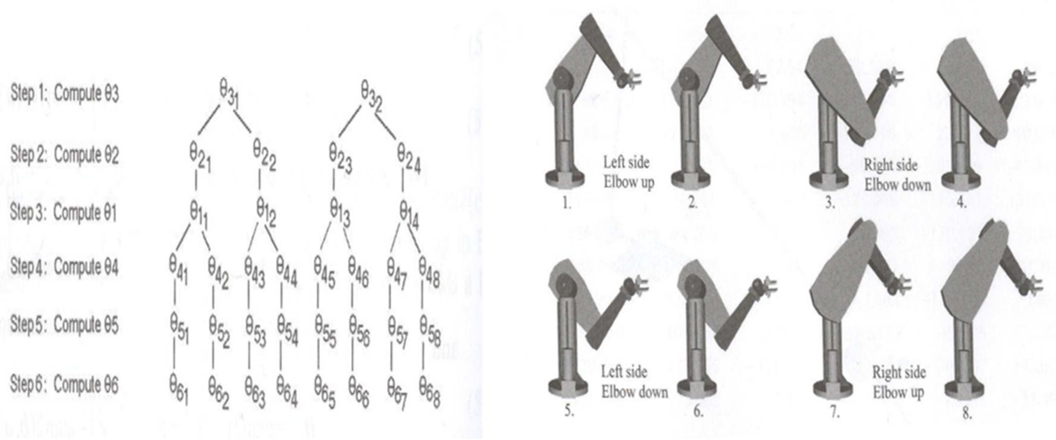
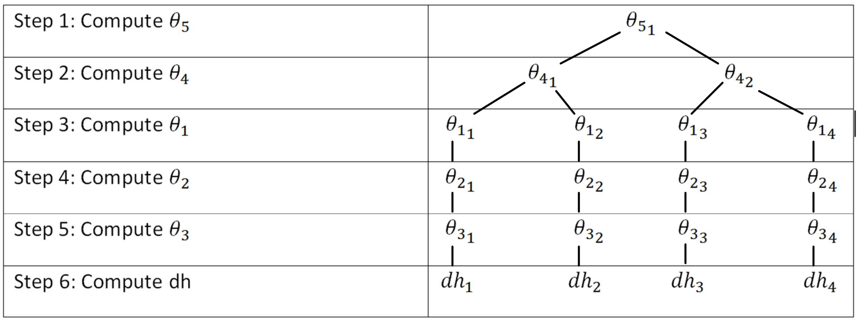


Figure 5-7 Puma Robot with Six Degrees of Freedom and Solution Tree (51) Showing Solutions

Manseur (51) constructs an inverse kinematic solution tree to demonstrate the number of possible solutions that can be obtained in a six degree of freedom Puma robot Figure 5-7 for their setup. The maximum possible number of solutions for any given point within the system's reach was found to be eight. Implementing a similar tree, it can be seen from Figure 5-8 that for the four degree robot and two degree tilt table, there will be a maximum of four possible solutions for any given point. This is however, the maximum number of solutions and is not always going to be obtained; for example the tilt table creates two solutions but this can be managed as the tilt has a limited amount of movement. Similarly, the second redundancy is the robot where two possible configurations exist; a left and a right solution, throughout this work the right solution shall be selected where possible.



Step 1: Compute θ_5	θ_{5_1}			
Step 2: Compute θ_4	θ_{4_1}		θ_{4_2}	
Step 3: Compute θ_1	θ_{1_1}	θ_{1_2}	θ_{1_3}	θ_{1_4}
Step 4: Compute θ_2	θ_{2_1}	θ_{2_2}	θ_{2_3}	θ_{2_4}
Step 5: Compute θ_3	θ_{3_1}	θ_{3_2}	θ_{3_3}	θ_{3_4}
Step 6: Compute dh	dh_1	dh_2	dh_3	dh_4

Figure 5-8 Representing the Level of Redundancy with this System

5.7 Degeneracy

Degeneracies are defined as points where an infinite number of solutions exist or where control over a single degree of freedom is lost. Most robots have mechanical stops to prevent certain degeneracies from occurring. One such example is when the two primary links of a SCARA robot have the same length. In this case, when the second joint angle is 180 degrees, there are an infinite number of solutions for the point at the centre of rotation for the first link; therefore most robots impose mechanical stops to prevent this case from occurring, this is the case on the experimental robot.

The main degeneracy within any system is set by the limits of the system. All points outside the reach of the robot are defined as degenerate points. Within the two-chain system used, several other cases of degeneracy exist. When the end effector is placed on the model space home position and at all points along the z-axis of this home position, the system would be in a degenerate configuration as the rotation of the end effector and the rotation of the table would produce an infinite number of solutions. Therefore, it is important that one principle joint is used to perform rotation with the other joint remaining at a fixed position. This is why the rotation of the tilt table handles orientating the component along the y-axis before the robot orientates the end effector.

This chapter has defined four different path planning methods (Lamina, Cartesian closest point, single joint and Configuration based closest point) and presented the system’s forward and inverse kinematic equations. These expressions will be validated against two test components in the next chapter before testing each of the algorithms on several physical models.

Chapter 6 Implementation and Results

This chapter presents the implementation of the sequencing approaches described in the previous chapters and discusses the results obtained when they were applied to benchmark components. The results for each of the different planning strategies (Lamina slice, Single joint sequenced, Cartesian closest point sequencing and Configuration space sequencing) are presented for each component and the results are discussed at the end of the chapter.

6.1 Implementation

The previous chapter outlined four different approaches to sequencing points and also derived the inverse kinematic equations so that the joint angles could be calculated. This section describes the mechanisms used for implementing the different sequencing algorithms and the specific software used to obtain results.

In Chapter 3, the system was outlined and the architecture was shown where a two-stage approach was used to process the points; firstly a data collection and secondly a data processing module were presented. Each of these units operated using a 10D data array, which in the collection unit was output in the form of a text file (known as the point file) and in the processing unit was read to obtain the movement instructions for the system.

The data collection unit creates the 10D data array based on the lamina slicing algorithm described in section 3.3.1.2. The output from this module contains all the data required to paint a component however, it does not contain the specific joint angles for each point. ‘*Maple*’ a computer mathematics package (137) (138) was used to read the 10D data array and perform the inverse kinematics calculations to calculate the joint angles.

The Maple programme was written in the form of a loop and calculates the T2 value first, then the T1 values (where two values tend to be found). The T1 values are tested to see if the first value is between the values of -$^{\pi}/_{2}$ and $^{\pi}/_{2}$ (i.e. ±90 degrees) if the first value is not then the second value for tilt is used for subsequent calculations (it is assumed the shift between values is 180 degrees). Using the values for T1 and T2 the position of the end effector is calculated in robot space (i.e. the coordinate frame from the base of the robot) and then the two chiralities are calculated for the robotic joints R1 and R2.

The Maple loop outputs a new 6D file (known as the configuration file) holding the robotic configurations for each of the points in the point file. Where lamina or single joint sequencing is being performed the data is then read into ‘*Microsoft Excel*’ where the values for the total joint movement are calculated and sequencing can be performed for single joints using the spreadsheet sort command.

Unlike the single joint or lamina approach, the Cartesian and the Configuration space closest point algorithms require an additional piece of software which was custom written to read in parallel both the configuration file and the point file. The programme was written as a command line application using C++ to temporarily read both files into a single array and then perform a multi-dimensional closest point search using the first point in the file as the start point. Sequencing the points based on the nearest un-used point to the current point. The sequenced points are then output back to the point file with the inclusion of the joint angles, the point file is then loaded into Excel to calculate the joint travel. The new system architecture can be seen in Figure 6-1 where the introduction of a new layer is added to calculate the joint angles.

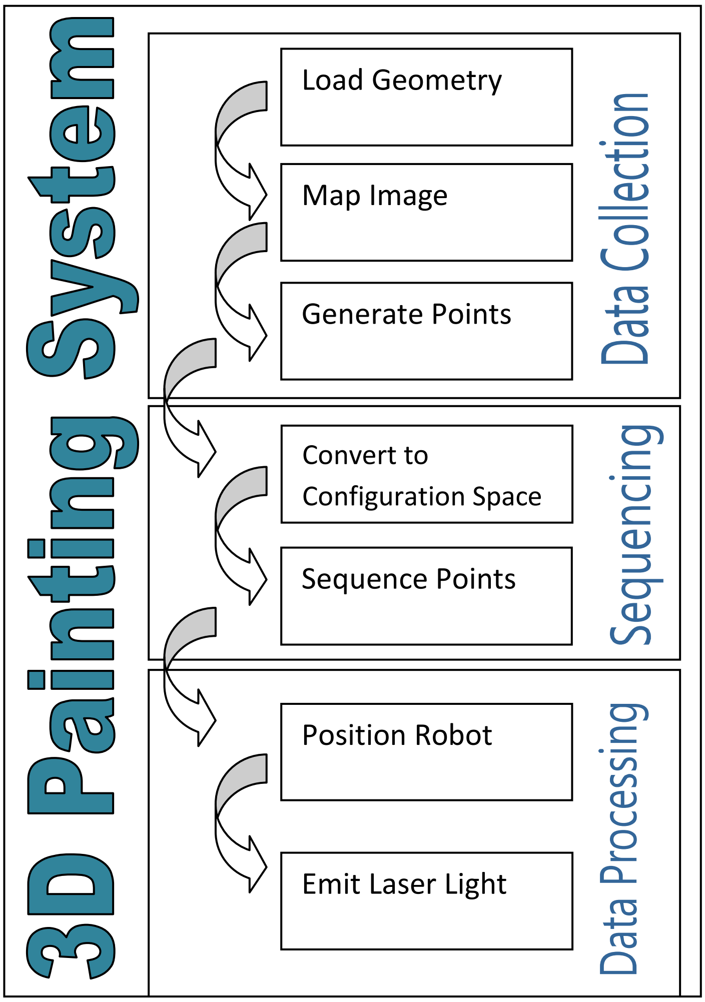


Figure 6-1 New system architecture

6.2 Results

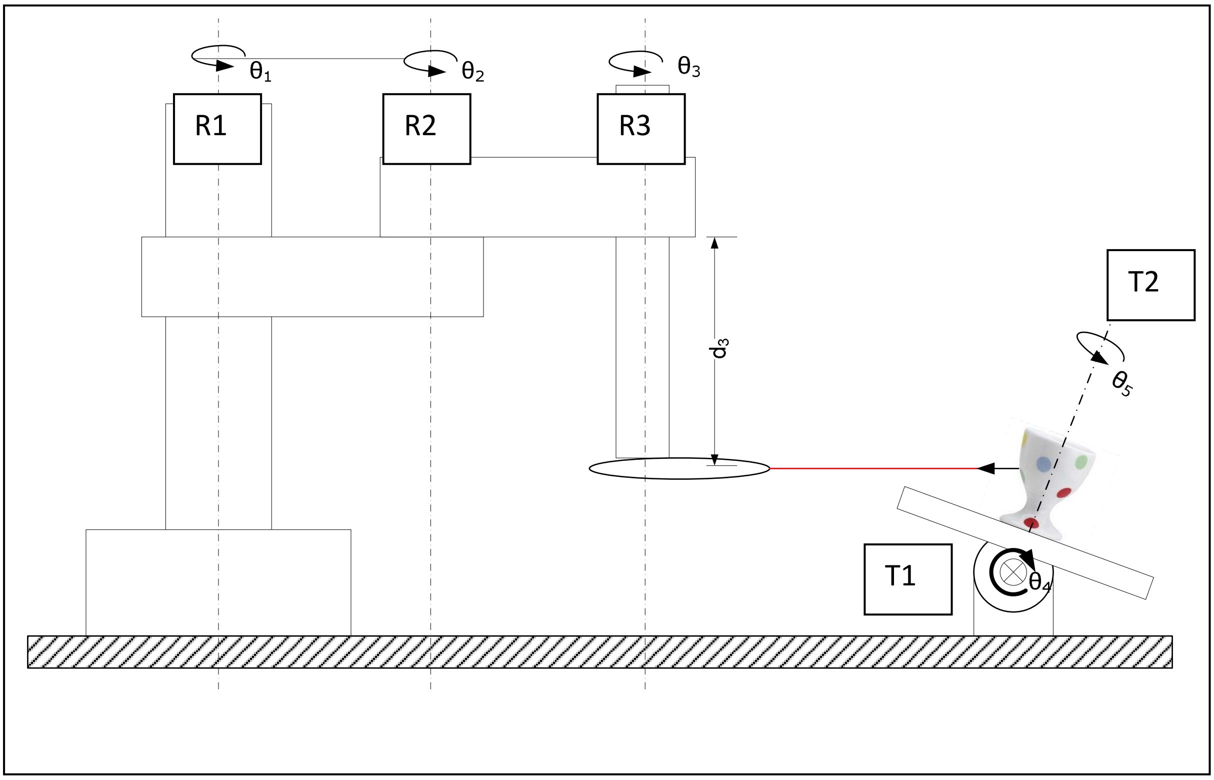


Figure 6-2 Schematic setup of robot indicating robotic joints and joint names

This section details the results obtained from sequencing the points based on the approaches outlined earlier in this thesis. Each of the different point sequencing algorithms has been applied to a number of different components. Ceramic bisque test parts were used because good results had previously been obtained using the photographic emulsion which adhered without the need for surface preparation on this material. Two additional components have been used which are based on synthetic computer-generated models that have been constructed specifically for the initial testing of the system.

The components can be seen in Table 6; the first row shows one of the synthetic models which is a cylindrical component. This model will allow the system to be validated using what is effectively a perfect mesh (i.e. no anomalous surface defects) whose results can be easily assessed (i.e. no values for tilt will be obtained). The cylindrical model was processed at a resolution of 10 dots per inch (dpi) with only the vertical cylindrical surface being generated (as the tilt table is mechanically unable to obtain an angle of 90 degrees which is needed to process the flat top of the model). The second test component in the table is another synthetic model generated from a warped cylinder. The deformed cylinder is a more complex model as it introduces the tilt aspect of the system, this means that the position of the end effector will move substantially. The deformed cylinder was also processed at 10dpi and the fourth column of the table shows the resulting point set size.

The four physical models used to benchmark the system were scanned using a Minolta vivid 910 scanner, which is a non-contact optical time of flight scanner. The scanner has an accuracy of 0.05mm this generates file sizes significantly larger than those of the "virtual" models shown in rows 1 and 2. The third component is an eggcup that produced a good surface mesh to represent the component; again, the data did not include the top of the model. The data was extracted at 10dpi and included some points obtained that were out of "*reach*" of the robot (i.e. the eggcup has a chamfered base which produces surface normals that pass through the base of the tilt table and are therefore in-accessible). These inaccessible points will be discussed in later sections.

The final two models are both figurines, one being a penguin and the other a wizard, these have been selected as they have relatively complex geometries with re-entrant areas and irregular surface normals. The penguin model has areas with smooth surfaces which quickly change normal orientation, providing a challenge for the path planning system to re-organise the points in a fashion best suited to the system. The penguin has many challenging features that significantly change the configuration of the robot as it moves between points that are within close proximity (i.e. the beak and the wings).

Each of the models has been selected to assess the capabilities of the system and test its adaptability to different geometries. It is expected that the results will show the greatest improvement for the more complex geometries; where significant re-configuring is required between points. However, it is also predicted that for simple regular geometries, such as the cylinder, a better approach may be found that visits the points in a more efficient manner such as processing all the points in vertical strips rather than horizontal strips. The next section will look at the results obtained and describe the robotic motion that was seen during the implementation of the different path planning algorithms.

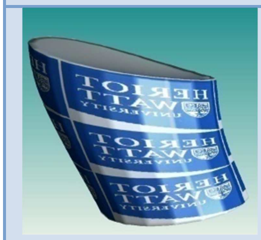
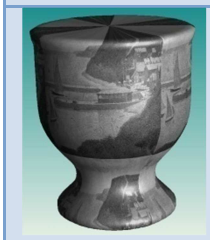
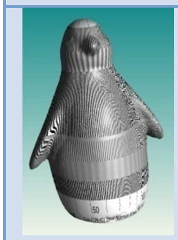
Component	Name	File Size	Number of Points	Virtual Or Physical
	Cylinder	901 Kbytes	1440	Virtual
	Validation component to look at the accuracy of the results and specifically concentrate on the T2 values obtained from the inverse kinematic chain			
	Deformed Cylinder	935 Kbytes	1443	Virtual
	Second validation component to analyse the results obtained for tilt and the robotic end effector. The model maintains the same faceted surface as the cylinder.			
	Eggcup	977 Kbytes	1500	Physical
	The eggcup is the first scanned model and will be used to look at the possibility of using single-joint sequencing for swept profile components (e.g. rotated).			
	Penguin	1622 Kbytes	1702	Physical
	The penguin is a much more complex model with a more varied surface and will be used to analyse the potential of both the cartesian and configuration space approaches.			
	Wizard	4248 Kbytes	2430	Physical
	The wizard will be used to investigate the different approaches and the knowledge gained throughout the experiments, specifically concentrating on the closest point algorithms.			

Table 6 Test components and detail descriptors

6.2.1 Validation 1: Cylinder Test

The objective of the trials conducted with this test cylinder was to validate the equations developed in Chapter 5. The cylinder was used for this because the accuracy of the results could be determined by inspection.

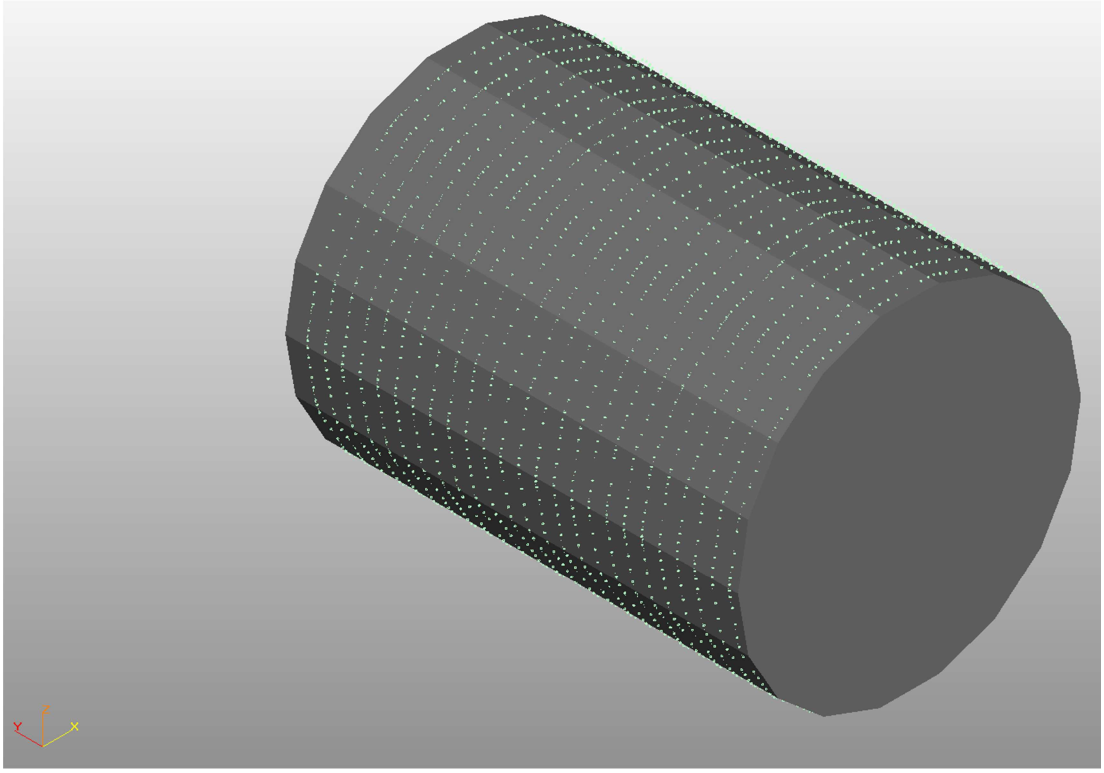


Figure 6-3 Position of points extracted from cylinder at 10dpi

The cylindrical object is a synthetic model which uses an 18-sided polygon to approximate the appearance of a uniform cylinder. Figure 6-3 shows the location of all the points extracted from the cylindrical model by the N-algorithm. To enable testing, these points will be treated as positions on the surface of the model that must be visited (thus simulating the creation of images on the surface). The figure also verifies the regularity of the data generated by the lamina slice procedure. The surface points cloud contains 1440 points, each of which is a 10D array, holding position, orientation and textural information.

6.2.1.1 Lamina Slice Path

Figure 6-4 and Figure 6-5 show the connectivity of the points after a path is interpolated through them. As the model is an approximation of a cylinder (i.e. comprised of many flat facets) it can be seen that several points on each layer have the same normal. Figure 6-4 shows the "seam" that is created when the algorithm moves down the model; depending on the first triangle found in the connection map the path will either go clockwise or counter clockwise.

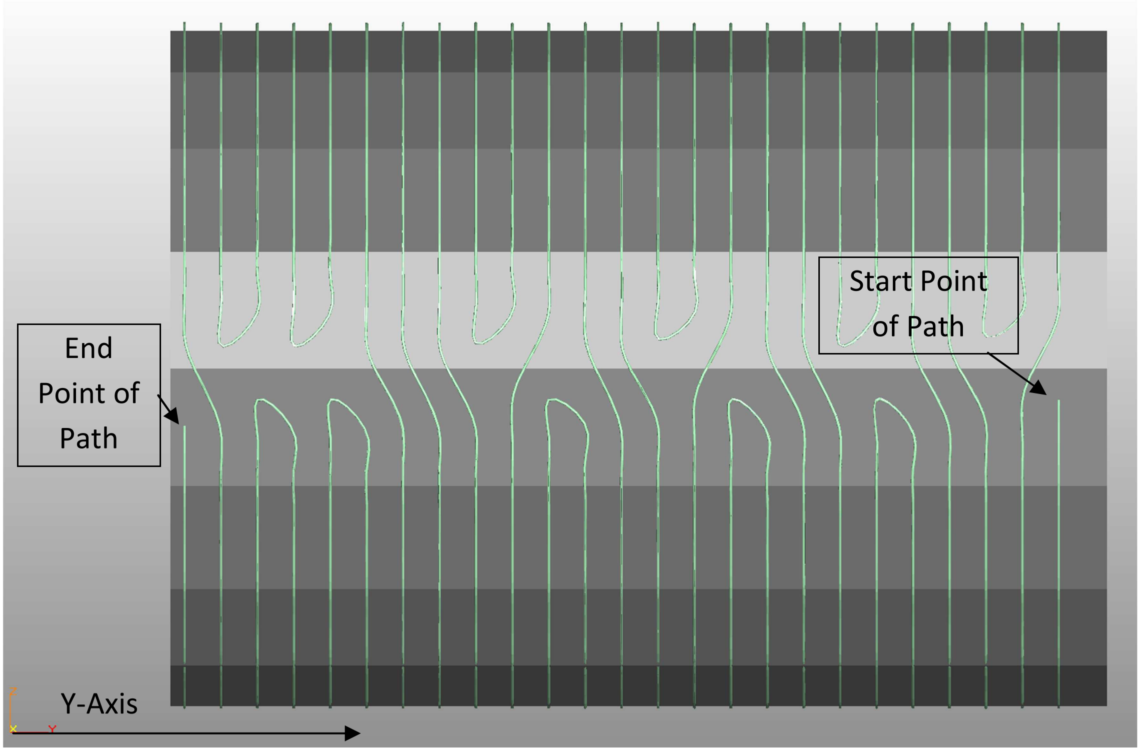


Figure 6-4 Lamina path side view

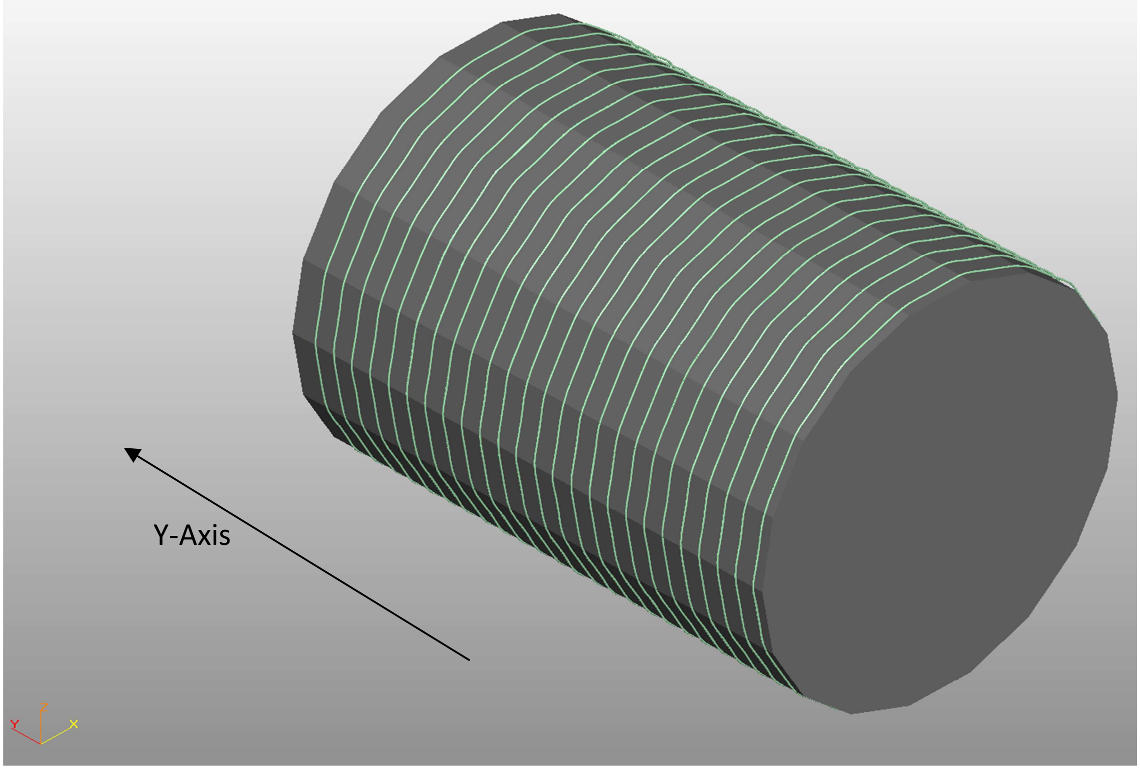


Figure 6-5 Lamina path perspective view

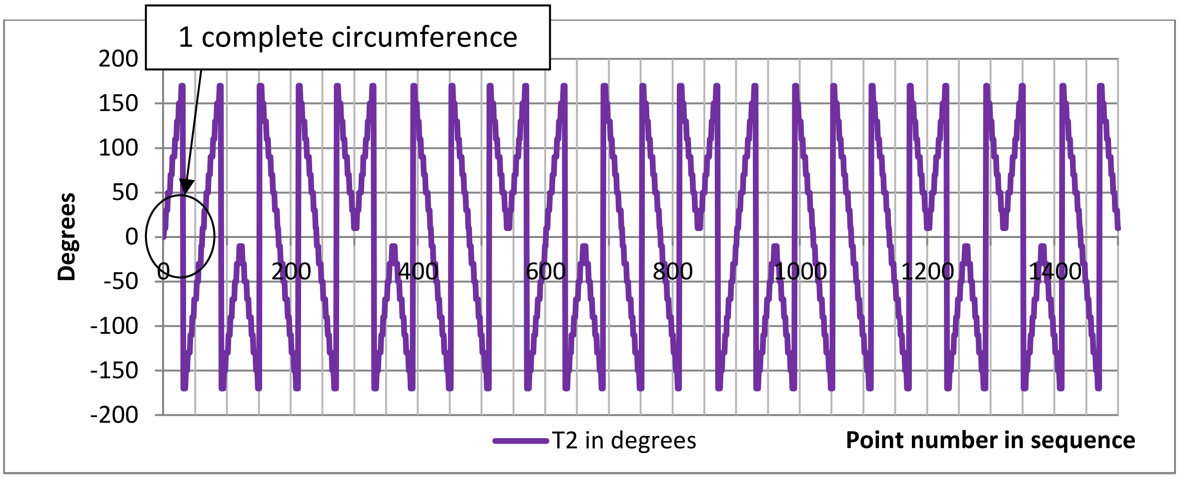
However, the position of the robotic end effector in the configuration space is equally as important as the Cartesian position of the points on the model (these coordinates will be used in planning the movement of the robot). Using point coordinates and surface normal vectors, the inverse kinematics of the tilt table can be used to calculate the T2 value for each of the points (i.e. the angle of rotation on the tilt table). The values obtained will be the angle between the normal vector and the x-axis. The results for a circular profile oscillate between 0 and π. They do not however, identify in which direction the rotation has occurred i.e. -180 degrees provides the same solution as +180 degrees. Further sign checking (i.e. looking at the quadrant the normal points towards) is required to calculate the actual rotational values. The period of one wave represents a complete revolution of the model on the tilt table and so, by inspection, one can see that the cylindrical model (at 10dpi) uses 25 individual layers and, as the model has a uniform circumference, the period of each wave remains constant. There are approximately 60 points per layer.

Graph 1 shows the results for the T2 rotation in degrees after sign checking has verified the correct orientation of the normal. Inspecting the first 60 points, a steady increase in the angle can be seen till it peaks at 180 degrees. A sharp movement is seen from +180 degrees to -180 degrees though sharp on the graph this would not be the case in real life as this would be the table moving from one extreme calculated value to another which represents the same physical place on the table. The graph then shows the table performing a complete circle from -180 degrees through to +180 degrees, another 180 degrees is complete representing two complete circumferences of the model. After completing two circumferences the path then appears to have switched to a counter-clockwise approach, this is caused by the connectivity of the triangle when the points were extracted.

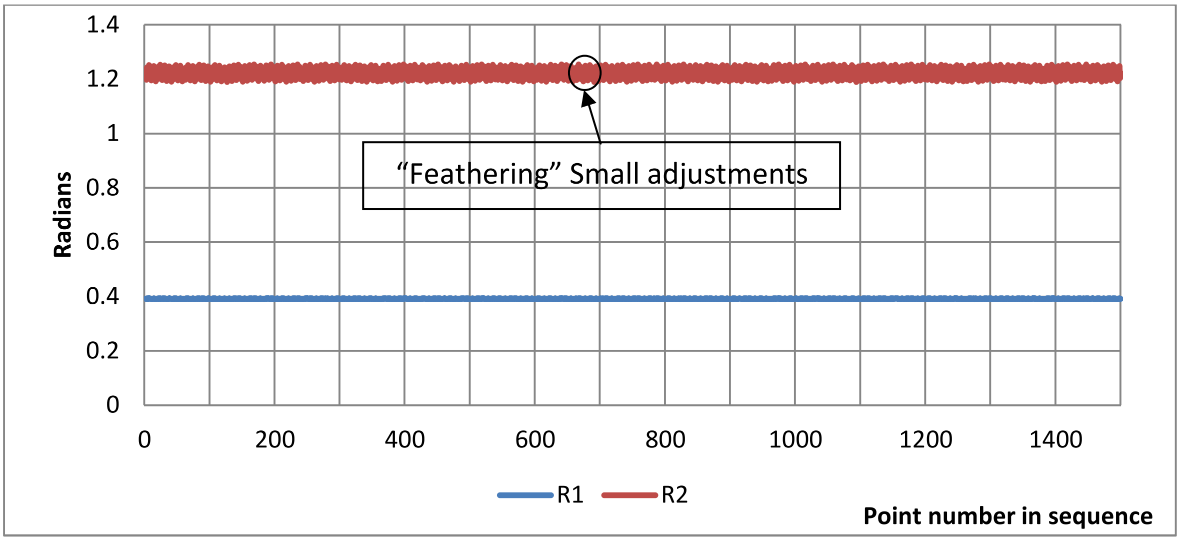
As the cylindrical component has vertical sides, the T1 angle (i.e. table tilt) remains at zero throughout (i.e. no tilt). The coordinates are converted into configuration space using the inverse kinematic equations described in Chapter 5. Using the resulting end effector position values, the joint angles for the robot are calculated. When a perfectly cylindrical model is used, it should be possible to obtain a single configuration for the major robotic joints R1, R2 with only a linear movement up and down, and a rotational movement for the table. However, this model is not perfectly cylindrical so small movements are required for the end effector.

Graph 2 illustrates that very little end effector movement is required for a cylindrical component so long as the rotational joint is repositioned for each point. A small amount of feathering is observed on the secondary robotic axis which can be explained by small adjustments to position for the normal.

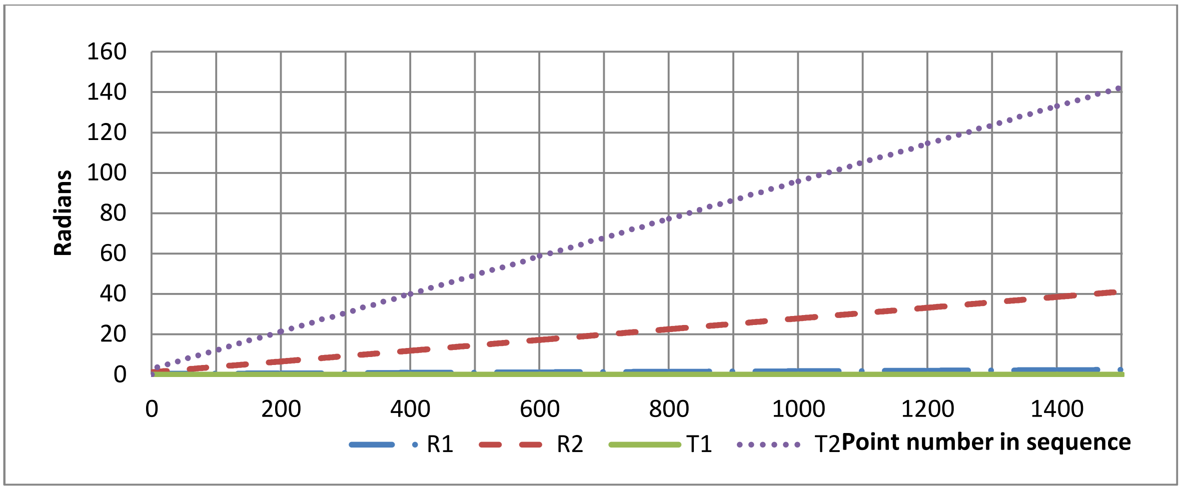
The results that have been obtained for the robot use the right-handed solution for the system (though a second solution does exist for the robot with the bend in the arm going to the left). The total distance travelled by each of the joints is plotted in Graph 3 where it can be seen that the T2 value (table rotation) contributes the greatest amount to the overall travel as would be expected with a regular cylindrical symmetry.



Graph 1 T2 joint angles in degrees for the cylinder model using the lamina slice path



Graph 2 R1 and R2 joint angles for the cylinder model using the lamina slice path



Graph 3 Total distance travelled by each axis in radians for the cylinder model using the lamina slice path

In addition to investigating the effects on the movements of individual axes, the reduction in largest joint movement required for each point to point command was also assessed in the following way.

The difference between the joint configurations for each pair of points in a path was calculated and the sum of the largest movements was used to calculate a value of max movement (as shown in Table 7)

Point	T2	R1	R2	Diff T2	Diff R1	Diff R2	MAX SUM
P1	1.7455813	0.3926700	1.2365094	1.7455813	0.3926700	1.2365094	1.7455813
P2	1.7455813	0.3916256	1.2176093	0.0000000	0.0010444	0.0189001	0.0189001
P3	1.7455813	0.3910085	1.1981333	0.0000001	0.0006170	0.0194760	0.0194760
P4	2.0942855	0.3932246	1.2441810	0.3487042	0.0022160	0.0460478	0.3487042
P5	2.0942855	0.3920056	1.2255142	0.0000000	0.0012190	0.0186669	0.0186669
P6	2.0942851	0.3912103	1.2062791	0.0000004	0.0007953	0.0192351	0.0192351
P7	2.4434626	0.3938664	1.2518333	0.3491775	0.0026561	0.0455542	0.3491775
P8	2.4434615	0.3924724	1.2333990	0.0000011	0.0013940	0.0184343	0.0184343
						Total	**2.5381753**

Table 7 Max move calculation showing the sum of all the largest joint movements for each point to point

The total sum of “Max move” values was calculated for the entire path. This number quantifies the distant travelled by the “dominant joint” in any coordinated Point-to-Point move. In coordinated Point-to-Point control (where the robot controller modifies the speed of each individual joint so they all move simultaneously, starting and ending at the same moment) the overall time for a movement is determined by the joint that has to travel furthest (assuming all joints have the same accelerations and max speed).

Although the exact detail of “coordinated point-to-point” control (e.g. rates of acceleration, max speed) will vary from robot to robot and from joint to joint it has been assumed (as a first approximation) that all other properties being equal *the time for a move from one configuration to another will be roughly proportional to the largest distance any joint is required to move*. Using this assumption the effects of coordinated path planning algorithms can be roughly assessed by the “max-move” metric. The Max move for the lamina path was found to be 164 radians.

6.2.1.2 T2 Minimisation Path

Figure 6-4 shows the lamina slice path and Graph 1 to 3 displays the results created for this approach. This section presents the results for a path generated by minimising the distance travelled by the T2 axis (i.e. Table Rotation) using a single joint path.

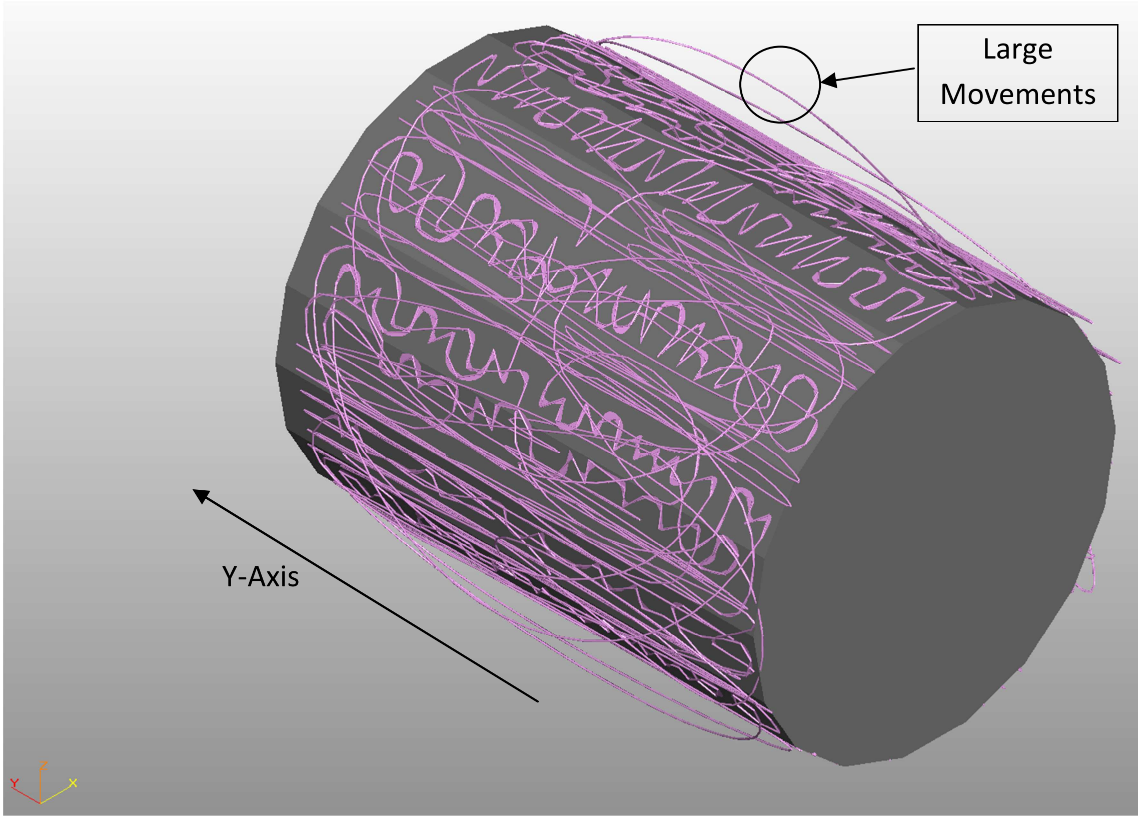


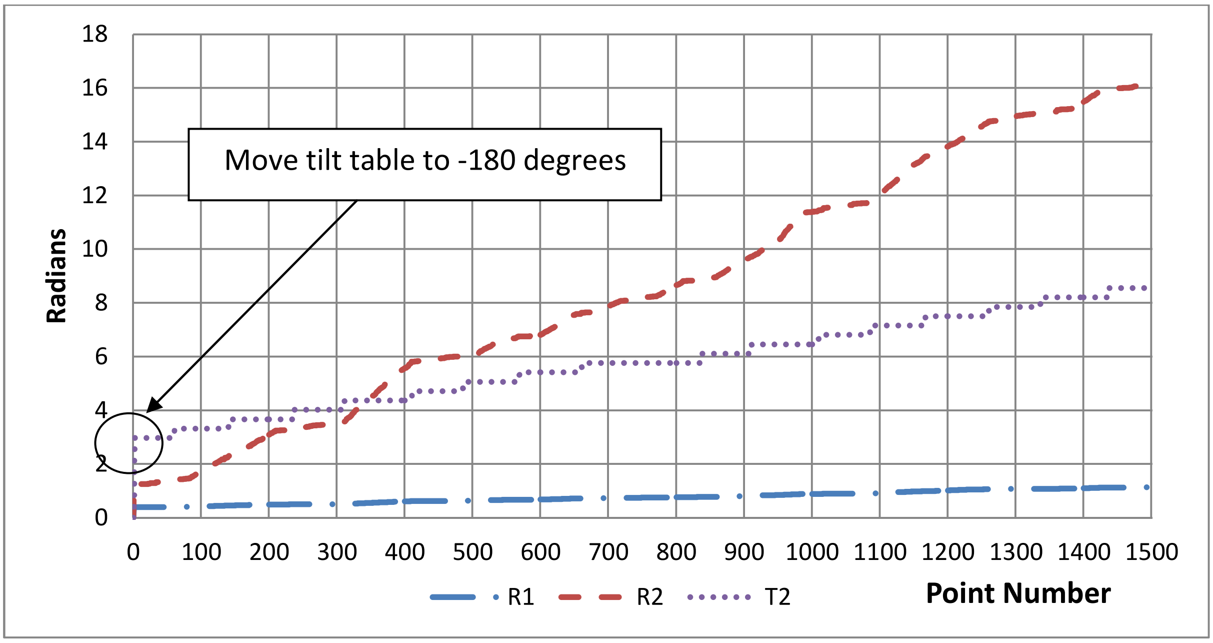
Figure 6-6 Perspective view of a single joint path using T2 for the cylinder model

This is the first of several different approaches for sequencing the points based on a single joint, in this case the T2 joint (i.e. the rotation of the tilt table). For the cylindrical model it is apparent that the movement of R1 and R2 is minimal for points sequenced by the lamina slice path. Graph 3 showed that all the movement took place around the R2 axis of the robot and the T2 joint of the tilt table (with by far the greatest movement occurring within the T2). Consequently, the sequencing of points in order to minimise T2 movement should reduce the total travel for the individual joint.

As Figure 6-6 demonstrates, the results for a single joint path using T2 creates a significantly different route to visit the points than the lamina slice. Graph 4 shows the change in the overall distance travelled, where it can be seen that the R2 axis becomes the largest moving axis with the T2 axis only moving enough to complete an entire circumference. Due to the single joint path algorithm the table must initially move to the -180 degree position (i.e. the value with the smallest magnitude) so an initial offset is seen on the T2 axis.

The path created using the single joint planning is visually more difficult to understand than the lamina approach, this is because the algorithm uses a rigid sequencing approach which looks at nothing other than one axis. When the points are viewed in the points cloud on the model at the beginning, (Figure 6-3) they appear to form not only radial slices but vertical slices as well. However, small variations on the individual triangles where the points are extracted mean that while using T2 should produce a vertically striped path (i.e. the opposite to the lamina approach which produces a horizontally lamina path) these small variations create a tangled path.

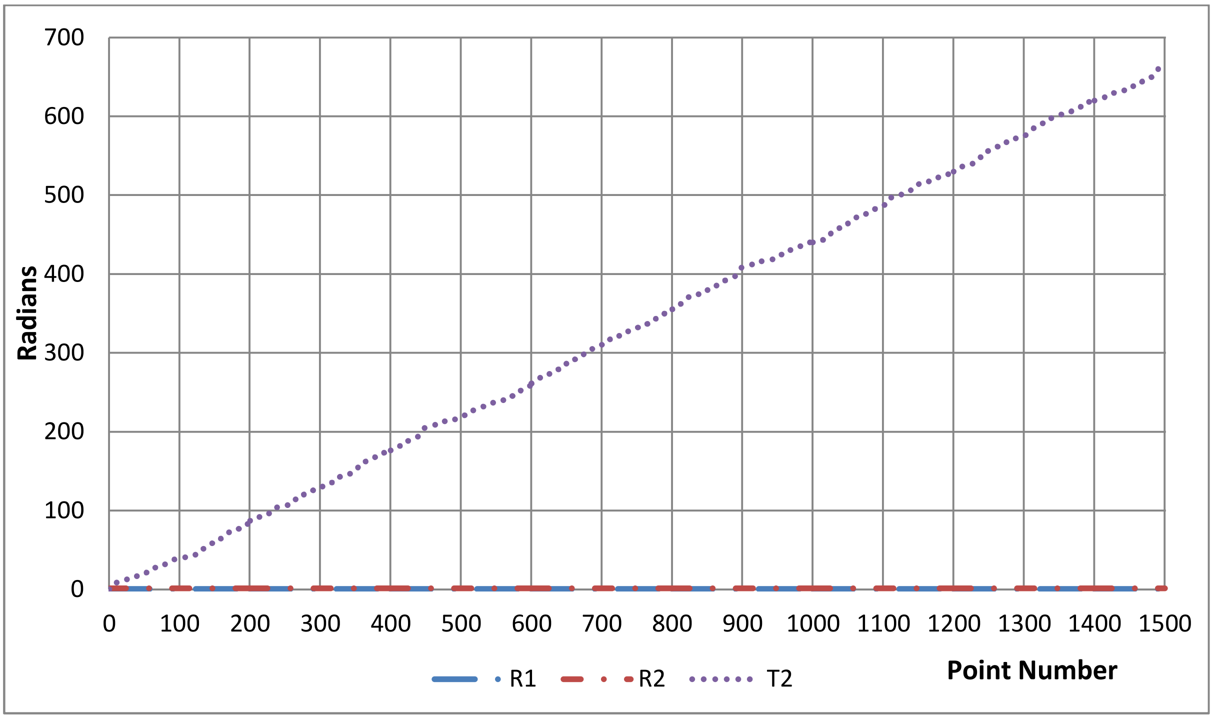
The path created is seven times shorter than the lamina path, which would suggest that it is important not to judge the path by its visual presentation as this may be misleading. The next section will look at a single joint path using a different axis to see if this reduction in path length can be obtained by sequencing using other axes. The Max move length for the T2 path was significantly smaller than the lamina approach at 23 radians, which would suggest that a large amount of the movement is coming from single axes, at each configuration change.



Graph 4 Cumulative distance for single joint planning using T2 for cylinder model

6.2.1.3 R2 Minimisation Path

While sequencing the results using the T2 joint has shown a decrease in the overall distance travelled, this is not replicated when sequencing is applied using other joints such as the R2 joint. Using R2 sequencing can be seen to increase the total path quite significantly, however, the distance travelled by the R2 joint is reduced to less than 2 radians of motion with the T2 joint increasing to 660 radians. This can be seen in Graph 5. The max move was calculated to be 663 radians which is made up of principally T2 movements.



Graph 5 Cumulative distance for single joint planning using R2 for cylinder (R1, R2 are all very small)

Graph 5 demonstrates the results for single joint planning using R2, though R2 and R1 appear to be negligible, the effect this has on the T2 is significant, where it is much larger than any previous approaches. Figure 6-7 shows the path obtained; similarly to the previous approach the path is very complex to visually follow. The next section will look at the Cartesian closest point approach which should be visually easier to understand as this is based on the nearest surface points to each other.

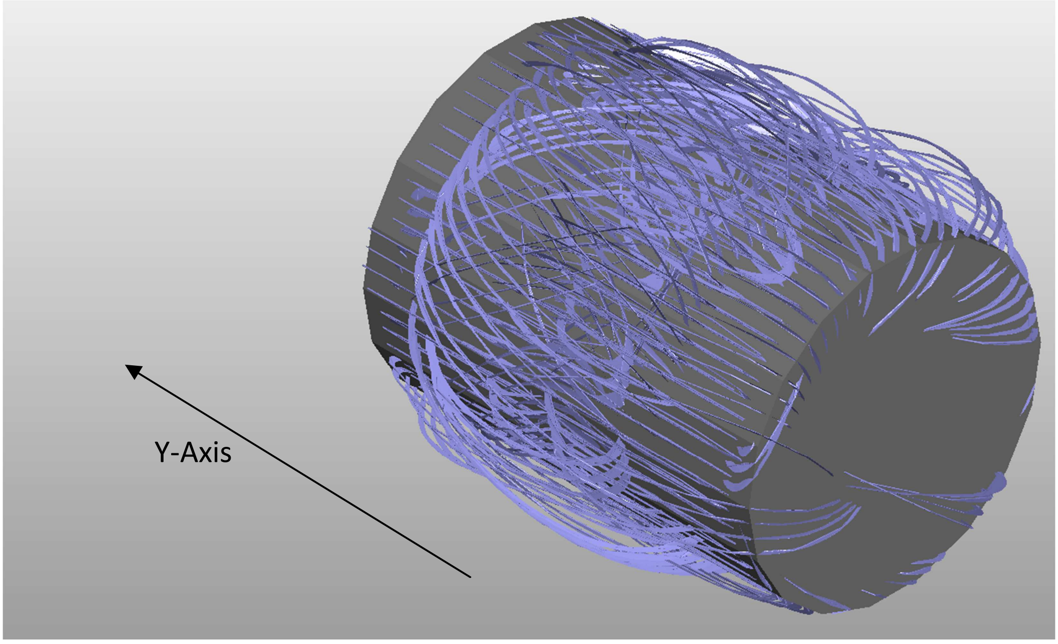


Figure 6-7 Perspective view of single joint path for R2 on the cylinder model

6.2.1.4 *Cartesian Closest Point Path*

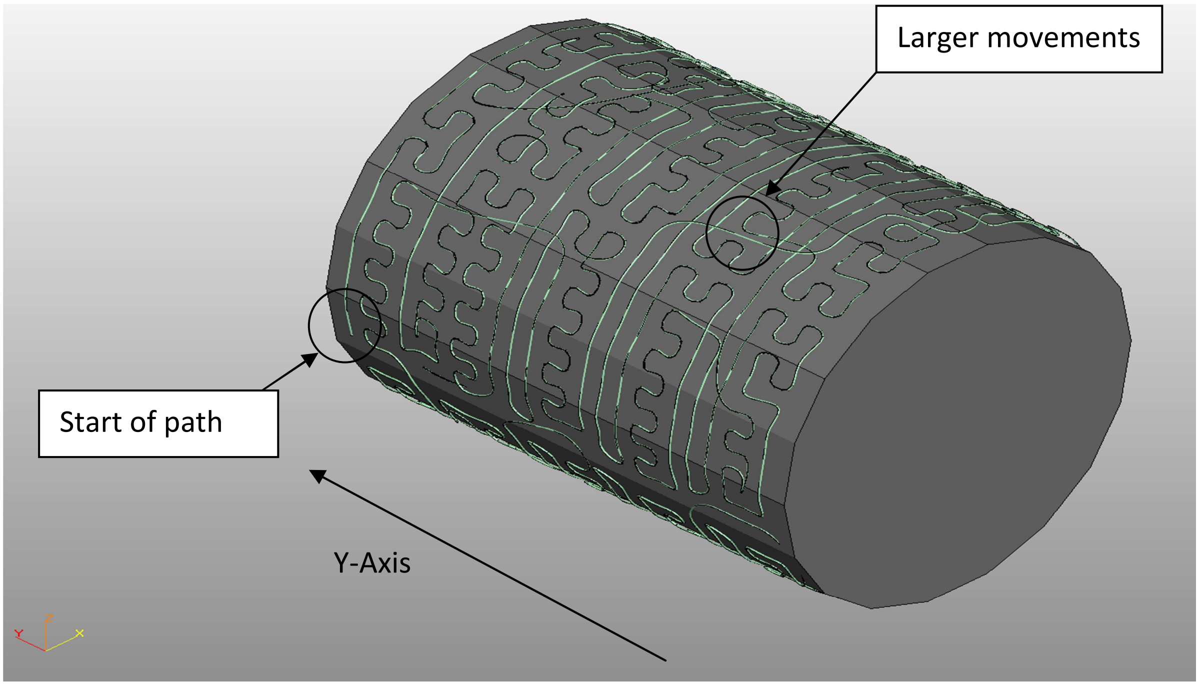


Figure 6-8 Perspective view of Cartesian closest point path for cylinder model

Cartesian closest point sequencing is a strategy that plans points based on their relative location in a Cartesian 3D environment. Therefore once a point has been visited, a search is conducted for the nearest unvisited point in all directions. Figure 6-8 shows a portion of the path that is generated when Cartesian sorting is applied, the path can be seen to move either around the component in a circumferential direction or along the y-axis up and down the model.

Figure 6-9 shows a single side of the cylinder where the path can be seen to move predominantly in a single direction to sequential points. However, in some areas it can be seen that the path must move a greater distance to find a point that has not been processed as all the surrounding points have already been visited. This requires the path to move a greater distance, perhaps because the points are not being generated across the surface with a uniform distance in all directions between them. Therefore when a point is closer on the layer below rather than completing the layer it would proceed to the next layer down.

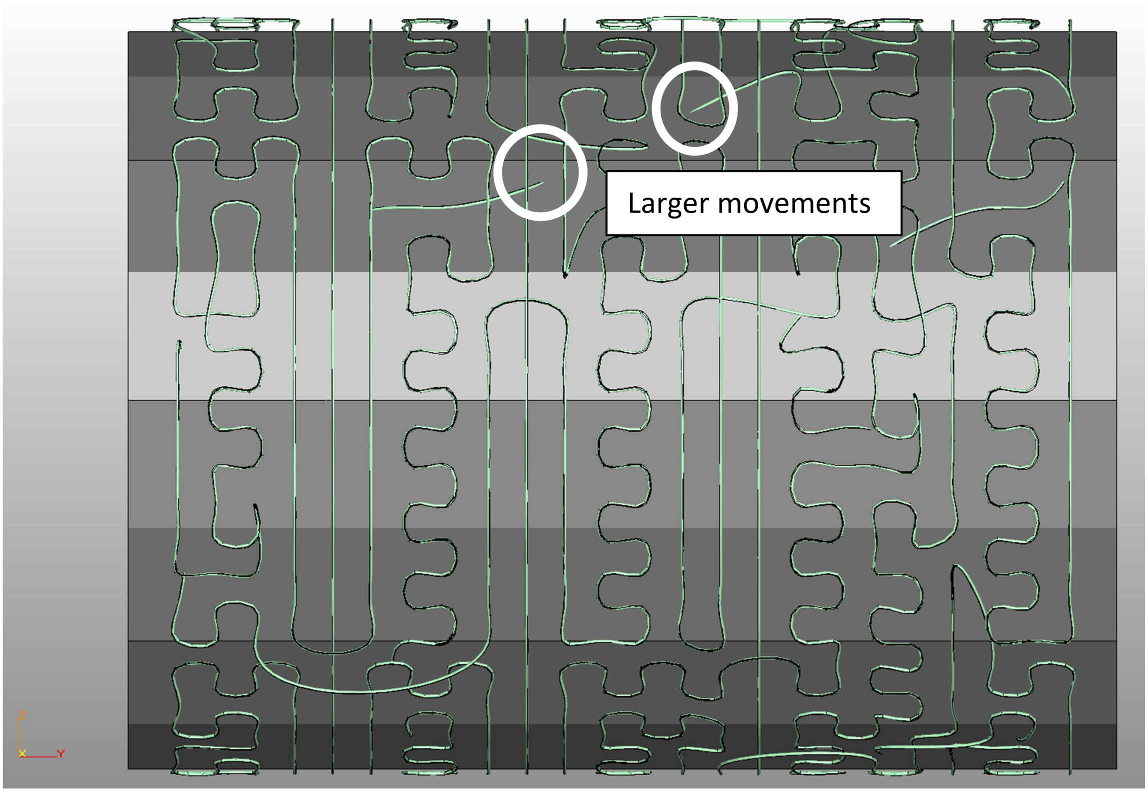
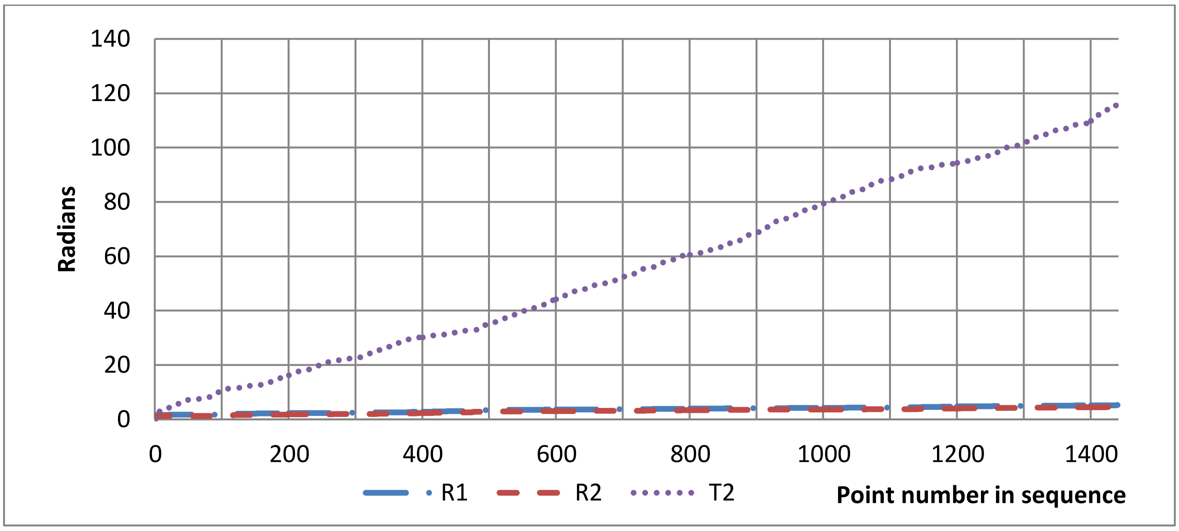


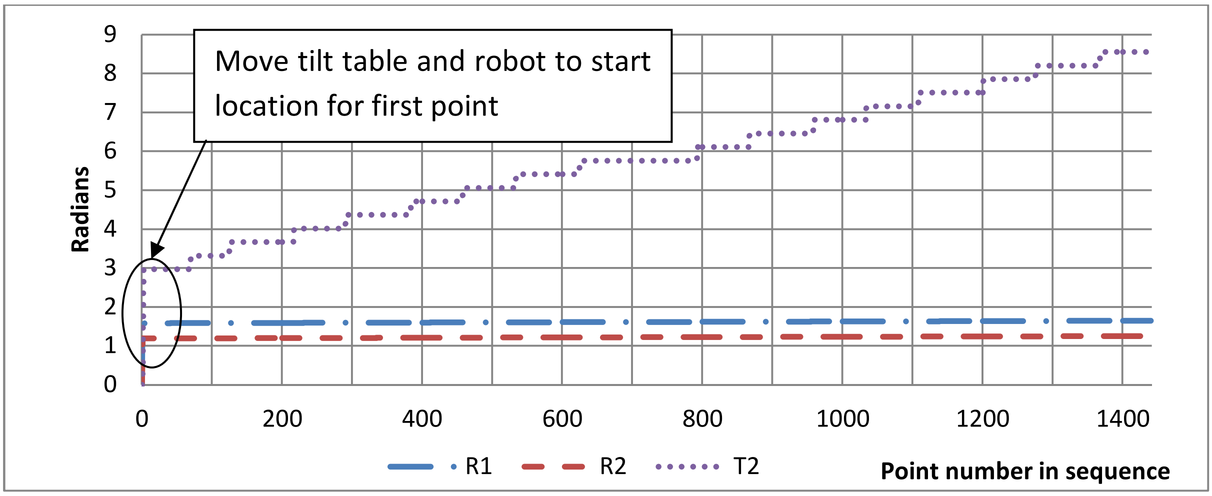
Figure 6-9 Side view of Cartesian closest point path on cylinder model

The overall Cartesian path shows a distinct similarity to the lamina approach and therefore the travel of each joint should be similar. It is possible to see from Graph 6 that using the Cartesian closest point planning has reduced the overall travel; the max move again made up a large amount of the total path at 121 radians. The T2 joint travels through 118 radians and the R1 and R2 joints travel through less than 10 radians each. The Cartesian planning created a shorter path than the lamina approach but it was much longer than the path generated by T2 minimisation.



Graph 6 Cumulative distance for Cartesian closest point for cylinder

6.2.1.5 Configuration Space Closest Point Path



Graph 7 Cumulative distance for Configuration space closest point for cylinder model

The last method investigated, again selected the next closest point but instead of using x, y, z coordinates, the points were defined in 4D configuration space (i.e. $\theta_1, \theta_2, \theta_3, \theta_4$).

Previous results have shown that configuration-based sequencing for individual joints can produce significantly different paths. So it is reasonable to speculate that using a closest point algorithm in the 4D configuration space could provide even better results as it takes into account (i.e. optimises) all axes rather than a single joint. The closest point algorithm evaluates the distance in radians between all the points and selects the one that is nearest in configuration space (i.e. the smallest cumulative distance in radians).

Graph 7 shows the results obtained for this approach and the overall cumulative distance can be seen to be reduced from both the original lamina sequencing (Graph 3) and the rotational point sorting (Graph 5). Figure 6-10 shows the path that is obtained using the closest point algorithm and it can be seen to be similar to the rotational organisation. However, using the closest point algorithm means that points can be processed in vertical lines and small variations in the normal and orientation do not impact in as large a manner as the stringent rotational sequencing.

Overall, the closest point algorithm in configuration space produces the best results with respect to distance (12 radians) and max move (8.7 radians). The results have also validated the kinematic equations used to model the robot and T2 for the table and subsequent sections will investigate if this outcome is repeated for other shapes of component. The different paths obtained all visit the same points but by the various path planning methods show that small changes to a single axis can create large changes in the overall sequence of points visited. The next section will look at validating the T1 joint.

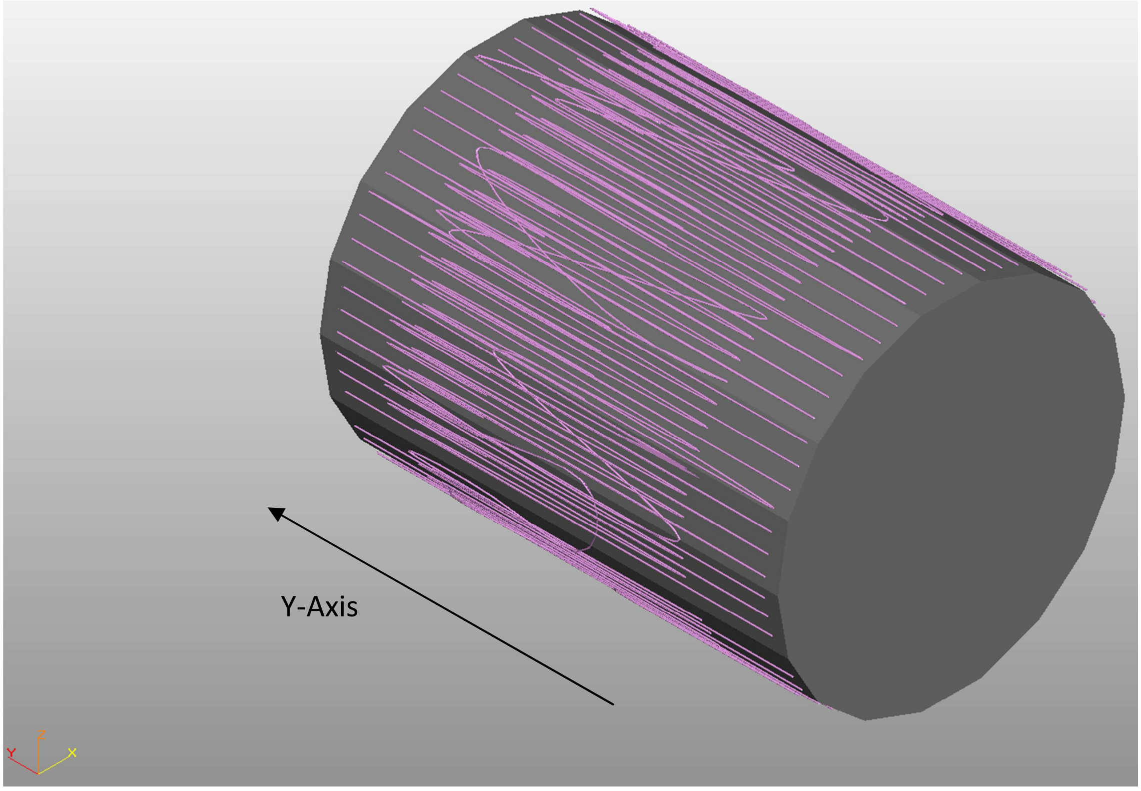


Figure 6-10 Perspective view of the configuration space path for the cylinder

6.2.2 Validation 2: Deformed Cylinder

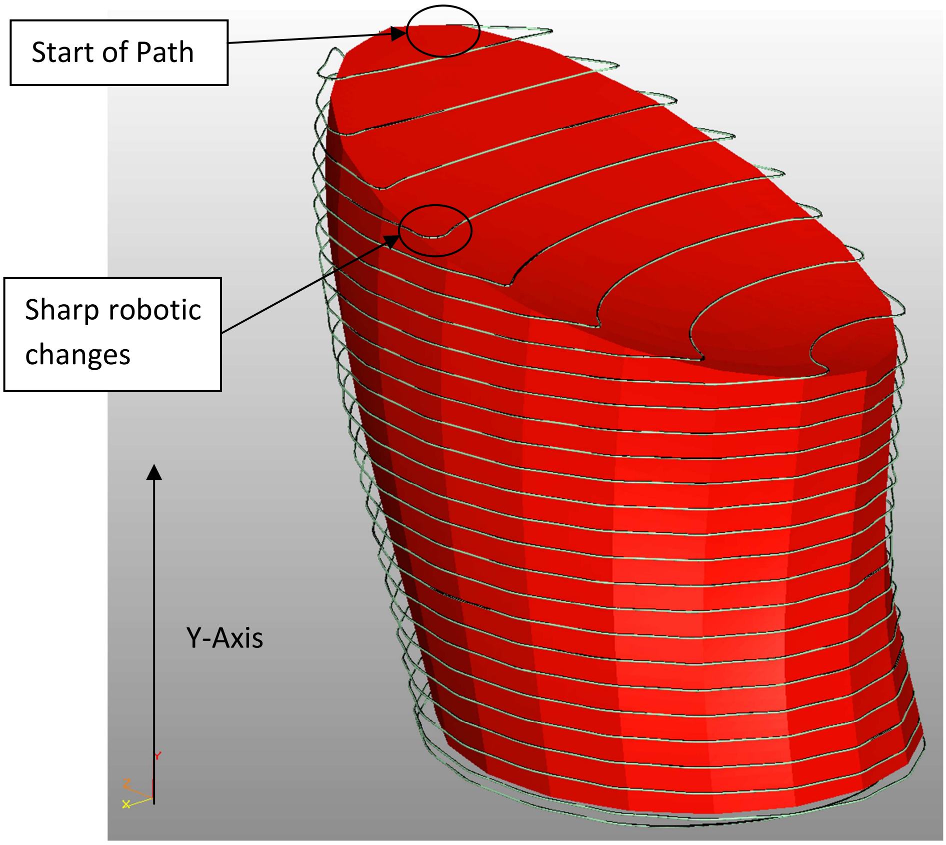


Figure 6-11 Deformed cylinder showing lamina slice path

This component is a synthetic shape generated from a warped cylinder to provide a more complex geometry than the cylinder, while still maintaining the integrity obtained from a computer-modelled component (i.e. no holes in the mesh or inverted normals). This model will allow the validation of T1 (table tilt values) and the investigation of the different path planning methods when applied to this more complex shape.

6.2.2.1 Lamina Slice Path

Figure 6-11 shows the connection of the lamina slice path for the surface points. Once again the path appears as multiple loops moving down the y-axis from the top of the component with even spacing between the layers. Figure 6-12 shows a side view of the surface path where the seam created by the mapping of the N-algorithm is visible down the spine of the component. The component can also be seen to have a convex and a concave side which will require the system to construct more complex paths than for the cylinder.

Graph 8 shows the values that were obtained for the tilt (T1) and Graph 9 for rotation (T2). The values calculated for tilt show quite a varied range during the first 400 points, with sharp increases which are accounted for by the change from the wrapped side of the object to the top face where the normals are significantly different to the sides. After 400 points, the normal becomes more cyclic as the points on the wrapped face of the deformed component are painted. As with component 1, the values for rotation (T2) can be seen to oscillate between ± π while the component maintains its cylindrical profile. However, the first 300 points produce slightly less continuous results as this represents the area of the model on the top surface where the approach angle swiftly changes as the path crosses from one face to another.

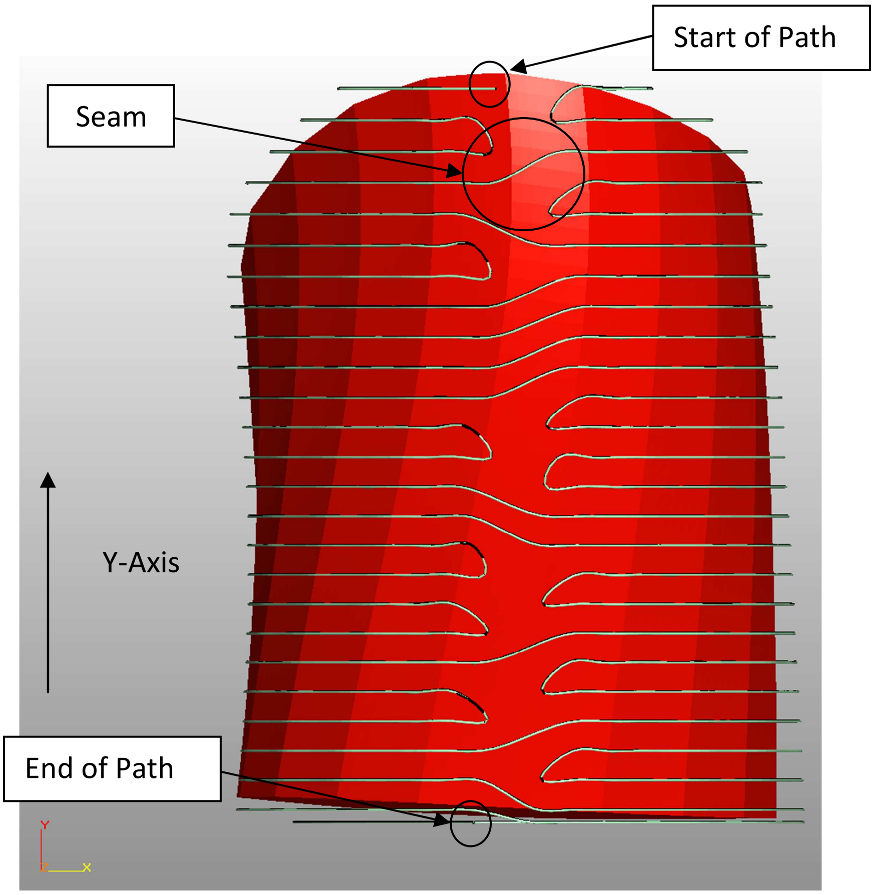
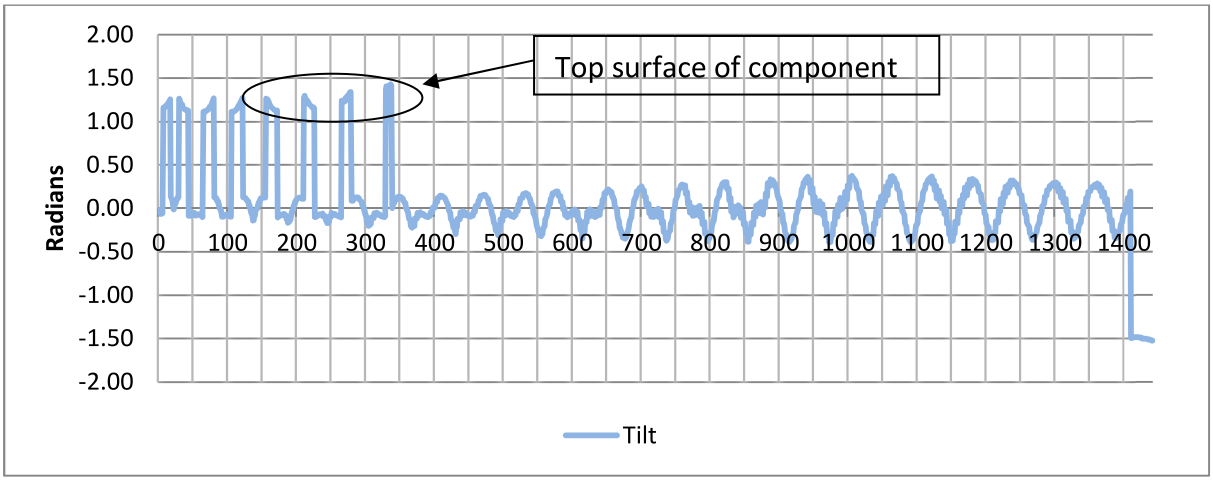


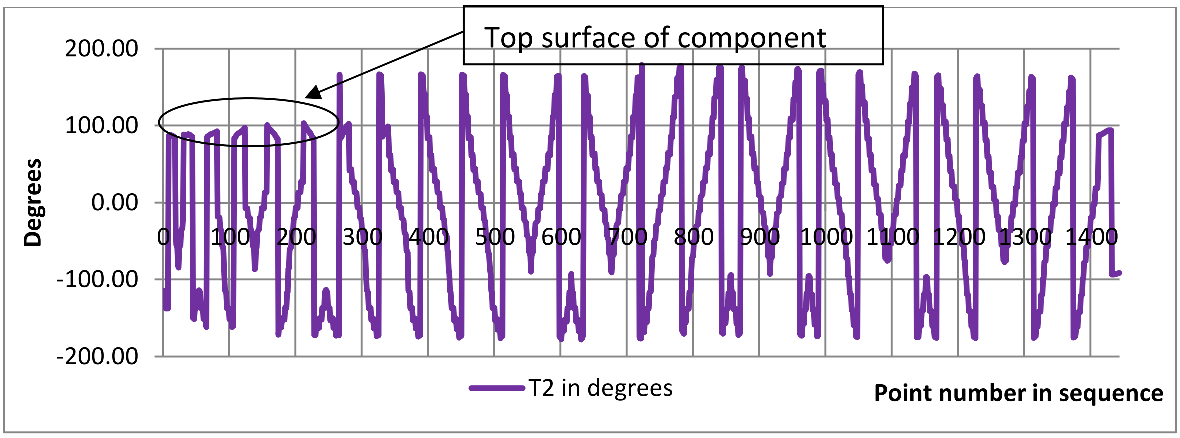
Figure 6-12 Side view of the seam on the deformed cylinder

The more complex shape of the deformed model produces a varied path for the robot with movement occurring in both the R1 and R2 joints (Graph 10). The results have shown a low range of movement for the R2 axis with slightly more movement occurring in the R1 axis. As the deformed component is based on a cylinder, the results would be expected to be in a similar pattern to the cylinder with slightly more movement due to the angular sides. This is seen in the later points where a regular pattern appears for the R1 joint with small movements for the R2 joint. The initial 300 points produce dramatic results which demonstrate the significant repositioning that the robotic end effector would be required to carry out to expose the points on the top surface of the model.

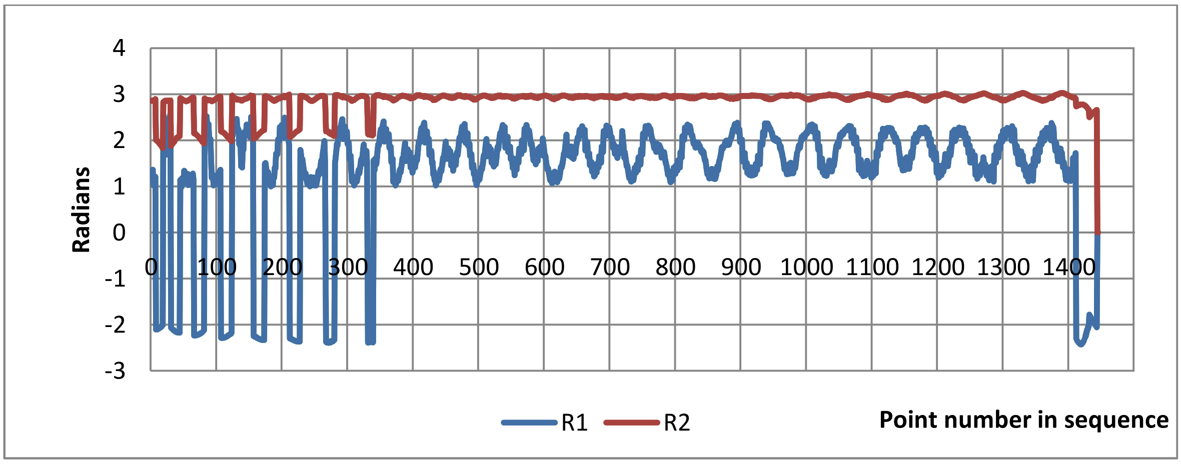
Though two solutions exist for each joint angle of the robot (e.g. so called lefty and righty), Graph 10 shows a single result based on a test to see if the R2 component is positive or negative i.e. clockwise or counter-clockwise as indicated by the sign calculated in the inverse kinematics.



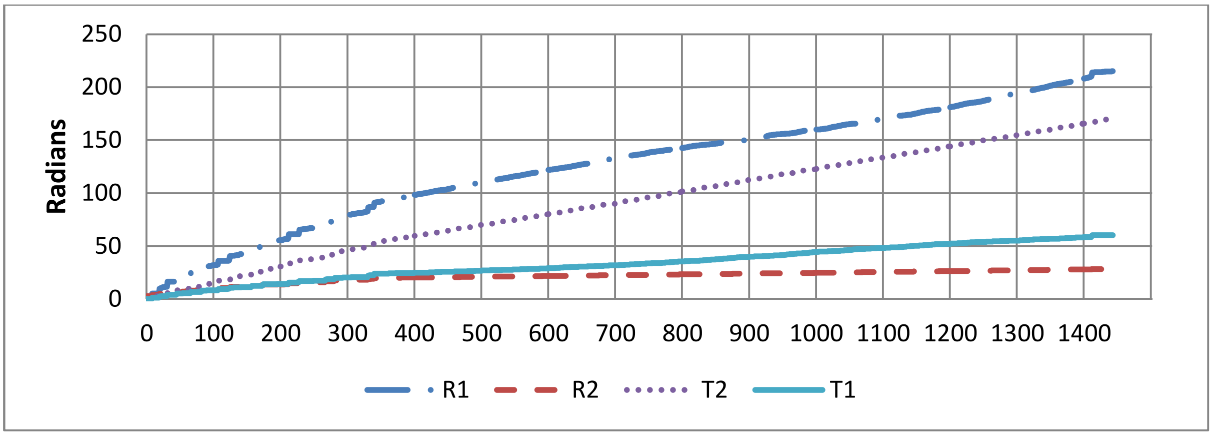
Graph 8 Deformed component tilt values (T1) while using the lamina slice path



Graph 9 Deformed component rotational angles (T2) for the lamina slice path



Graph 10 Deformed component joint angles R1 and R2 for lamina slice path

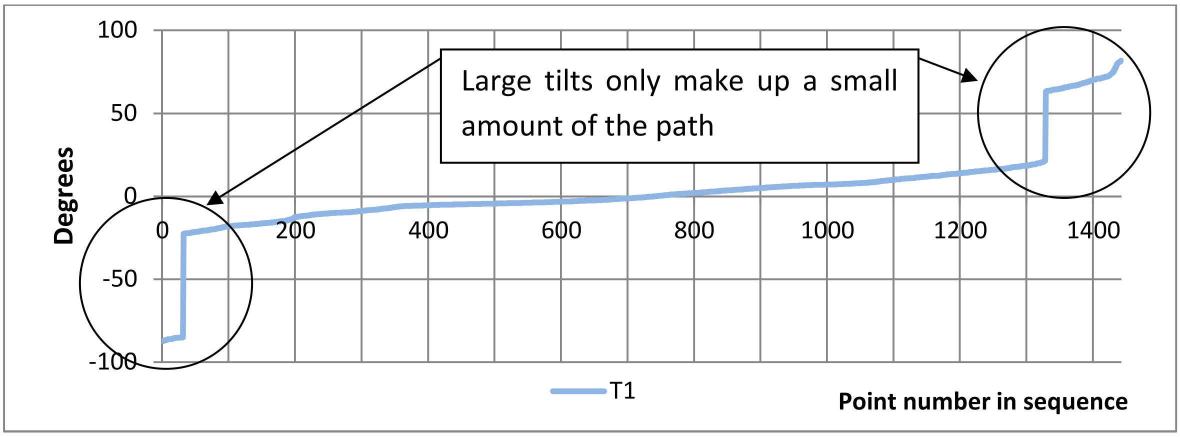


Graph 11 Cumulative joints travel R1, R2, T2 and T1 for deformed cylinder using lamina slice path

Graph 11 shows the cumulative travel for each of the joints in the system. The graph shows that each of the joints undergoes a steady increase, with the R1 and T2 (rotation) joint contributing slightly more to the total travel than the T1 (tilt) and R2 joints. The total travel for the lamina slice path is 326 radians and the max move length is 552 radians. The following sections investigate how alternative sequencing methods impact this result.

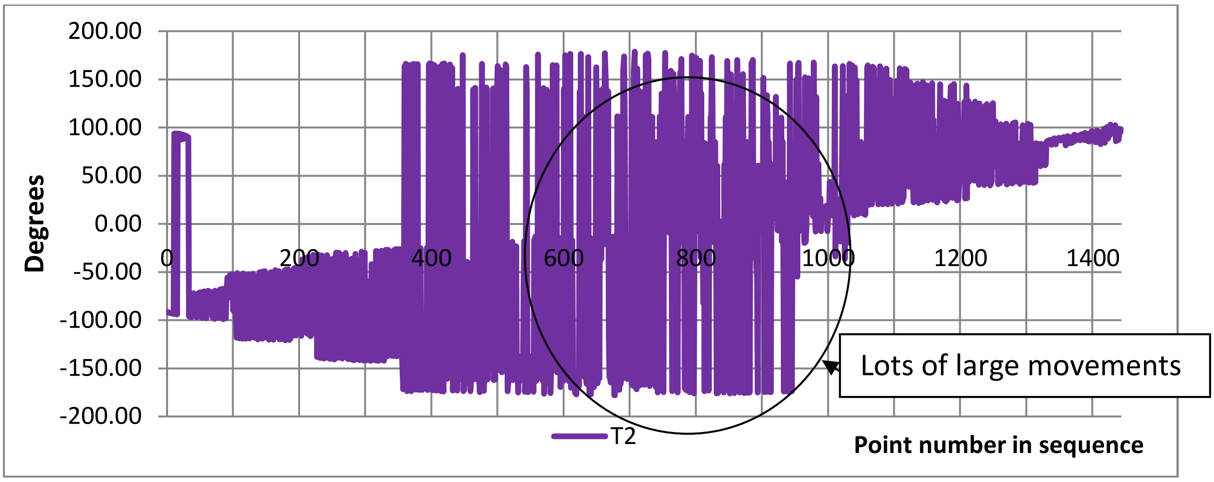
6.2.2.2 T1 Minimisation Path

The single joint path based on T1 (tilt) is very difficult to visualize as the path tends to travel over a large range of movements based on the increase of the T1 joint. When the points are organised based on T1 it is difficult to see the range of movements that the tilt table must perform to expose the complete component. However, it is possible to observe that the majority of the points occur within a much smaller range (i.e. -23 degrees through to +23 degrees) with only a couple of hundred points occurring outside this range (Graph 12).

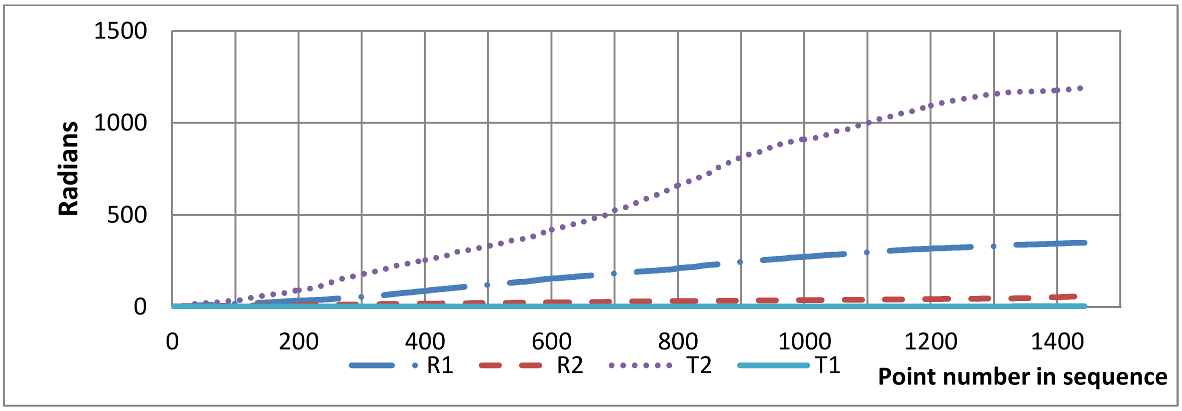


Graph 12 Deformed cylinder sequenced based on tilt angle (T1)

When the cumulative distance is calculated for the T1 path, the results suggest that though a significant reduction has occurred for Tilt, the robotic joints R1 and R2 have not changed greatly and the T2 rotational component has increased substantially. Graph 13, shows the rotational movement of the tilt table and records hundreds of large positional changes throughout the path. The total distance travelled by the system has increased to 1681 radians with the T2 component contributing 1195 radians; this can be seen in Graph 14.



Graph 13 T2 joint angles when sequenced using T1



Graph 14 Cumulative distance for each axis after T1 actual sequencing

6.2.2.3 T2 Minimisation Path

The organisation of the points based on the T1 joint has been seen to increase the overall travel. In the cylinder, T2 sequencing was seen to reduce the overall path length compared to the lamina path; with the more complex form of the deformed component, this section will see if this reduction is reproduced. The path generated while sequencing a single joint using T2 can be seen in Figure 6-13. Graph 15 shows the rotational values after they have been sequenced to minimise T2 travel from smallest to largest values. The mid-point in the rotation occurs approximately halfway through the points and the results range from -180 degrees through to +180 degrees.

Graph 16 shows how the angles R1 and R2 vary after single joint planning by T2. The graph shows that slightly more feathering occurs (i.e. small quick movements) along the path but it remains consistent. The total distance travelled (Graph 17) using the T2 path is 785 radians with R1 contributing the largest amount at 450 radians, the T1 and R2 joint still remain consistent at several hundred radians. The results show a significant jump occurring after approximately 1000 points and this is seen in the results for the R1 joint as the robot moves from the right solution to the left solution on multiple occurrences. With further investigation, it was found that the programme only found a single solution at this point which was caused due to limits being placed on the calculation.

The T2 path can be seen in Figure 6-13, which shows that each facet on the model's surface would appear to have a path created processing all the points on it. Although the path is difficult to follow, this is caused by the visualisation programme (e.g., where the path self-intersects causes faceting issues on the path) it is possible to decipher some regular patterns.

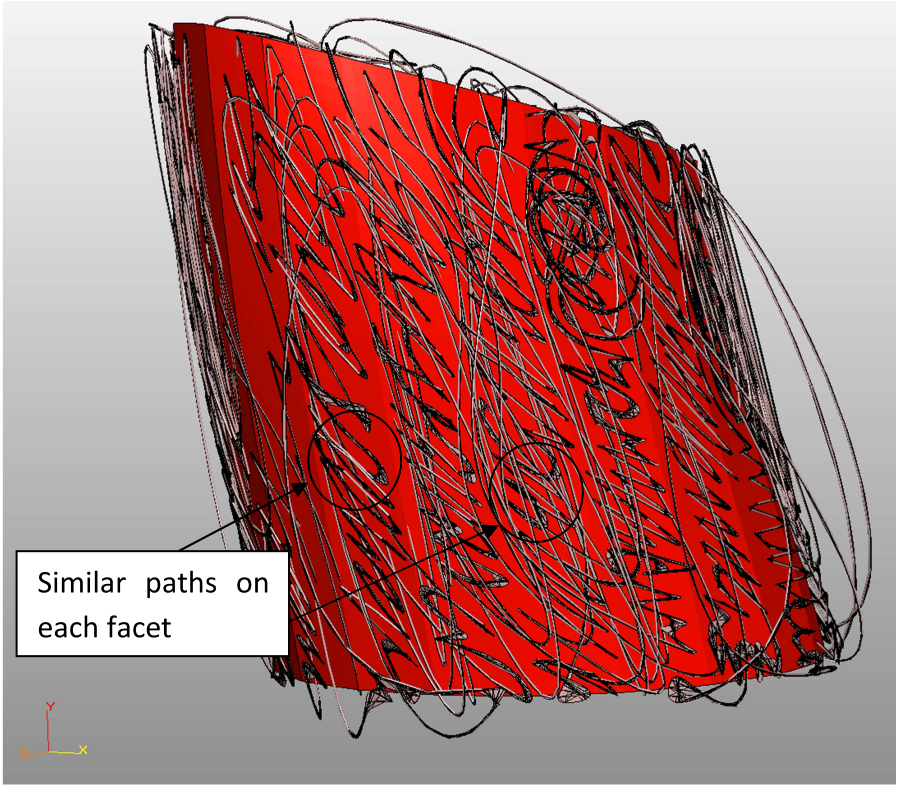
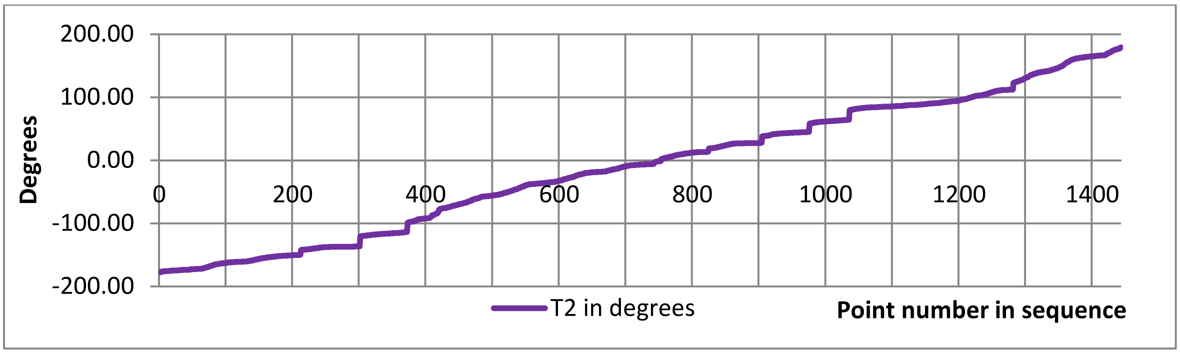
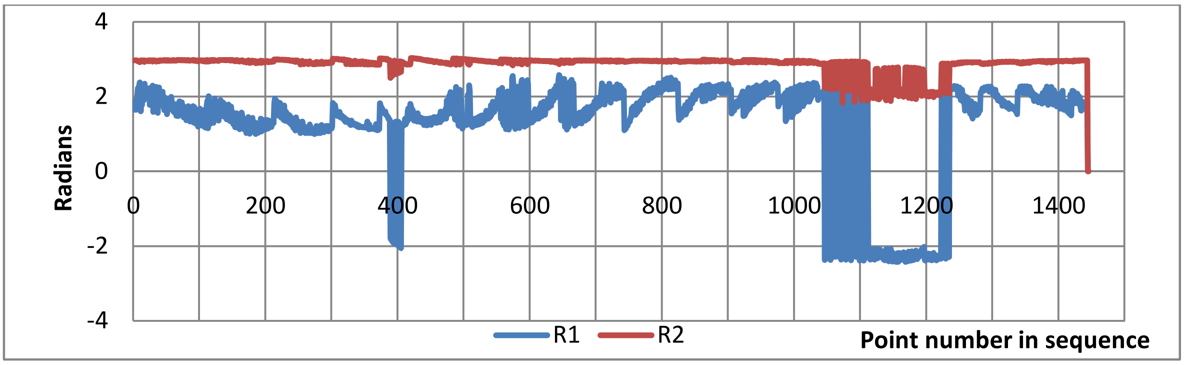


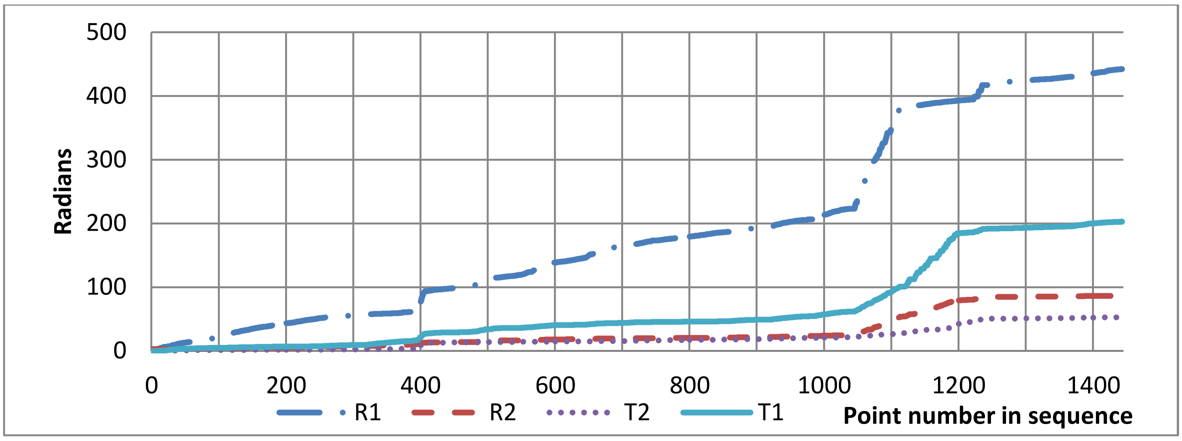
Figure 6-13 Deformed cylinder single joint path for T2



Graph 15 T2 joint angles for the deformed cylinder single joint T2 path



Graph 16 Deformed cylinder R1 and R2 joint angles for T2 path



Graph 17 T2 sequencing showing cumulative distance travelled by each axis

6.2.2.5 *Cartesian Closest Point Path*

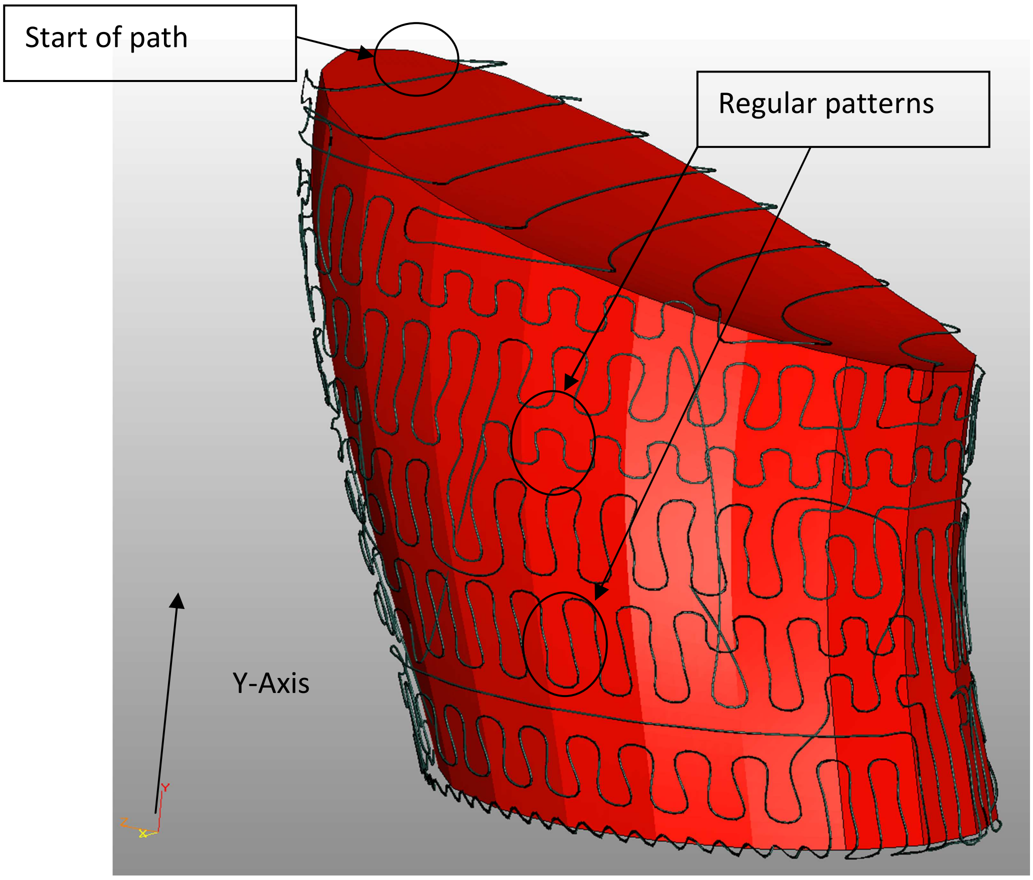


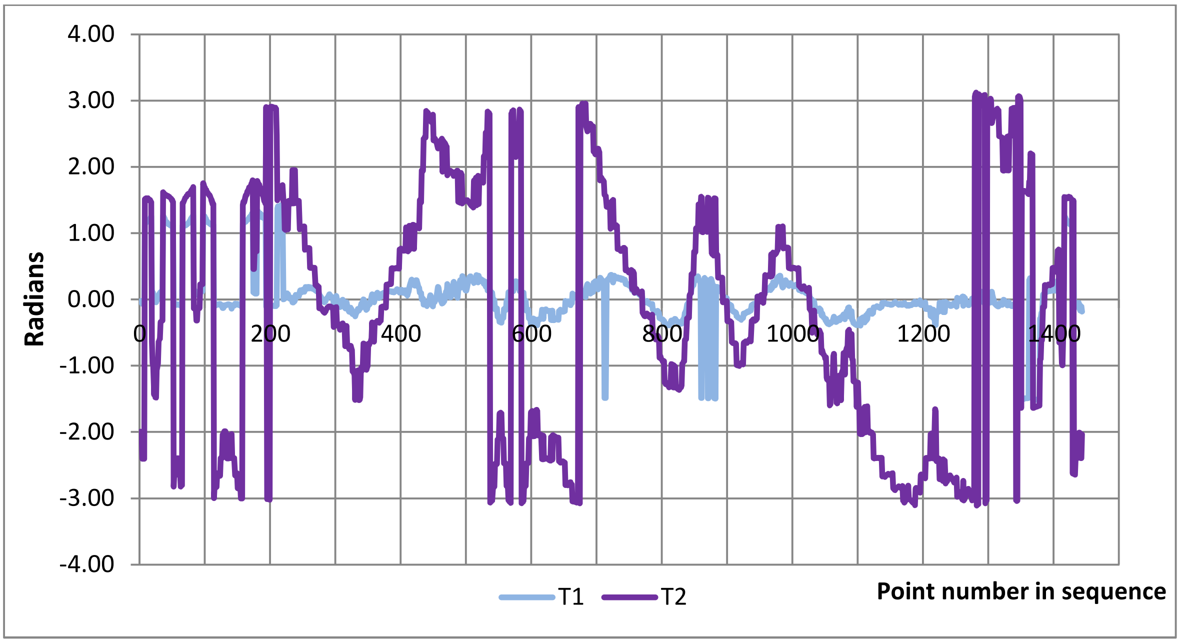
Figure 6-14 Deformed component with Cartesian closest point path

Figure 6-14 shows the surface path for the Cartesian closest point approach. The path appears to be relatively well structured, processing lamina groups of points then proceeding to the next layer. The inclusion of the top surface of the component would yield some large positional changes for the robotic end effector as these are being painted, therefore large configuration changes would be required on several occasions. Despite the large positional changes for the end effector, the path appears to be relatively easy to follow, with few overlapping areas.

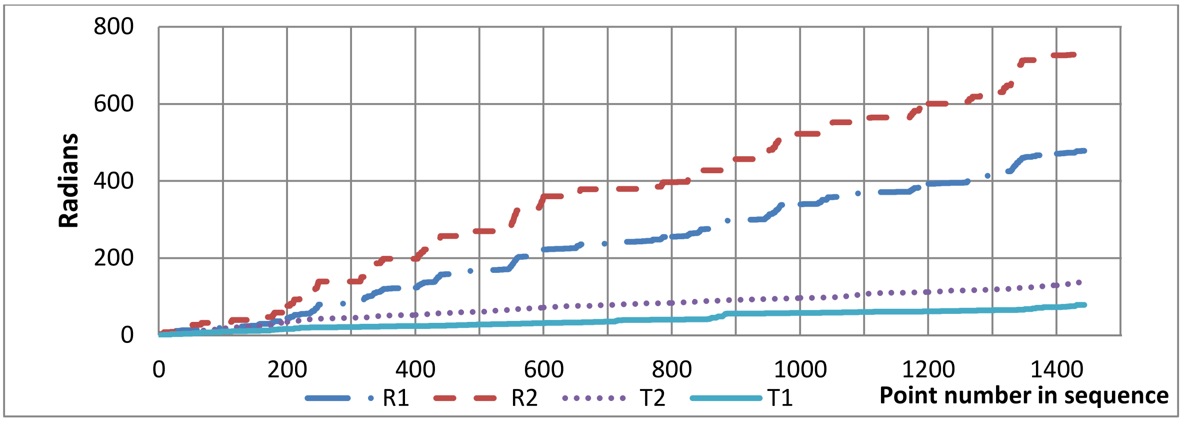
Graph 18 shows the effect that using the Cartesian closest point path planning has on T1 and T2 values. The path would appear to have significant movements over the initial 200 points, after which it appears to settle down and perform operations over small batches of points at each rotational position. The path appears to be slightly more organised than expected with a couple of large positional changes resulting in big changes of the tilt table's orientation.

The Cartesian algorithm produced a clear path in Cartesian space. However, the effect that it has on the configuration travel can be seen in Graph 19, where the robotic joints R1 and

R2 travel through substantially more radians than in lamina approaches. Though the path appears to be much shorter from its clear display, the results suggest that the total movement required by the joints increases by almost double the lamina approach to 1440 radians. This large increase can be associated with the several large end effector repositions that the Cartesian path requires. The max move (886 radians) is slightly more than half the total travel which would suggest that, while changing configuration, the amount of movement of joints other than the dominant one is increasing.



Graph 18 T1 and T2 joint angles for deformed cylinder after Cartesian closest point algorithm



Graph 19 Distance travelled by each joint for deformed cylinder after Cartesian closest point algorithm

6.2.2.6 Configuration Space Path

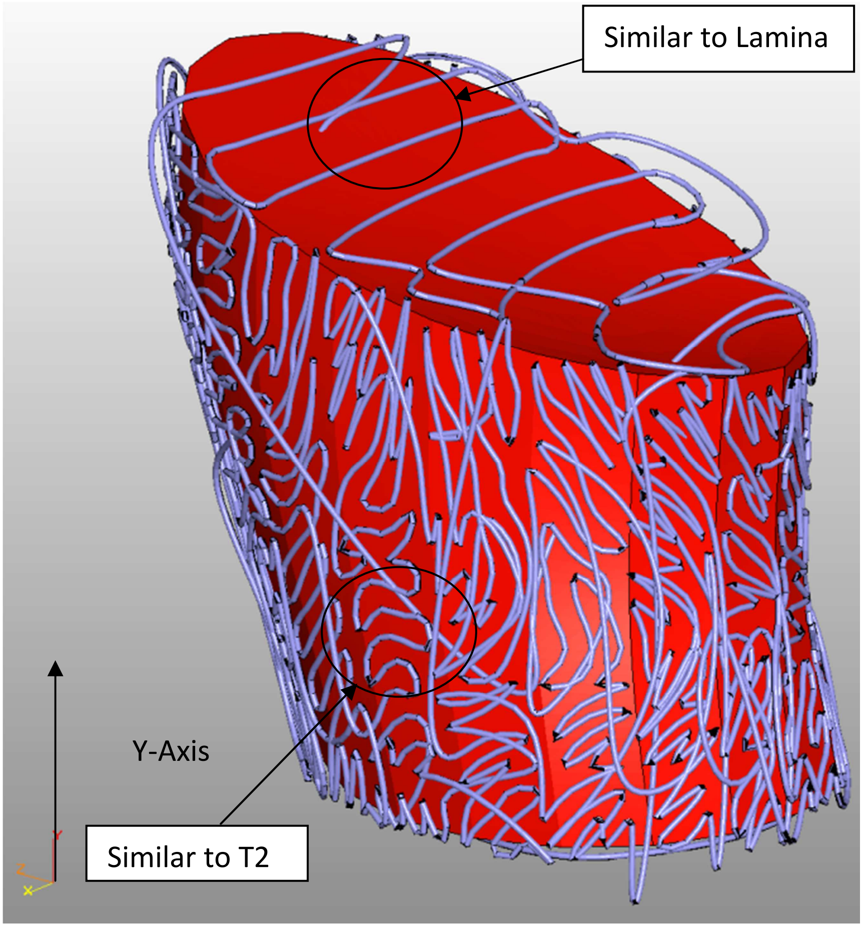
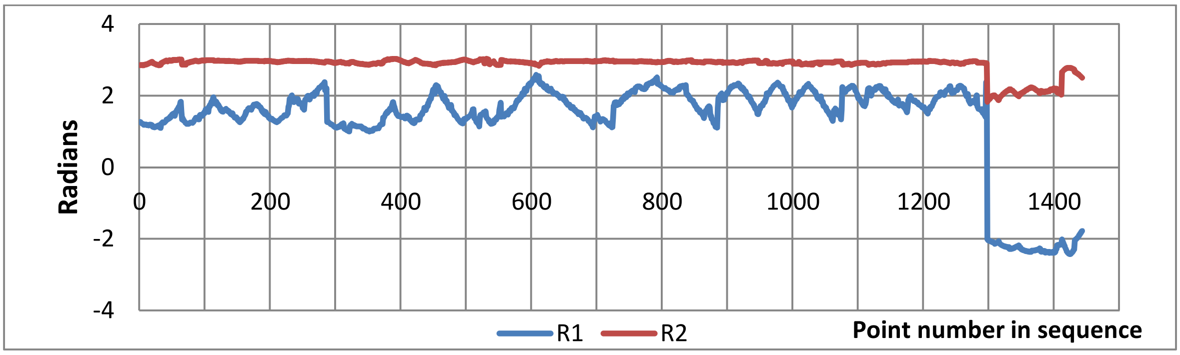


Figure 6-15 Configuration space path for deformed cylinder

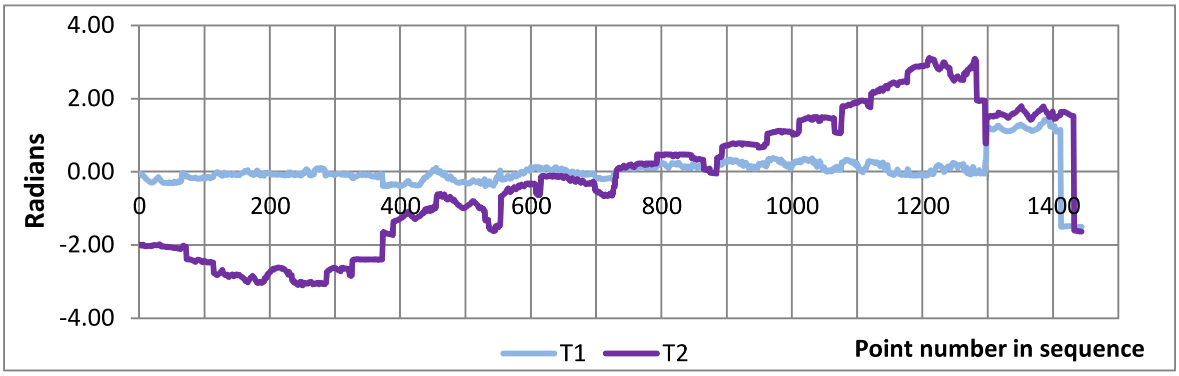
The configuration space closest point path can be seen in Figure 6-15 where elements of the different paths can be seen. Here the top face of the model is being handled in the same way as the lamina slice path, and the sides of the component are painted in a similar manner to the T2 path where individual facets appear to be painted together. The path can be visually followed over most of the component, which is unexpected with the more complex geometry as the path is derived based on the robot's joints rather than a human perspective.

Graph 20 and Graph 21 show the joint angles for the configuration space closest point path. The results show much cleaner lines with very little feathering occurring. As the algorithm is based on the closest configuration, it can be seen that very few large changes occur between points. The exception occurs in the R1 joint where the swap from positive solution to negative solution is seen around the 1300 point. The results show that no single joint has a linear profile.

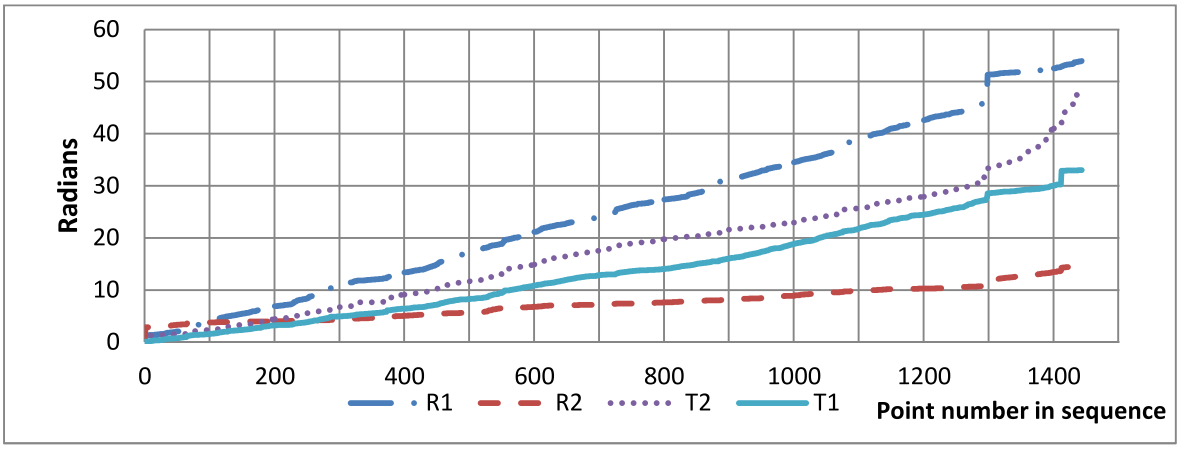
Graph 22 shows the total distance travelled by each joint where the R1 joint produces the greatest contribution (54 radians) to the overall total distance which is 150 radians; the max move for the path is 74 radians.



Graph 20 R1 and R2 joint angles after configuration space closest point sequencing



Graph 21 T1 and T2 joint angles after configuration space closest point sequencing



Graph 22 Total distance travelled by each joint after configuration space closest point sequencing

6.2.3 Eggcup Component

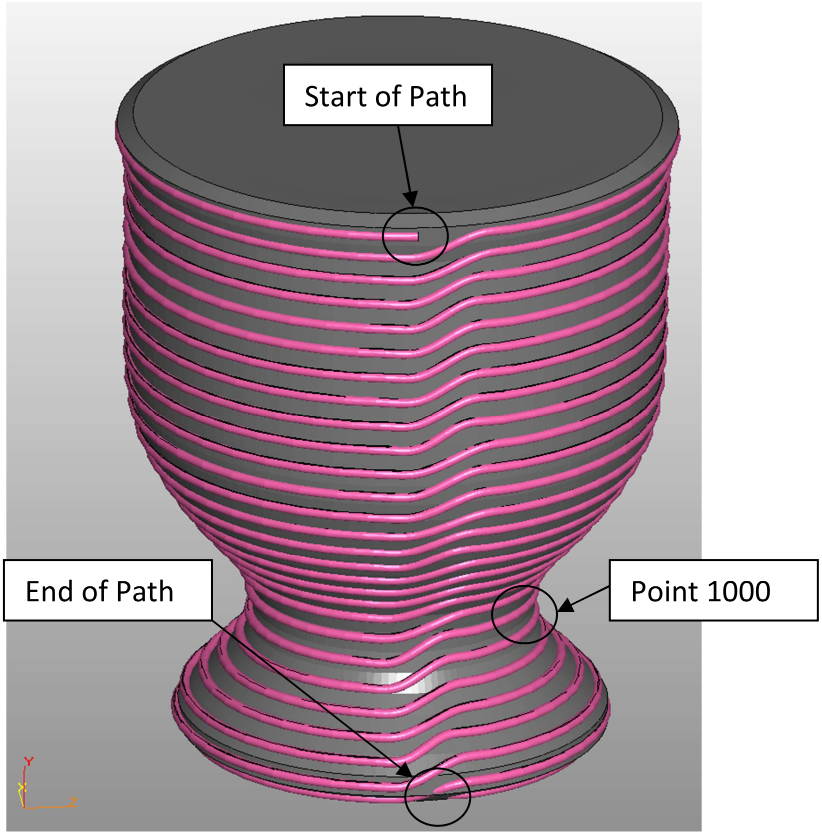


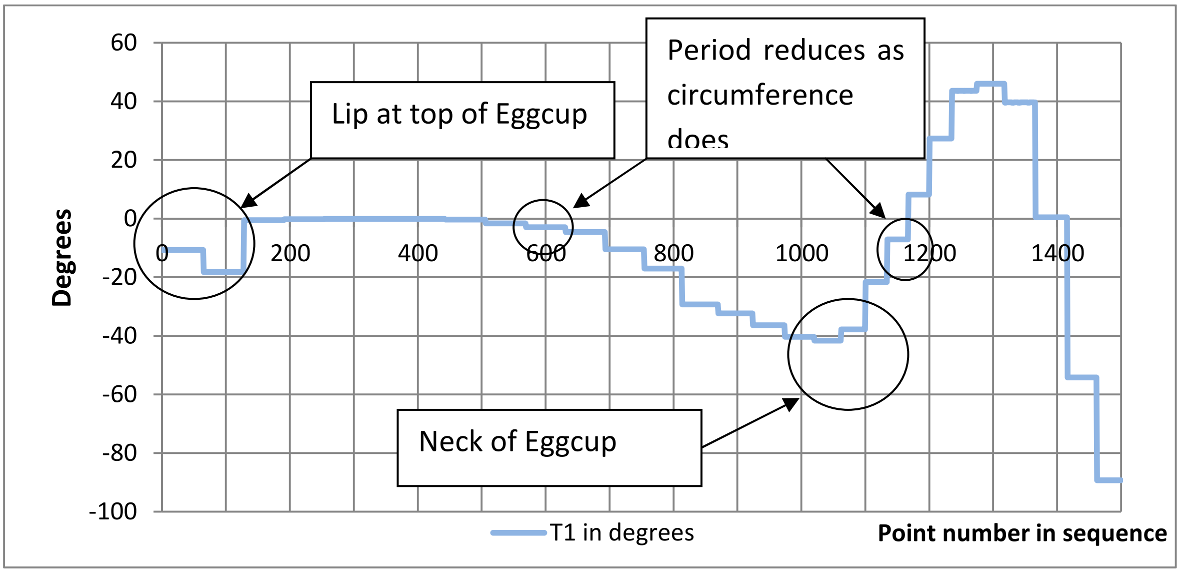
Figure 6-16 Eggcup with lamina slice path

The eggcup point set was created from a scan of a physical, ceramic model approximately 5cm x 3cm x 3cm (all the components used during this work measured less than 10cm in any direction). This provided a digital model that accurately matched a physical object suitable for experiments. The scan data is comprised of several thousand points that were meshed with a 2D image mapped on to them. Unlike the previous synthetic models, the normals for the model are calculated directly from the mesh by using a vector dot product. As with the deformed component, several points would be difficult to paint as the angle of tilt is greater than that which the table can provide. For this reason, the points on the top and bottom surface have been discarded.

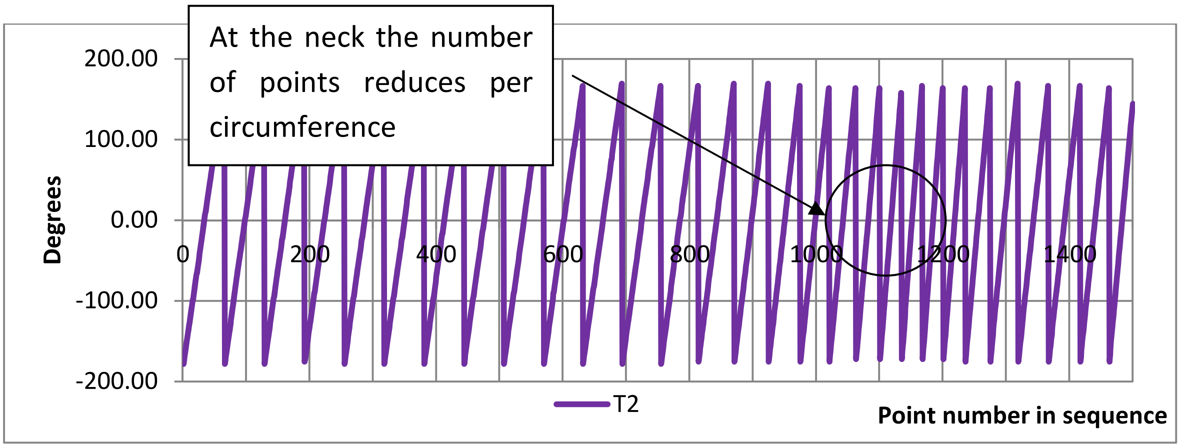
6.2.3.1 Laminar Slice Path

Figure 6-16 shows the painting path on the surface of the model within our workspace. The curved profile for the eggcup provides an interesting challenge for the system. For example, although the path may move down for each slice this may not be the case for the robotic end effector as the curved profile may require the robot to remain stationary while the table tilts.

Graph 23 shows the angles of tilt for the eggcup component with an initial tilt representing the small lip at the top of the component. The flat area on the graph is caused by the relatively straight top sides of the component. The slow increase in gradient represents the curved surface bending in towards the neck of the eggcup which occurs when the tilt is at zero again. At that point, the cup must tilt in the opposite direction to accommodate the slope culminating in the base ring where the eggcup tilts in the opposite direction. As the component is circular, the T2 joint angle oscillates between ±180 degrees. It should be noted that the period of the wave reduces as the circumference of the model decreases at the neck then increases as the circumference increases; Graph 24.



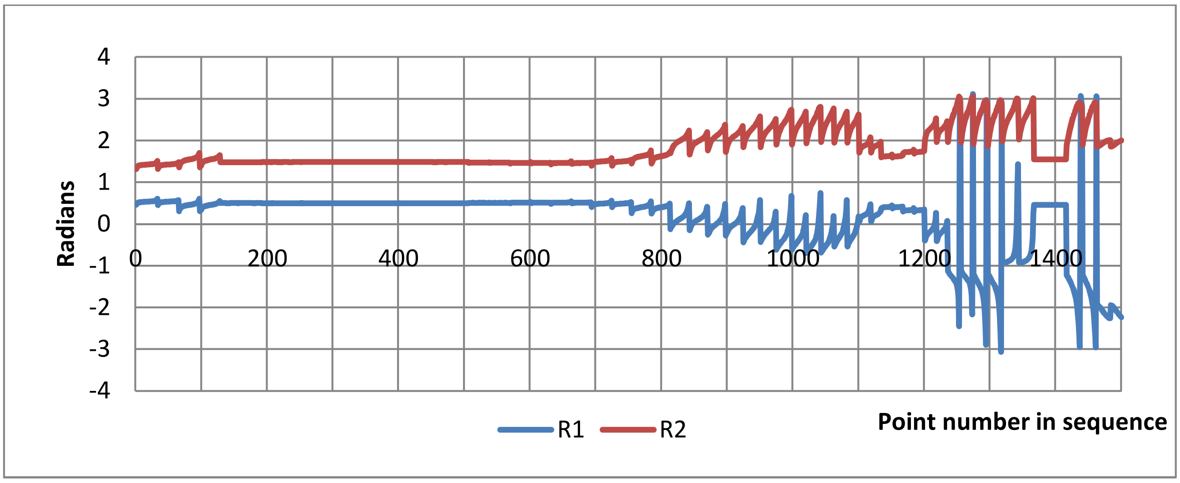
Graph 23 Eggcup model T1 angles for lamina slice path



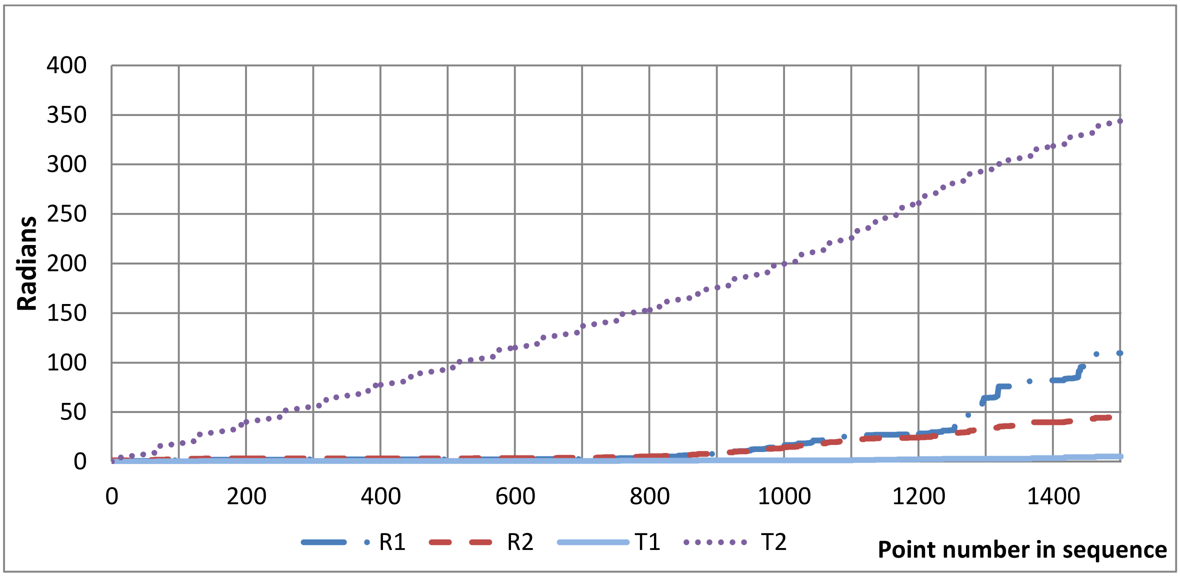
Graph 24 Eggcup model T2 values from lamina path

In a similar fashion to the cylindrical component, the robotic joints R1 and R2 maintain a limited range of movements for the regular curved sides (i.e. between 140 and 700 points). When the area around the neck of the eggcup is exposed, the robotic joints R1 and R2 are required to move more (Graph 25), at the smallest circumference of the neck (i.e. located between 1150 and 1200) the movement returns to a limited range.

Graph 26 shows the distance that each axis travels when the lamina path is used. The results suggest that over the first 700 points the movement that the robot is required to perform is limited, but when the neck of the component is being processed, the robot is required to perform more repositioning. The T2 component can be seen to be the biggest contributor to the overall travel contributing 343, out of the total 503, radians that the system moves. The tilt contributes a very small amount to the total movement in the region of 6 radians. The max move length for the lamina approach is 403 radians.



Graph 25 Robotic joints R1 and R2 for eggcup model using lamina path



Graph 26 Total distance travelled by each axis in the lamina slice approach to the eggcup component

The eggcup component provides '*real*' data for testing, with anomalous points present due to meshing errors caused by defects in the scan data. The data will be analysed by first looking at a path generated by the Cartesian closest point and finally the proposed configuration space path.

6.2.3.3 Cartesian Closest Point Sequencing

Figure 6-17 shows the path when the points are sequenced using the Cartesian closest point path. It is interesting to observe that, in areas where the rate of the gradient change is high, the path follows vertical lines (i.e. down the y-axis), whereas areas where the surface gradient is low, horizontal lines are produced (similar to the lamina approach). The path can be seen in full in Figure 6-17 where two distinct areas can be seen to be similar to those created using the lamina approach; the area at the top of the component (where the sides are perpendicular to the laser) and the area around the neck of the eggcup (where again the normals are horizontal).

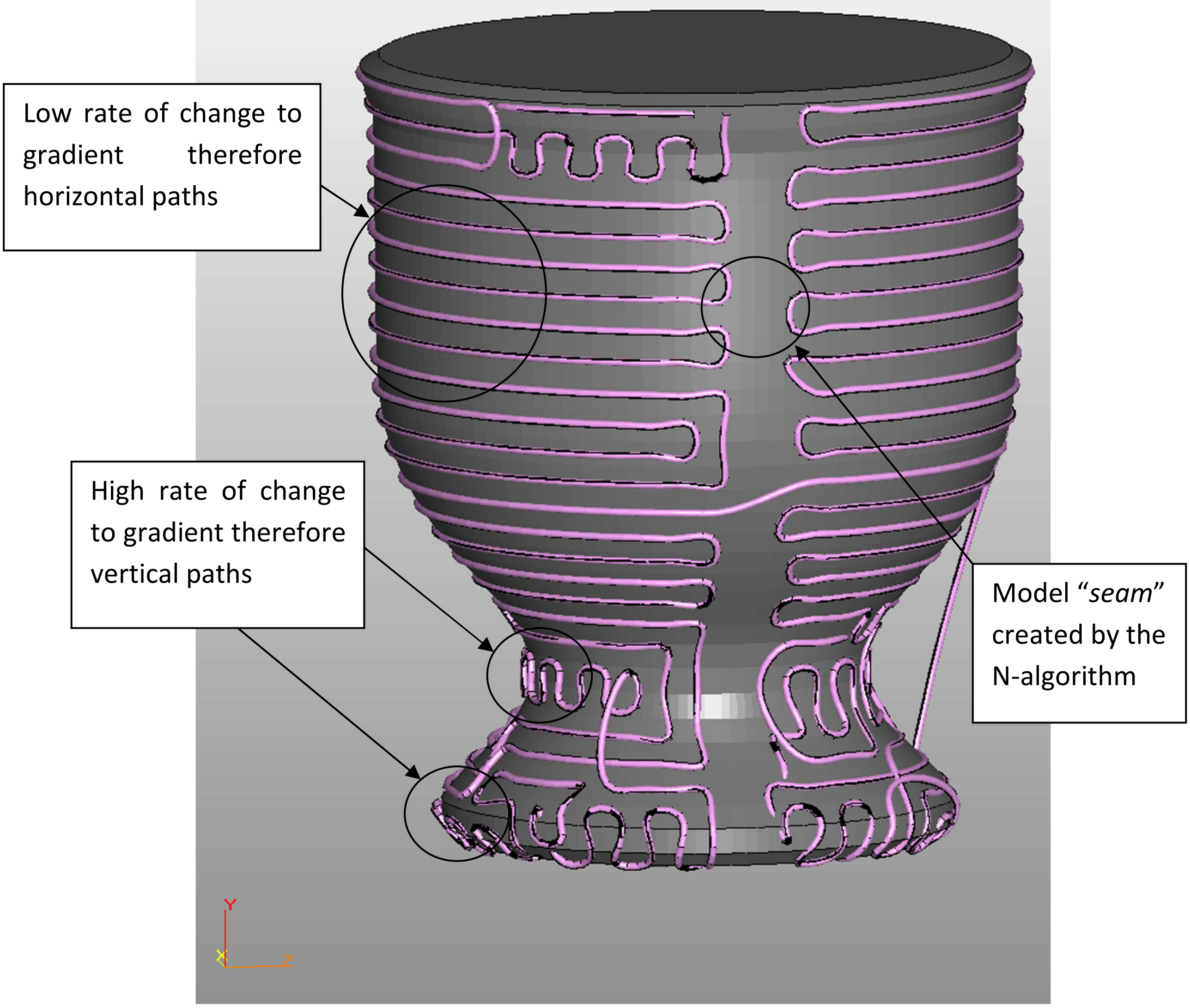

Low rate of change to gradient therefore horizontal paths
High rate of change to gradient therefore vertical paths
Model "*seam*" created by the N-algorithm
Y
X
Z

Figure 6-17 Cartesian closest point path for eggcup model

These results can be understood when one considers Figure 6-18 which illustrates why points on the surface with high changes in surface gradient are closer than points on the same x-z slice. The points are extracted to form the data set by the N-algorithm based on an approach that moves across the surface of the model therefore from one x-z slice to another the distance changes based on the surface curvature. However, the Cartesian closest point algorithm works without considering the model's surface (i.e. as the bird flies between points) therefore points with a greater rate of change in the x-y plane than the x-z plane will form a vertical pattern whereas when the opposite condition is found a horizontal (circumferential) pattern will be observed (similar to the lamina path).

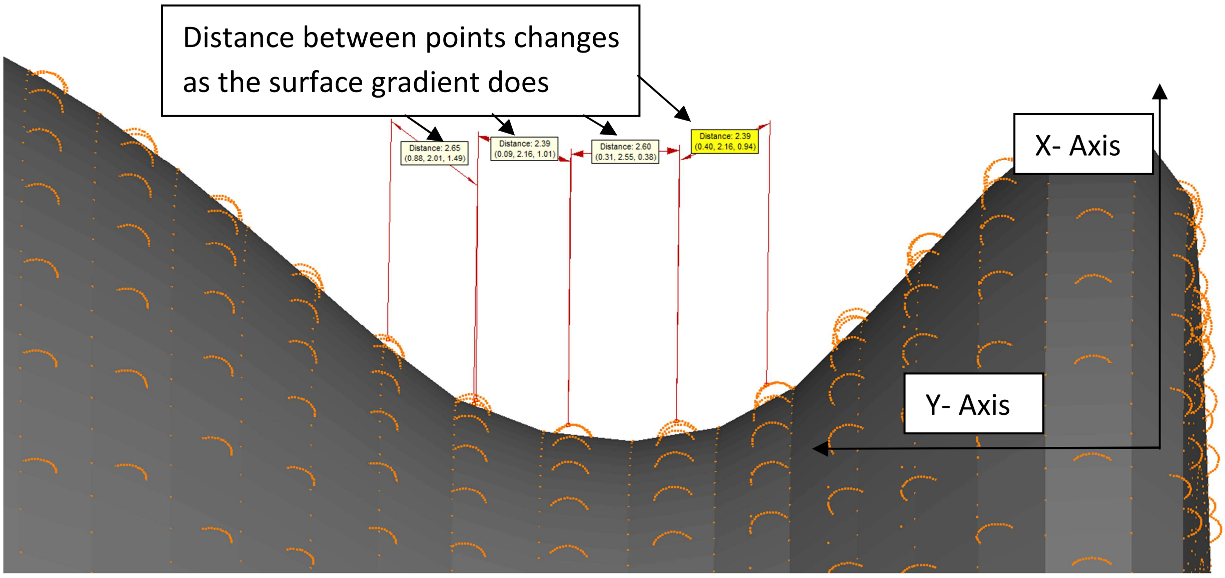
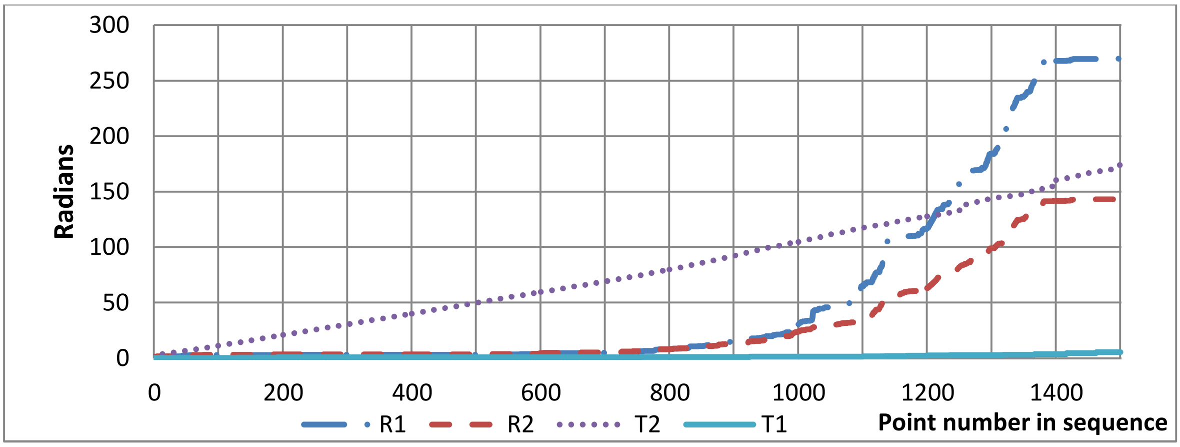

Distance between points changes as the surface gradient does
Distance: 2.65 (0.88, 2.01, 1.49)
Distance: 2.39 (0.09, 2.16, 1.01)
Distance: 2.60 (0.31, 2.55, 0.38)
Distance: 2.39 (0.40, 2.16, 0.94)
X- Axis
Y- Axis

Figure 6-18 Surface curvature affects distance between points

The total joint travel for each axis can be seen in Graph 27, where it can be seen that a consistent rotation T2 is observed throughout the path. The first 600 points appear to have relatively small amounts of movement. A similarity can be seen in the profile that is obtained for the R1 and R2 axes, with a steeper gradient appearing for R1 than the other joints. The final 100 points of the path, where the neck of the eggcup is being processed, appear to reduce the movement again for several joints, similarly to the first 600 points which were at the top of the eggcup where the lamina slicing pattern has emerged.



Graph 27 Cumulative joint travel for eggcup model using the Cartesian closest point path

6.2.3.4 *Configuration Space Closest Point Path*

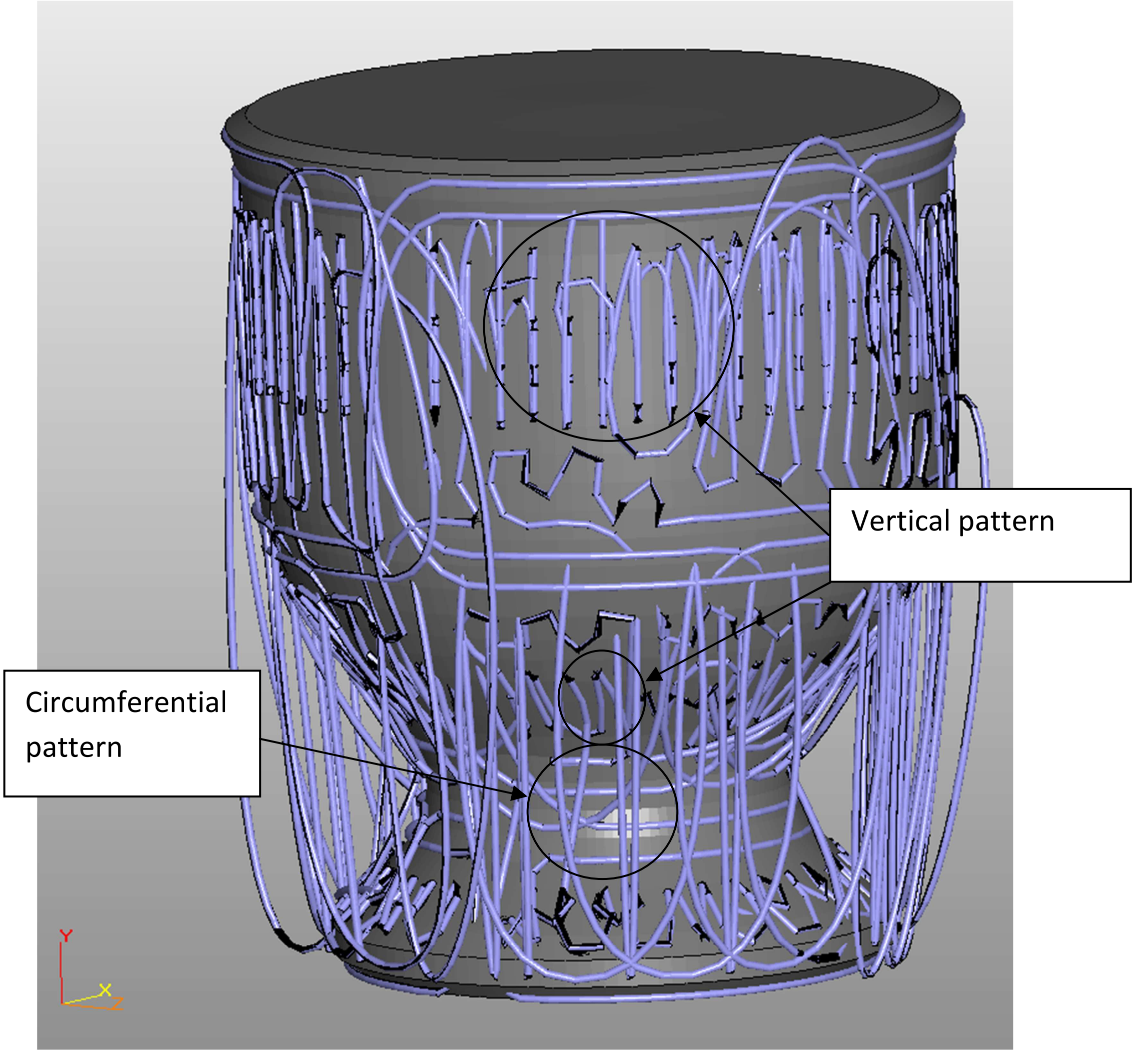
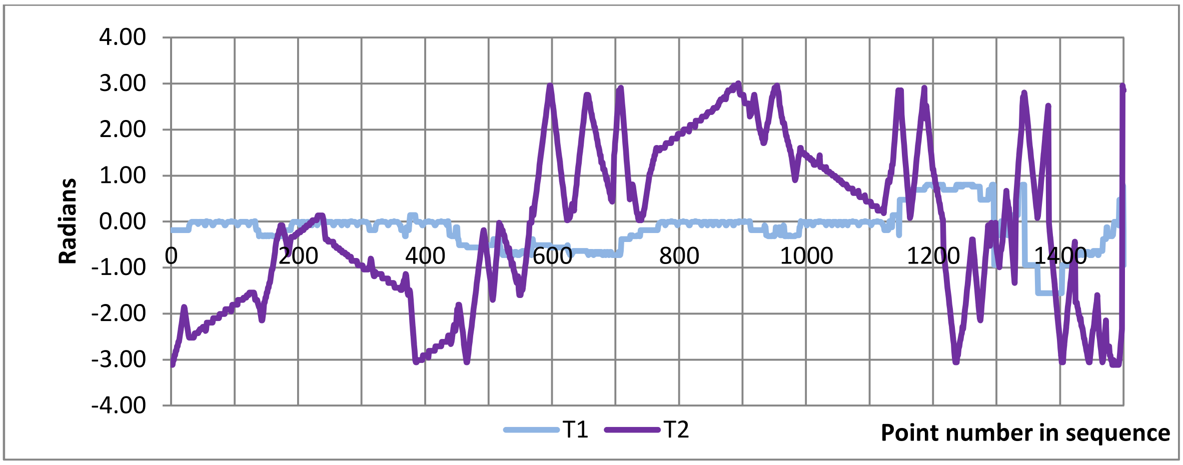


Figure 6-19 Configuration space closest point path for eggcup

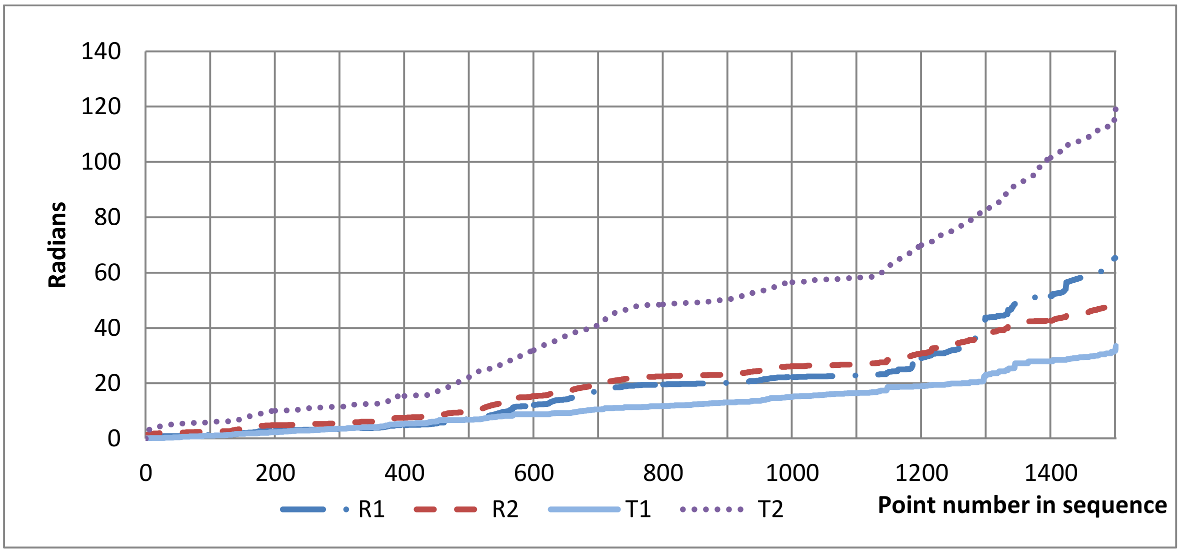
The path generated by the configuration space closest point path can be seen in Figure 6-19 where distinct regions of vertical and horizontal patterns can be observed. At the top of the model, it is possible to see the path painting points in vertical columns similar to the cylinder. It is interesting to observe that one point near the base of the cylinder must have a similar configuration, as the path appears to paint a single point below the neck. It is also interesting to observe that the reverse to the Cartesian path appears to occur; at areas around the neck, the path appears to become circumferential and on the areas with lower curvature, a vertical pattern appears to be used.

Graph 28 shows the joint angles for the configuration space approach. It is possible to see that the variations that appear for single joint optimizing are removed however, small adjustments can be seen to make sure the nearest configuration is used.

The joint's travel can be seen in Graph 29 where the T2 axis remains the greatest contributor to the overall travel. The total distance is 267 radians with T2 contributing 118 radians. When these results are compared to the lamina approach (where the total travel was 503 radians), it can be seen that the path created by the configuration space algorithm is only slightly more than half of that created by the lamina approach. This is again reflected in the max move value for the path where it reduces from 403 radians to 157 radians.



Graph 28 T1 and T2 joint angles for configuration space path on the eggcup



Graph 29 Configuration space path total distance travelled by each joint for the eggcup

6.2.4 Penguin Component

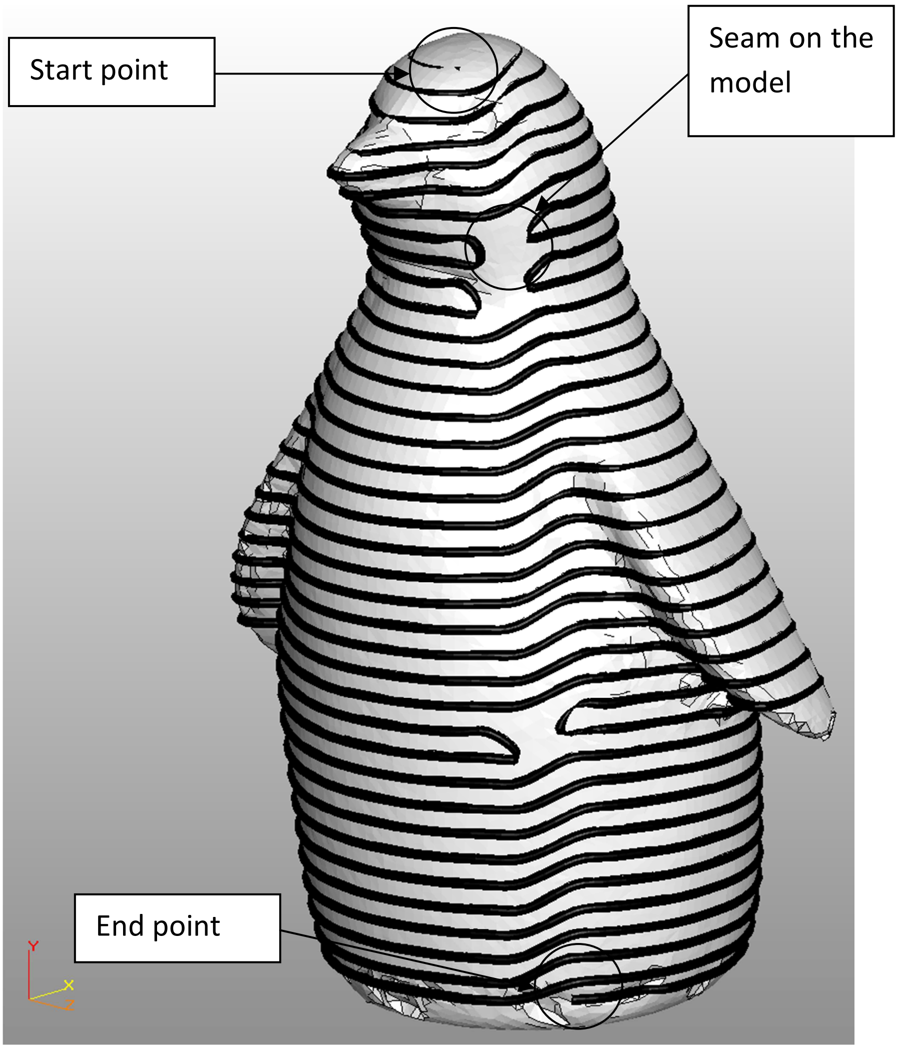


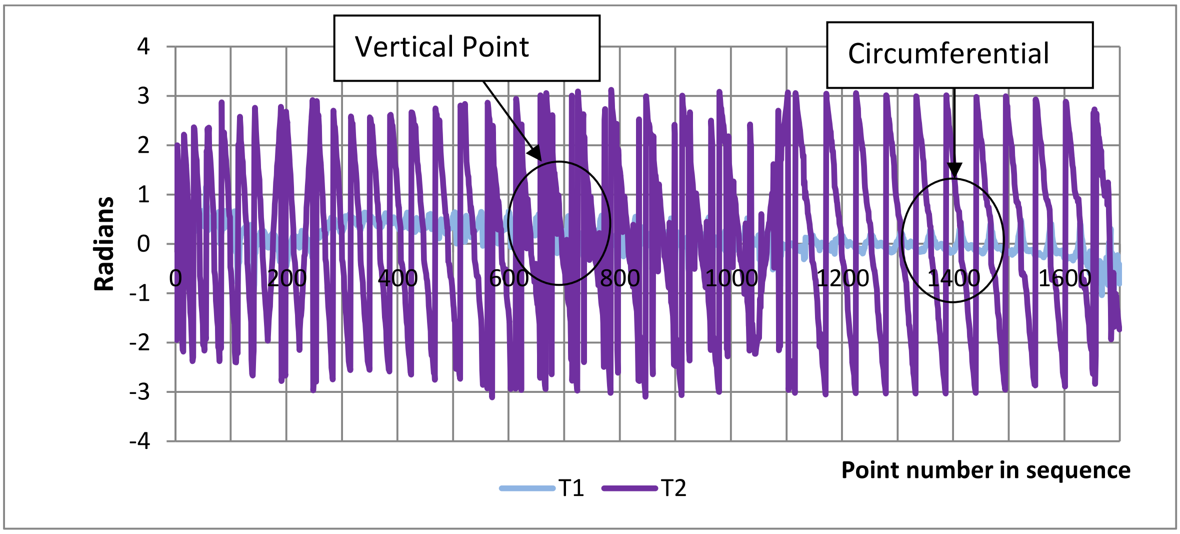
Figure 6-20 Penguin model with lamina path

The penguin is a significantly more complex component than the previous tests as it has several sharp angular changes and intricate details. The N-algorithm generated 1702 points across the surface of the model. Although several points may be outside the physical capabilities of the system (i.e. workspace) to paint, they will be included in this analysis.

6.2.4.1 Lamina Slice Path

Figure 6-20 shows the surface path for the penguin. It can be seen that the variation in surface normal associated with the points on the path is significantly greater than previous models and the end effector would create a "*dome*" around the penguin. The uniform layers for the end effector of the robot that were present in other components such as the eggcup would not be created for the penguin.

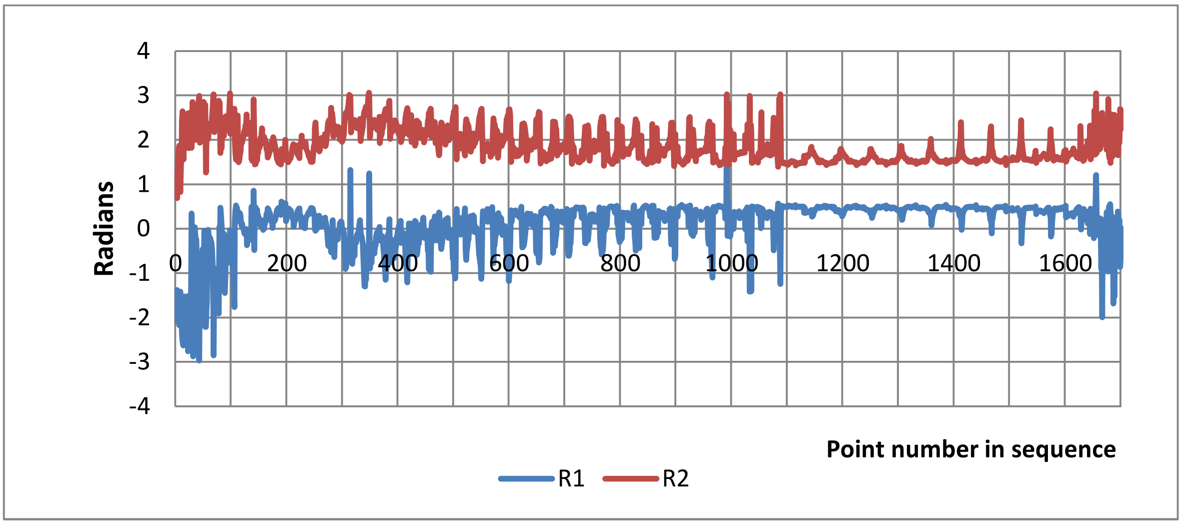
Graph 30 shows the T1 and T2 values for the penguin from which it can be seen that the tilt occurs mostly between 1 and -1 radians (approximately ±57 degrees). The T2 component again operates between ±180 degrees, but is significantly more varied throughout the process and only forms a regular pattern as it approaches the base of the component where the shape becomes more regular. As the penguin has undulating features on each slice, it is difficult to gauge where each layer ends. However, for areas where more regular paths can be seen, the regularity of the slices appears to vary significantly this can be seen over the first 100 points and then later after 1100 points.



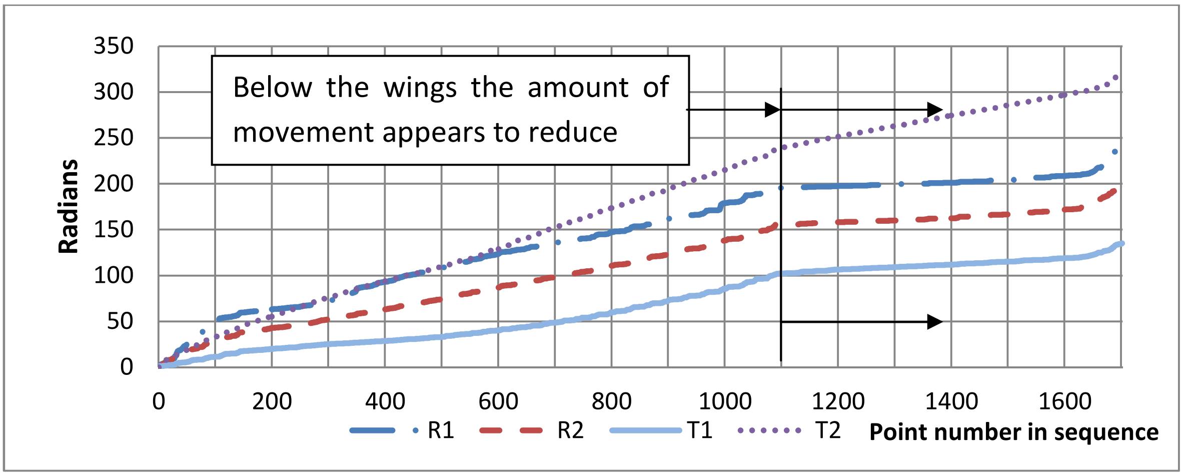
Graph 30 T1 and T2 joint angles for the penguin using lamina path

As with the other components, the robot movements are much less than the tilt table movements. This can be seen in Graph 31 where the range of movement remains fixed at approximately 2 radians. The results show that the start of the path (i.e. the crown of the penguin's head) requires a different position to the rest of the component. The first 300 points encapsulate the head of the penguin where larger tilts are required. After 600 points, the results show that the robot remains in a confined work area with small changes. Once 1100 points have been exposed, the robot's movement reduces further, which would indicate that the surface normals are largely horizontal with much of the movement being provided by the rotation of the tilt table. The area situated at 1650 points produces results that are slightly more erratic and is indicative of the tail of the penguin and the damaged area round the base of the model. The scanning of the components was performed on a rotary table to provide the most accurate results with the equipment available. This allowed the upper area of the penguin to be fully visible and a smooth scanning path could be obtained over the entire surface. At the base of the component, where an acute angle is formed between the component and the table, the beam reflected off the table creating anomalous surfaces. The component was healed using a mesh-healing algorithm however; small errors (i.e. damage) near the base of the component were still visible on the computer model.

Graph 32 shows the distance travelled (i.e. cumulative rotation) in configuration space for each of the joints. The initial 200 points show that the robot appears to be repositioning frequently with the T2 joint also moving. After the initial movement by the robot, the three axes R1, R2 and T1 seem to increase in parallel. Only the T2 axis increases at a greater rate. Once 1100 points have been exposed, this pattern continues but at a reduced rate. The final 50 points associated with the tail of the penguin increase the rate of movement for all axes. The total movement of all the joints using the lamina slice approach is 890 radians and the max move for the path is 442 radians.



Graph 31 R1 and R2 joint angles for penguin with lamina path



Graph 32 Total joint travel for the lamina path on the penguin

It can also be seen from the results that the path created by the lamina approach requires the robot to move a substantial distance due to the complex nature of the shape. The different path planning techniques have been used on simple geometries so far which have normally been based on a cylindrical profile with limited features. The penguin presents a new challenge as intricate detail on the head and around the wings will present a new trial to the different paths. The next section looks at the use of the Cartesian closest point path which should create a path that can be followed visually however; it will be interesting to discover the effect it has on the total joint travel. The model will then be looked at using the configuration space path to discover if controlling the robot's joint movement can dramatically reduce the path.

6.2.4.4 Cartesian Closest Point Sequencing

The penguin model has a variety of different surfaces with varying amounts of both curvature and tilt, the Cartesian closest point path can be seen in Figure 6-21. The chest of the penguin is a relatively low curvature, low tilt surface, the Cartesian closest point path can be seen to adopt a linear horizontal pattern similar to the one created in the lamina path. This horizontal path ends once the path has crossed the chest, which is due to an area of higher curvature causing a vertical pattern to be used. The path returns to a horizontal pattern as it progresses down the body of the penguin before returning to a vertical pattern near the base over the damaged area of the mesh.

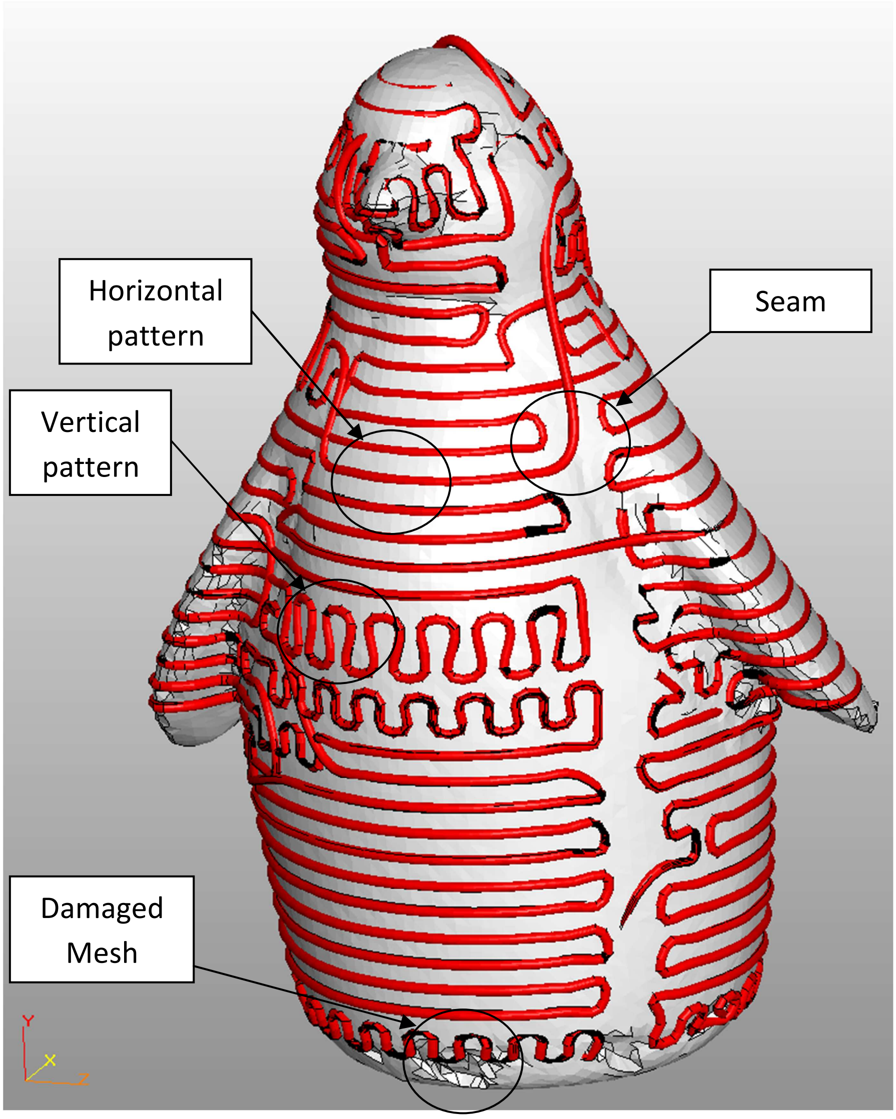
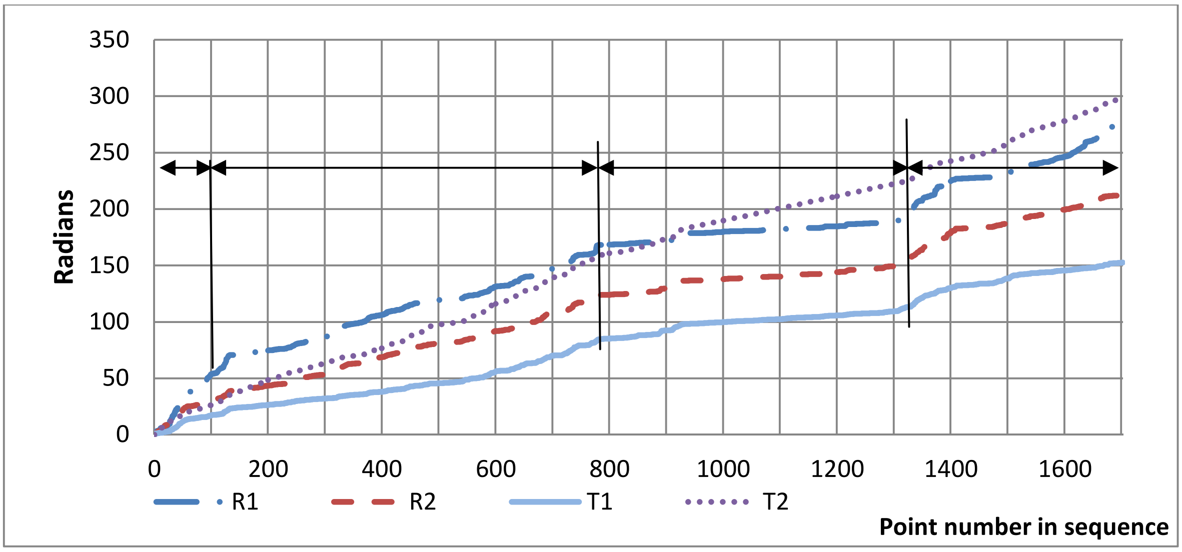


Figure 6-21 Cartesian closest point path on the penguin model

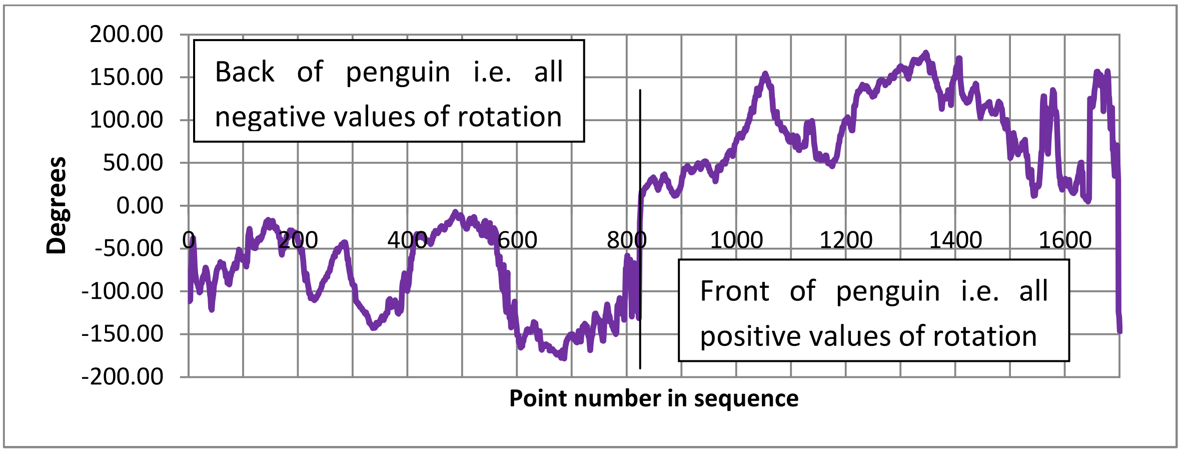
The graph plotting the total travel by each of the joints against point number shows that R1, R2 and T1 indicate similar rates. The only difference appears in the first 70 points where they each vary gradients, though the rest of the results appear to be an offset value from the other joints. The T2 joint travels the greatest distance (300 radians) and has a consistent gradient throughout much of the process indicating a very similar result to the lamina approach. The Cartesian approach appears to break the component into four separate sections where the joint angles move in a consistent manner asindicated on the graph. This is again very similar to the lamina approach however, the period for each of the quadrants changes from the lamina to the Cartesian approach.



Graph 33 Joint travel for Cartesian closest point path on the penguin showing similar sections

6.2.4.5 *Configuration Space Closest Point Sequencing*

The configuration space algorithm is specifically designed to handle components of a high complexity by looking at the robot joint data rather than the surface geometry. This approach ignores any preconceived paths and creates a path which minimises the movement of the joints. The path appears to concentrate on "*exposing*" the negative rotation side first (i.e. from 0 degrees to -180 degrees). This can be seen in Graph 34 which also shows that the full range of rotational values is used. Though the path appears to be rather erratic, it does not produce the feathering (i.e. the short quick changes in position) that has been seen on other components and which increases the movement greatly.



Graph 34 Penguin T2 solutions for configuration space sequencing

The end effector path for the penguin can be seen in Figure 6-22. Although it is difficult to follow the path visually, it is possible to see that in certain areas the path creates the vertical linear lines that the T2 minimization approach used on previous models. The path appears to have a much broader range of movements than on other components. This is due to the more complex surface on which it is operating. It can also be seen that in certain areas, the path produces a large Cartesian movement which must be associated with a smaller configuration movement.

The joint travel for each of the four axis can be seen in Graph 35, which shows that the T2 axis moves through the greatest distance at approximately 135 radians, the R1 and R2 axes move through 88 and 78 radians respectively but follow a similar profile to the T2 axis. The T1 axis again follows a similar profile however; the reduction in the gradient means the result appears to be constantly increasing from 830 points. The system travels through a total distance of 370 radians, which is a significantly smaller amount than the 890 radians that the lamina approach produced. The max move value for the configuration space path has reduced to less than half that seen on the lamina path.

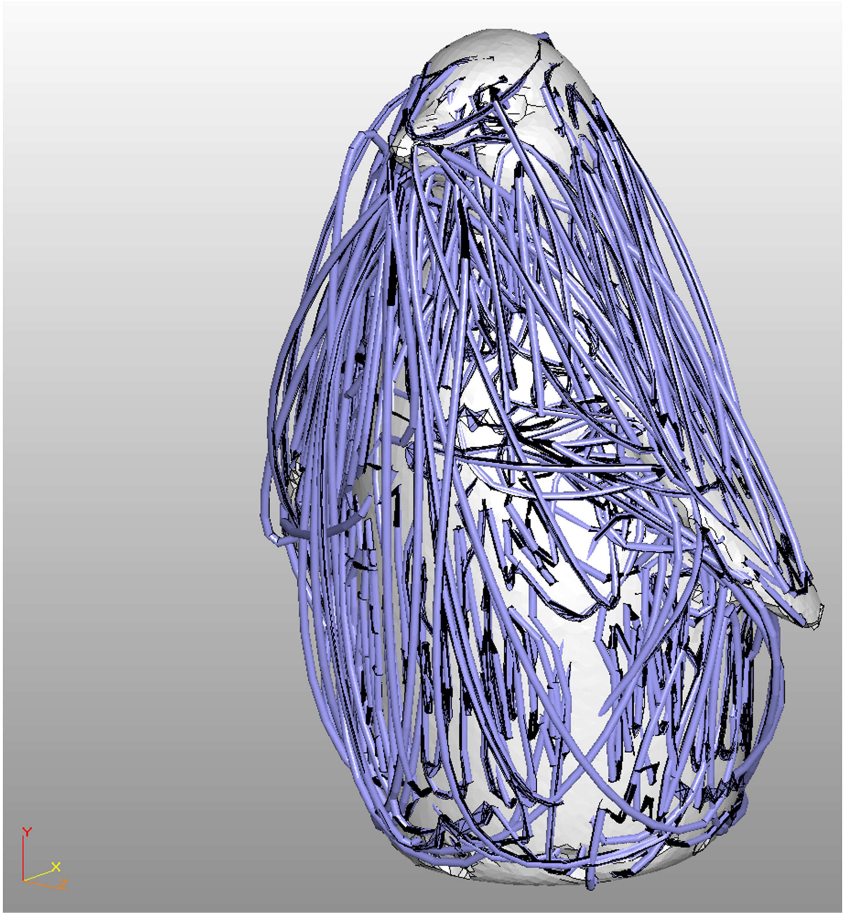
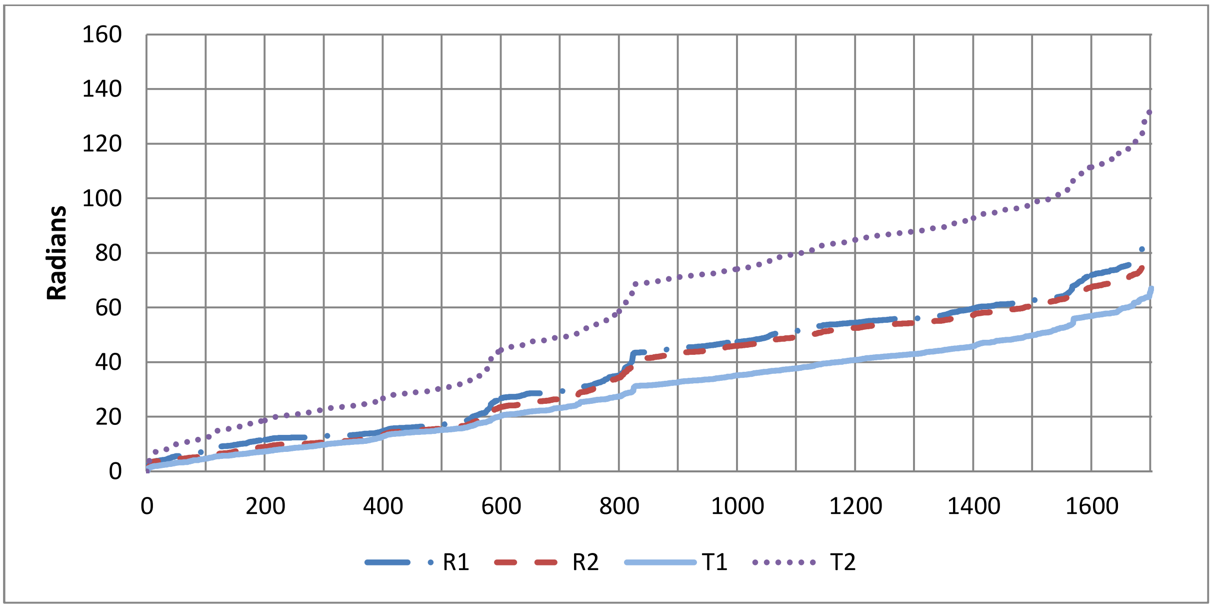


Figure 6-22 Configuration space closest point path for penguin model

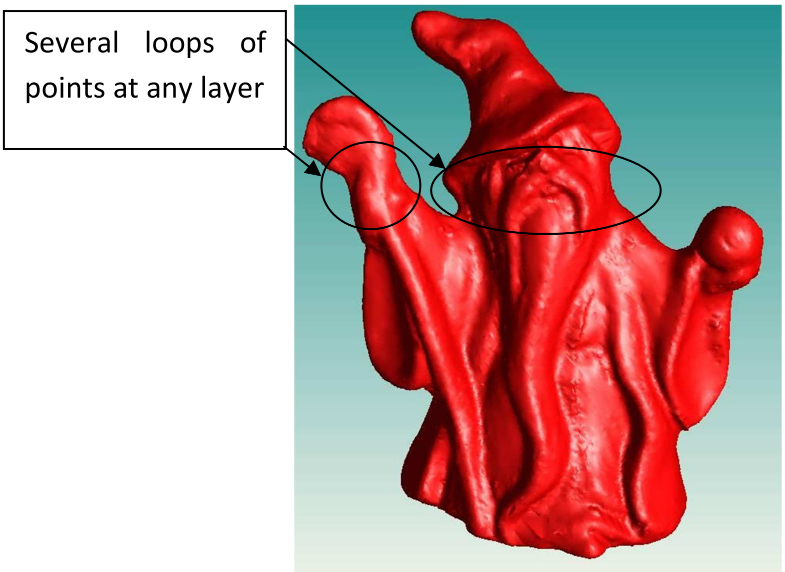


Graph 35 Joint angle travel for penguin model in configuration space sequencing

6.2.5 Wizard Component

The wizard component (Figure 6-23) is a highly complex model with re-entrant surfaces. This makes it an ideal model to test the intricate movement of the robot. The production of a digital model from the scanner data was difficult due to the complex nature of the model. Multiple scans were performed with different scanning paths incorporating registration markers to aid in the reconstruction of the solid model, this reduced the surface errors but did not eliminate them completely from the final computer model. The point generation algorithm operated over the surface of the model and extracted points at 10dpi resolution. This produced 2430 points which can be visualised as a path in Figure 6-24. This shows that the path includes the staff-holding arm on the left of the image Figure 6-23 which indicates that the wizard is facing forwards. Though this shows the surface points, the connectivity of the end effector path produce a much more complex path.

6.2.5.1 Lamina Slice Path



Several loops of points at any layer

Figure 6-23 Wizard without texture map

As the complex nature of the surface produces a much more varied end effector path the joint angles required to obtain these positions move through larger ranges to reposition. Looking specifically at the T2 joint, it is possible to see that the values still oscillate between ±180 degrees as the model requires painting over the entire surface (Graph 36). However, the movement between these values changes dramatically from that seen in the simpler models such as the eggcup.

Graph 36 shows the T2 joint angles. The first 400 points are associated with the upper part of the wizard, with the area following this i.e. 400 to 600 points, covering the wizard’s staff and neck. A more regular pattern appears after 600 points which ends at 1900-point mark, and could be seen to end at the wizard's waist. The final section covers the rest of the model from the waist down and produces a more regular pattern again.

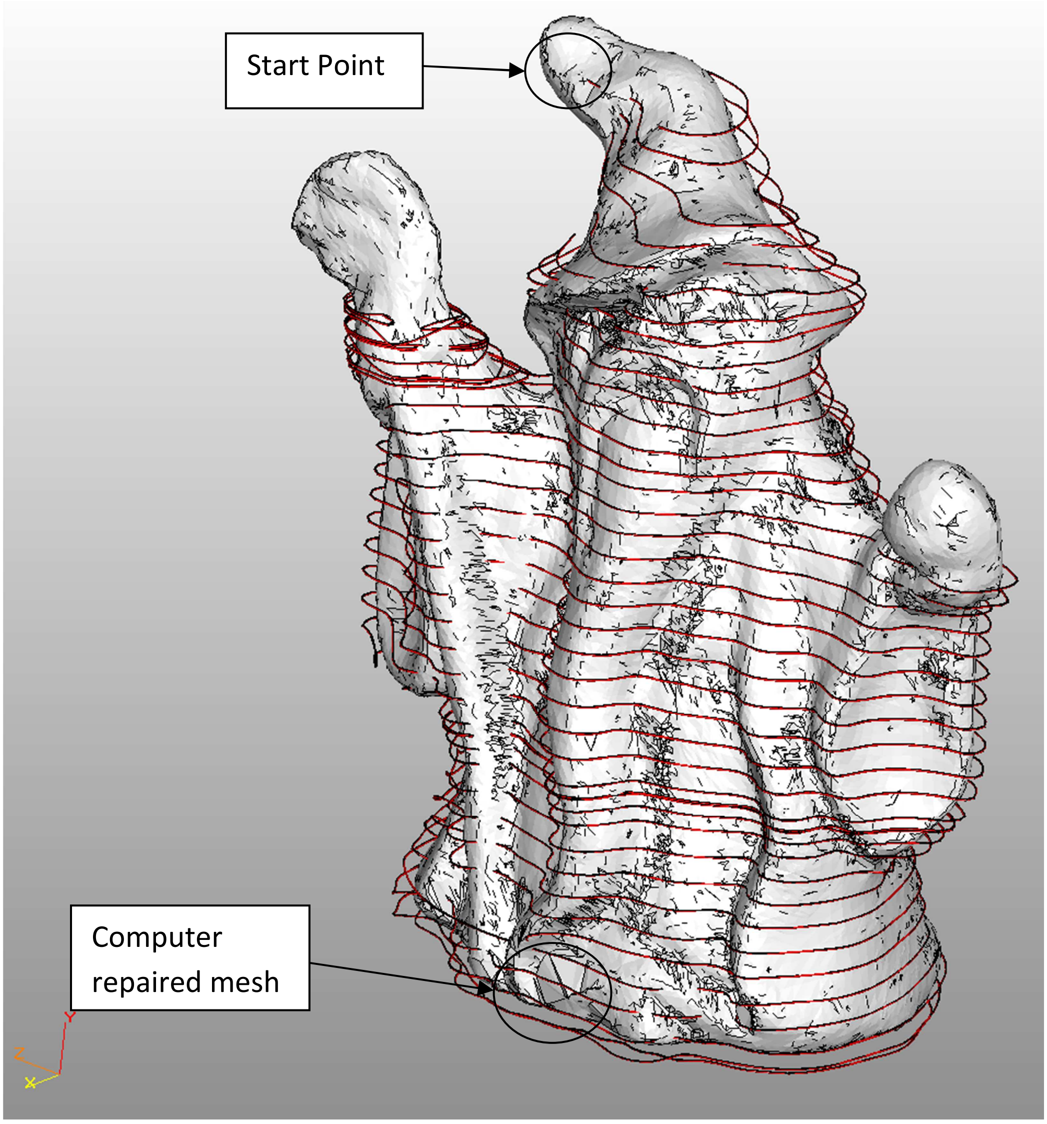


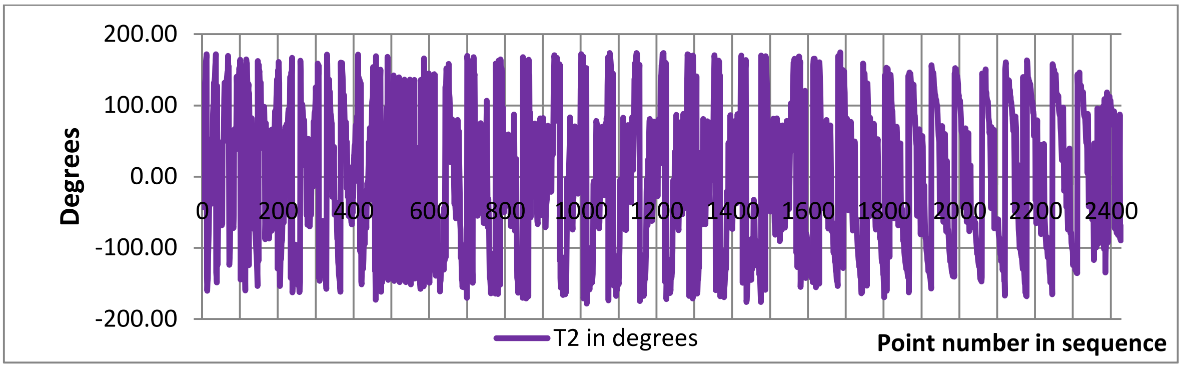
Figure 6-24 Lamina path on wizard model

The results for T1 are much more varied, with the majority of the tilt occurring within the ± 40 degrees variation. The area surround the hat (i.e. first 200 points) shows a large range of movement required, though at several points on the path the tilt requires to go outside the physical capabilities of the present system.

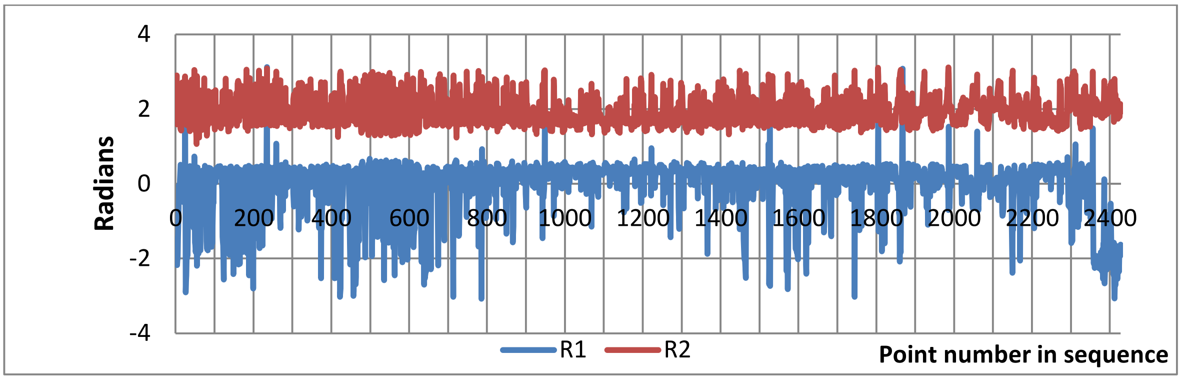
The robotic joints R1 and R2 can be seen in Graph 37 (where the convention of maintaining a positive R2 value whereever possible has been continued). The results suggest that the joint is not required to move through its full range of movements. While using this convention it is possible to see that the R1 axis is required to move through a much larger range of values than R2 (a side-effect of maintaining a positive R2 value).

The wizard's results have shown that a much more complex set of joint angles are required to expose all the points. It can also be seen that a large amount of feathering occurs in certain axes and this would be associated with the large travels which can be seen in Graph 38. The wizard is the same size as all the other models tested. However, the lamina approach to processing the wizard requires a great deal more joint travel than for the other test models. The results indicate that the T2 and R1 joints undergo the same amount of travel and T1 the shows the least travel. The overall movement is 2923 radians; more than double that for any other model and the end effector path shows why this model is so complex to paint. The max move has reduced to almost a third of the total movement which would suggest that all the joints are moving a significant amount between configurations.

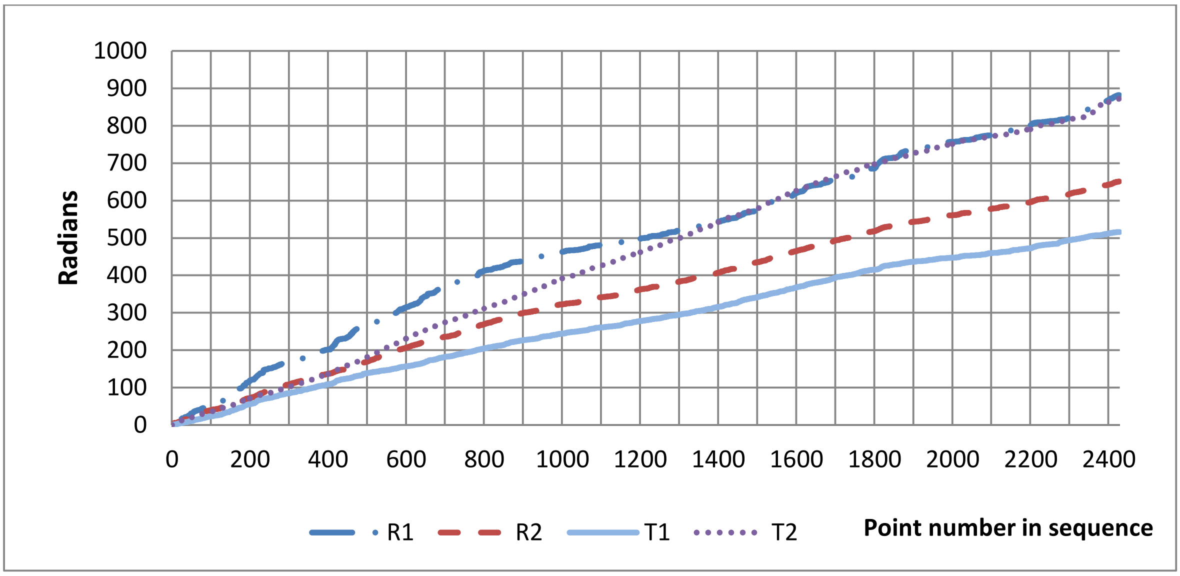
This model is the most complex model that the system could be required to paint in a commercial environment. The hand painting of such a component would require several coats of paint and a skilled artist, if the sequencing could improve the painting time the customization of such a component would be of great benefit to the manufacturing industry.



Graph 36 T2 joint angles for wizard while using lamina path



Graph 37 R1 and R2 joint angles for wizard using the lamina path



Graph 38 Joint travel for the wizard using the lamina path

The results obtained for the wizard are more difficult to understand due to the complexity of the model and due to the number of points. The path undulates more across the surface of the model than on previous models and with unique features like the staff and the hat, the wizard should create an interesting range of paths. The next section will look at the painting of the points using a Cartesian closest point path and then finally a configuration space path will be constructed.

6.2.5.2 Cartesian Closest Point Path

The Cartesian approach aims to process points based on surface closeness. With the wizard, the lamina path has shown that its complex shape creates an equally complex end effector path. The path created using the Cartesian closest point is presented in Figure 6-25 where it is possible to see that, rather than large sections being processed using either horizontal or vertical patterns, a combined approach is taken with small sections of each approach. The path appears to have at least two large positional changes one moving from processing points on the hat down to processing points on the orb (in the opposite hand to the staff). These large positional changes are caused by all the points in an area being painted resulting in the algorithm moving a larger distance to find a point still to be painted; these changes could be avoided with careful selection of the start position.

Selecting the start point obviously has a great affect on the final path, one approach could be to use a spatial point distribution algorithm to create a variety of different start positions and calculating the route for each of them. Although such an approach would not guarantee the best path, it could be used to identify common trends as to where better paths could be started. A second approach may be to look at an individual path to find where the greatest movement between points occurs and investigate if basing the start position from that point creates a shorter path. A final approach may be to implement some form of neural network which builds knowledge of different components and then uses geometric information to build a path based on previous models and statistical analysis to create a "*best guess*" for the shortest path.

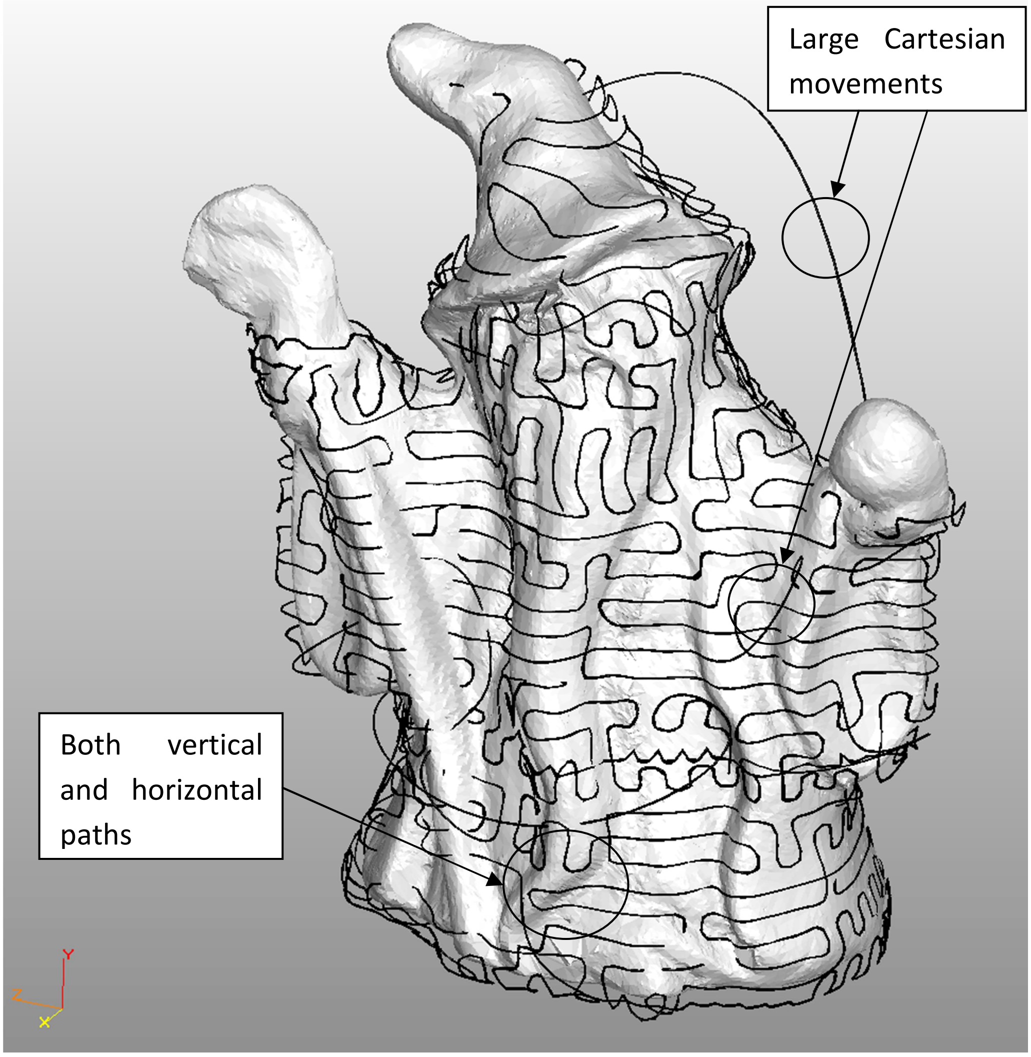
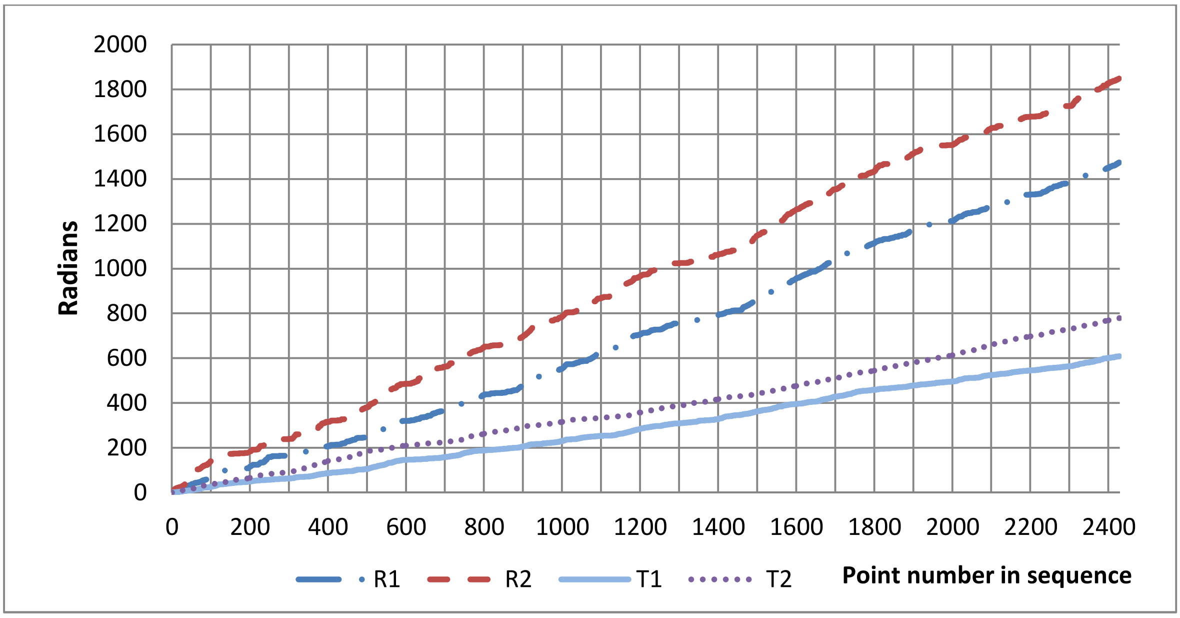


Figure 6-25 Cartesian closest point path for wizard

The overall travel that each of the joints is required to perform can be seen in Graph 39, which shows that the robotic joints R1 and R2 perform the greatest movement. The overall results show a large increase compared to the lamina approach. It is interesting to note that the Cartesian planning algorithm appears to have reduced the travel the T2 axis is required to undertake, this is in stark contrast to the other joints in the system. The max move for the Cartesian approach is 1724 radians.



Graph 39 Total travel of each joint using Cartesian closest point path for the wizard component

The final section will look at the configuration space path, the Cartesian path has shown an increase to the total joint travel compared to the lamina path, it is therefore interesting to find out if the path can be significantly reduced.

6.2.5.3 Configuration Space Closest Point Sequencing

Throughout this thesis, the aim has been to produce an algorithm capable of refining the path for complex geometric models. The results so far have shown that the configuration space closest point produces shorter paths than other approaches. The paths generated by the configuration space method cannot be usefully illustrated (Figure 6-26) because the trace over the surface of the wizard is obscured by many different movements between different locations.

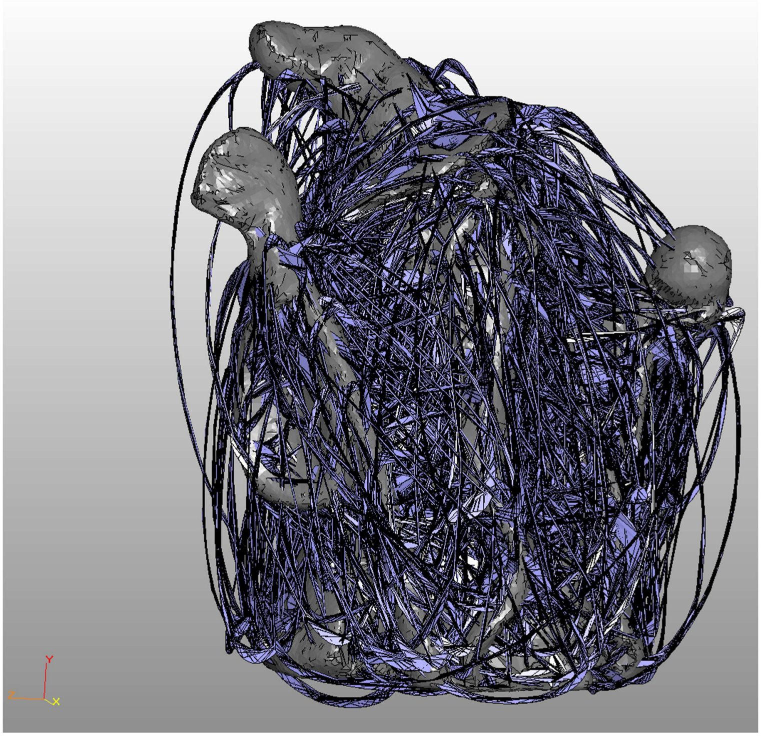
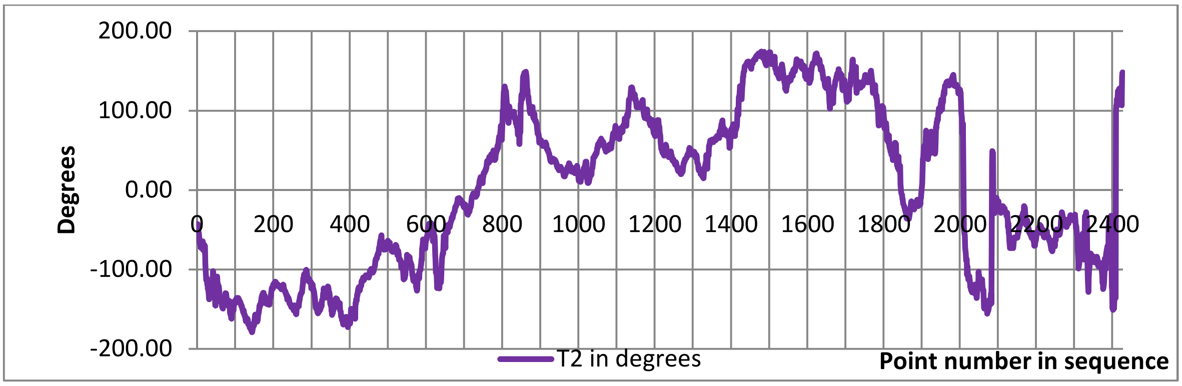


Figure 6-26 Configuration space path for wizard

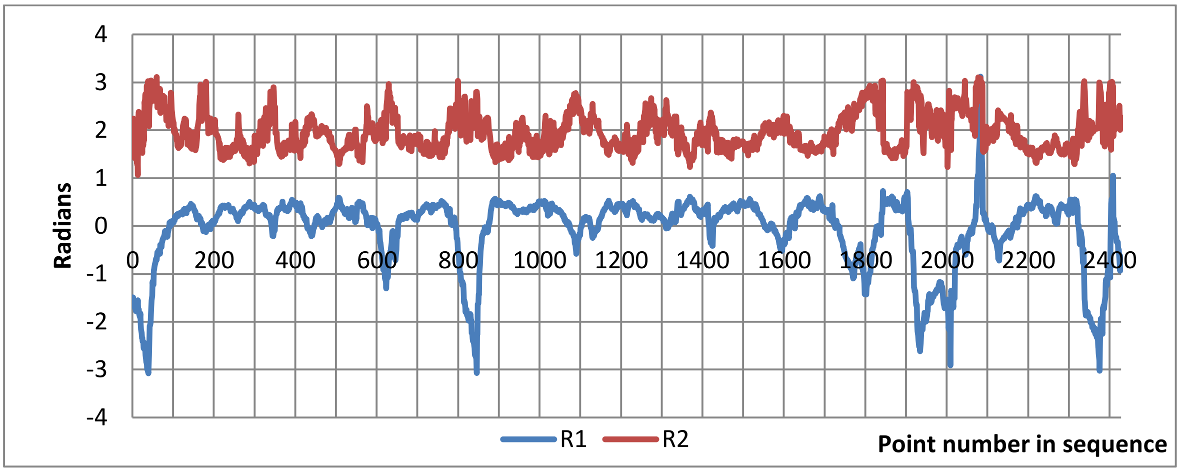
Because the configuration space path seeks to minimise the joint movement the rapid changing of joint positions that has been observed in previous paths is removed by using the configuration space path. This is best seen in Graph 40, which shows the T2 joint angles throughout the path. The robotic joint angles R1 and R2 can also be seen to produce much smaller movements between points (this can be seen in Graph 41) when this is compared to Graph 37 for the lamina path which appears to have lots of movement between points hence creating a feathered appearance.

The joint travel can be seen in Graph 42 where the R2 axis can be seen to undergo the greatest amount of travel. It is interesting to note that this is similar to the human practices mentioned earlier that the detailed components are painted using fine movements of the last members of the kinematic chain, such as the fingers in hand painting. The wizard's complex surface creates a large amount of detailed area. T2 is the second largest contributor to the overall travel and this has been the case for many of the components.

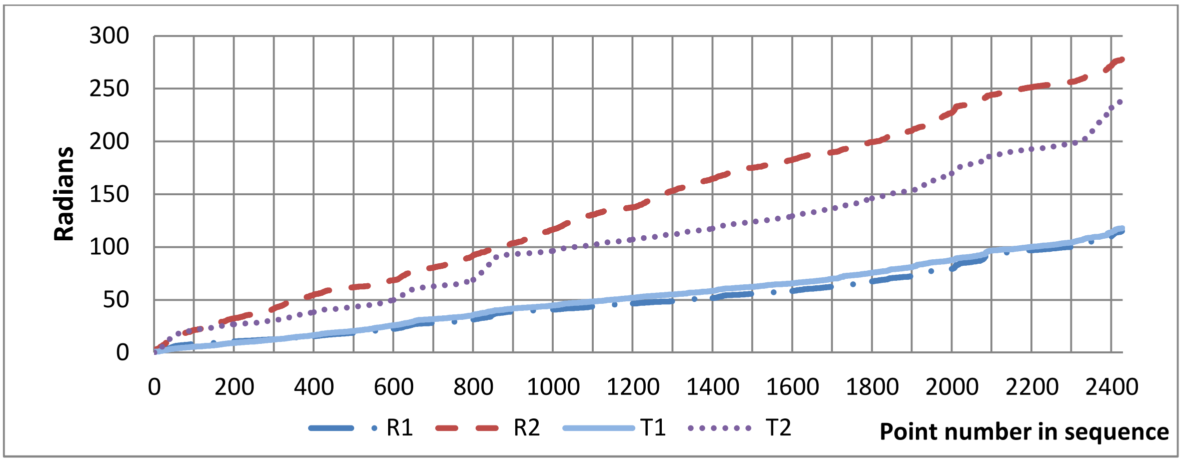
The configuration space algorithm created a path with an overall joint travel of 750 radians, which is much less than the lamina (2923 radians) and the Cartesian (4710 radians) path. Though the path does not appear visually less complex than the other paths, it is important to note that it would not be expected to look visually pleasing in a 3D environment as it is considering the specific test rig's robotic configuration. The max move for the configuration space path is approximately 1/3 of the total travel at 250 radians.



Graph 40 T2 joint angles after configuration space sequencing for the wizard



Graph 41 R1 and R2 joint angles for configuration space sequencing over the wizard



Graph 42 Total joint travel for each axis in configuration space algorithm over the wizard

6.3 Discussion of Results

Component / Sequencing	Lamina	T1	T2	R1	R2	Cartesian	Configuration
Validation 1: Cylinder	186 (164)		26 (23)	717 (467)	665 (663)	128 (121)	12 (8.7)
Validation 2: Deformed Cylinder	562 (552)	1682 (1315)	924 (616)	2074 (1594)	1942 (1587)	1437 (886)	150 (74)
Eggcup	503 (403)					593 (423)	267 (157)
Penguin	890 (442)					938 (465)	370 (182)
Wizard	2923 (1075)					4710 (1724)	750 (250)

Table 8 Summary of path lengths and max moves in radians obtained for different sequencing strategies, Max moves are in brackets

Table 8 summarises the results obtained for different sequencing strategies, the shortest path has been highlighted in yellow and it is apparent that the configuration space algorithm is constantly the shortest path for every test component. It is also possible to suggest that single joint sequencing does not reduce the path further than lamina sequencing in the majority of components, excluding T2 sequencing of the cylinder where a regular rotational makes lamina slicing larger.

Each of the components was designed to test a specific aspect of the sequencing methods; the cylindrical component was used to analyse single joint sequencing for a faceted computer-generated model. The results suggested that T2 sequencing would produce a much shorter path than the lamina approach due to the regular rotational values that were observed for the facetted model, where vertical columns of points could be processed rather than horizontal rotational points. This would reduce the amount of rotation to 2π rather than the 50π that would be required with the lamina approach. Though this sequencing of points is apparent for a system that only looks at the rotational joints, the reduction in the total travel of T2 would perhaps be offset by an increase in the prismatic joint (i.e. linear prismatic joint on the end effector). In other words, it is important to consider that although these results decrease the rotational joints, they may increase in the prismatic joint travel and so for any further work the prismatic joint must be incorporated into the results for a more complete picture.

The cylindrical test component showed that both the Cartesian and the configuration space algorithms rendered shorter paths than the lamina approach with the configuration path being less than 10% of the distance of the lamina approach. The configuration space approach is similar to the T2 path and uses only one jump between facets (i.e. columns of points). The Cartesian path uses an approach similar to the lamina slicing and has multiple jumps between facets, though rather than handling a single layer at a time like the lamina approach the Cartesian method appears to process two layers for each rotation.

The deformed component aimed to look at the single joint sequencing on a model that integrated a tilting component T1 into the configuration and how this affected single joint sequencing. The introduction of an additional configuration variable increased the complexity of the system; however, it did not change the outcome of the single joint sequencing where it was evident that it increased the overall travel. The use of the single joint approach relies on an incremental strategy where the smallest value is first and the largest last. This ordered constraint prevented points that were close in configuration with very small incremental changes from being processed together, if another point existed with a smaller incremental change in the sequencing joint then it was exposed first. For example, two points next to each other being sequenced by T2 may have a 0.02 radian difference, however a point on the opposite side of the component may only be 0.01 radian's difference and therefore the system must reconfigure for the point on the opposite side).

The results for the deformed cylinder suggest that, out of the single joint sequencing strategies, the T2 sequence produced the shortest path. This could be due to the simple rotational profile that the deformed component exhibits. The T2 path suggested that the greatest movement in Cartesian space would be to perform the painting on the top surface of the model, perhaps the T2 path would have produced a significantly shorter path had these points been excluded from the experiment. However, the value obtained for the configuration space approach suggests that the top surface's impact on the overall path can be minimised. Overall, the deformed cylinder demonstrates that single joint sequencing has difficulty operating over variable components, and more specifically, the sequencing of points based on the robotic joints R1 and R2 tends to increase the overall path.

The eggcup was the first scanned model investigated and was testing the capability of the system to handle a more varied data set with surface defects being detected by the scanner. An initial problem with the scanned data was that the normal vectors for each of the triangles were facing in the opposite direction which needed to be corrected for all scanned models. The results once the normals were corrected again showed that the single joint approach was unfeasible. The Cartesian approach produced a slightly larger path than the lamina approach by approximately 18% however; the configuration space approach produced the shortest path at 53% of the lamina approach.

The configuration space algorithm was developed for components such as the penguin where no clear strategy is obvious. The penguin has many different surface normals on each lamina slice and therefore processing the component would require a large amount of repositioning for the robotic end effector. The configuration space approach looks at the entire model and adaptively creates the surface path, concentrating not on the Cartesian position but the joint positions. The results suggested that a reduction greater than those with previous components could be obtained by using the configuration space strategy. Indeed the path generated using a configuration space approach was 41% of that generated using the lamina approach.

The final component represents the ambition of the optical painting system; it is a complex figurine with constantly changing surface normals and subtle details that any rapid-production system would like to replicate or paint. The lamina approach showed a much larger value for the joint travel (2923 radians) than previous methods and the Cartesian approach increased this further (4710 radians). The configuration space approach managed to reduce the value (750 radians) by nearly 75% of the lamina approach. It is evident from the results that the configuration space approach produces the lowest joint travel, however, it is surprising to find that the lamina approach normally creates the second shortest path out of the routines tested as it was originally perceived that the Cartesian closest point would be the second shortest.

Throughout the results section the total travel for the paths has been used to compare the effectiveness of each path. Table 8 however, also contains a second number which represents the max move for each path. This number is sum of all the largest joint movements for each point to point movement. It is interesting to note that as the complexity of the components has increased, the percentage of the total path that the max move makes up appears to reduce. This would indicate that rather than a single joint being predominant for each movement, the secondary axes are requiring more movement. Though this number has little value in sequential path planning for other planning approaches, such as time-coordinated paths, this would be an effective means of assessing the path length.

The results have shown that for each of the components, the configuration space approach produces the shortest path. To state that the best path planning strategy for every component is the configuration space approach would be difficult, though for each of the components that have been tested it has produced the lowest joint travel. The limitations and assumptions underlying the configuration space approach will be discussed in the next chapter.

Chapter 7 Conclusion

This thesis has investigated the generation of an efficient path for the surface coverage of 3D objects. The results have established that an approach using the sequencing of points based on their configuration space position generates the shortest path. This chapter reviews the initial aims and objectives along with defining the limitations of the approach before highlighting the contribution that this novel approach presents and finally identifying possible areas of further work that have been discovered throughout the research.

7.1 Aims and Objectives

The aims of this thesis were identified in Chapter 1 and will now be discussed.

The principle aim of this thesis was to minimise the production time for the rapid painting system. The results have shown that the configuration space path has reduced significantly the amount of joint travel that is required to visit all the surface points. Taking the reduction of the path as a reduction in the painting time, it can be suggested that this aim has been completed for the given start point. However, it would require further investigation to prove that the absolute “minimum” path had been found, as the effects of varying the path’s start point have not been investigated.

The path planning approach has been tested with several different geometries, and has successfully produced a shorter path than the N-algorithm. The Maple programme has calculated the joint angles for all of these paths and has found a real solution for all end effector positions. The Maple programme was further tested to see if a point that was unreachable would yield a solution and it was discovered that though it found a solution, that solution only existed within the complex domain (i.e. it was not a reachable value).

The configuration space path planning approach is a machine-centred approach, which relies on a Cartesian points file. These are then used to calculate the joint angles for a specific machine. The algorithm is adaptable to any point file and, so long as the kinematics of the machine are known, it could be applied to any machine. The nature of the application would not affect the path planning algorithm unless additional constraints are required.

This thesis has concentrated on the specific robotic configuration that is required for the optical painting application and the kinematic model has been derived based on this. It is hypothesised that, as long as the kinematic model can be derived and the redundancies constrained, then the theory could be implemented for any robotic setup. The calculation of higher degree of freedom systems may pose a difficulty with conventional computer algorithms; however, with the constant evolution of computer systems the processing power of a single computer may be capable of handling increasingly complex systems.

7.2 Limitations

The path planning approach undertaken within this thesis has been fully described in Chapter 5. The approach has several limitations which will be discussed.

1. The key aim of this thesis is to minimise the production time and it has been assumed that minimising the joint travel, minimises the production time. The algorithms that have been written for the Cartesian and configuration space closest point look at sequencing the points in a manner to reduce the path length. However, both of these algorithms use the initial point in the list as the start point for any path and therefore develop a sequenced path based on this initial variable. Therefore, the sequencing is defined by the start point and further improvements may be possible by selecting a different start position.
2. Many different aspects of the system determine the efficiency of a path and throughout this work the focus has been on joint movement and optimising the four primary rotational joints. However, the complete kinematic chain has a further two joints which have not been included in the calculations (in order to simplify the problem). It can be hypothesised that the rotational end effector joint would not produce better sequencing results and as the joint angles try to position the component with the normals aligned down an axis, the joint movement for the end effector would be small. However, the prismatic joint would introduce an interesting new planning approach to the system based on end effector height. It is felt that like T1 and T2 sequencing, this incremental approach would increase the total joint travel due to its lack of flexibility.
3. As with any scientific experiment the quality of the results is very dependent on the quality of the data used to create them. The computer-generated model had a regular faceting which meant that several points on each slice had the same normal. This meant that the end effector positions were close for each facet. The data obtained from the scanner required a certain amount of post processing to make sure that it was registered correctly and mesh-healing algorithms were used to remove excess data points. The scanner itself struggled with high curvature areas where the white bisque provided a reflective surface and corrupted data which was evident on the base of the penguin model (where a healing algorithm has been applied). However, the geometric form and the joint angle calculation algorithm processed the data that was provided and produced the results. This thesis was not testing the capabilities of the scanner but the ability to process the surface data provided.

4. The data obtained from the point extraction algorithm is experimental data and must be treated as this; it therefore does not handle self-interactions or visibility issues which would need to be addressed for a commercial application. This will be addressed in the further work section of this thesis (Section 7.3).

7.3 Further Work

In order to improve, the capabilities and performance of the path planning system future work has been identified in several areas.

1. Currently the system only encompasses the four major rotational joints angles (R1,R2, T1 and T2). For a more complete system, the integration of the prismatic joint and the end effector rational joint would be required. The end effector rotational joint was excluded from the calculations as the solution set that was obtained aimed to align the normals along the axis of the model space. Therefore, the orientation of the end effector would simply be orientating along the same axis of model space. The prismatic joint was excluded from the calculations to simplify the process however, further work could be considered to look at the effects of its inclusion and how the different axes respond to movement (i.e. settling time and acceleration).
2. It has previously been mentioned that the Cartesian and the configuration space paths are derived from the first point in the list (that is to say that the start point of the path is the first point in the list). As the initial list is created by the N-algorithm using lamina data extraction, the first point tends to exist at the highest point on the component and therefore the paths are derived from this point. It would however be interesting to look at the effect on the path when a different start point is selected perhaps in the middle of the component. It would be possible to create an algorithm to test all the locations on the surface of a model though, this would be a time consuming application.
3. This thesis has concentrated on developing a path for all the points obtained from the point extraction algorithm. It has not focused on the commercial exploitation of the results obtained. It has been apparent from the start of this thesis that a practical manufacturing system would be required for any commercial implementation of the optical painting system to check for visibility and self-interacting points (i.e. points where the beam interacts with the model before the desired surface or points where the robot collides with the model). Many different visibility algorithms have been presented for different applications and therefore further work would be required to adopt a previously developed algorithm or to produce a custom algorithm.
4. Throughout this thesis, the concentration has been on a point to point planning approach where each point is visited and exposed before proceeding to the next. It

has however, been understood that the most effective paths operate in a continuous fashion rather than stopping at each point. In Chapter 3, the current configuration of the cell controller was described, where sequential instructions are processed limiting the system to point to point planning, it would be very interesting to look at implementing a continuous path as this should reduce the painting time. Several key factors make continuous paths very difficult to implement within this application it would be very important to prevent the path from crossing itself, as this would result in overexposure of points. It would also be important that large robotic configuration changes were minimised as this again could cause overexposure. Perhaps the most difficult aspect would be controlling the laser intensity and movement rate so that image quality could be maintained. This form of problem has been seen in other applications such as spray painting where only relatively simple surfaces are painted using robots, as smooth movement of the robot is difficult to achieve (within the automotive industry they generate paths that extend well past the end of the component and allow for paint to be lost on overshoot).

5. The work carried out in Chapter 4 looking at the point extraction algorithm did not yield an improvement in image quality; it did however identify different strategies for point sampling. Further work would be required to investigate various different sampling approaches and the affect that sampling has on the image quality. An interesting avenue for research would be the capability to vary the point size and see how this might affect the coverage of a component, especially on areas of low detail.

7.4 Research Contribution

The proposed method represents an advance on conventional approaches, demonstrating that machine centred planning can be effective regardless of the input point set. The path planning system has the expandability to be used on a variety of different applications. Where a list of Cartesian coordinates is known and the robotic setup can be derived, it is possible to produce an efficient configuration space based path.

The results have shown that a reduction in the joint travel can be obtained by using a configuration space approach and it has suggested that a general configuration space strategy tends to produces the shortest path. Several other approaches have been outlined within this work and it has been possible to compare the results for the configuration space approach to the results from various other strategies. For all the components tested, the configuration space algorithm has reduced the joint travel by more than 40%. This may not reduce the production time by a similar margin, but it could be safely assumed to reduce the value by a significant margin.

Though the work on extracting the points more intelligently did not yield an image quality improvement, it did open avenues for further work into assessing image quality and intelligent point extraction.

References

1. *Direct writing of digital images onto 3D surfaces.* **Sung, Raymond, et al., et al.** 1, Edinburgh : Emerald Group Industrial Robot: An International Journal , 2006, Vol. 33.

2. **Duan and Singh.** Analysis of the vechical brake judder problem by employing a simplified source-path-receiver model. *Proceedings of the Institute of Mechanical Engineering, Part D: Journal of Automotive Engineering.* 2011.

3. **Engineering, Manufacturing.** NC Pioneer John T Parsons Dies. *Manufacturing Engineering.* July, 2007, Vol. 1.

4. *Factory Automation: From Numerical Control to Fieldbuses.* **Houston, John.** s.l. : IEN Europe's Milestones Article Series.

5. **JianHua, Fan.** *Geometric Modelling of 5-Axis Sculptured Surface Machining.* Birmingham, UK : PhD Thesis, 2006.

6. VX CAD-CAM. *QUICK MILL.* [Online] VX CAD-CAM, 2007. www.vx.com/quickmill.cfm.

7. **Ritchie, James and Simpson, Graham.** *Engineering Applications A Project-Based Approach.* Edinburgh : Butterworth Heinemann, 1998. 0-7506-2577-5.

8. **Planit Software Ltd.** *Edge Cam.* [Online] Planit Software Ltd., 2008. http://www.edgecam.com/.

9. **George F. Schrader, Ahmad K. Elshennawy.** *Manufacturing Processes & Materials.* s.l. : Society of Manufacturing Engineers; 4th edition (July 2000), 2000. 978-0872635173.

10. **Hexagon Metrology.** Product Description Hexagon Metrology. *Hexagon Metrology.* [Online] 14 August 2008. ftp://ftp.it.dea.it/Product_catalogues/Global/Global_GB.pdf.

11. CMM Programming Limited. *CMM Programming Limited.* [Online] http://www.cmmprogramming.co.uk/index_files/cmm-services.htm.

12. **Xspect Solutions, Inc.** CMMs and their history. *Xspect Solutions, Inc.* [Online] 14 August 2008. http://www.ecmmsales.com/cmm-history.htm.

13. **Renishaw .** 5-axis measurement: Renscan 5 a measurment revolution. *Renscan 5 a measurment revolution.* [Online] 19 August 2008. http://www.renscan5.com/en/6055.aspx.

14. **Carl Zeiss.** Industrial Metrology. *Carl Zeiss Europe.* [Online] 2008. http://www.zeiss.de/.

15. *3D Imaging and Ranging by Time–Correlated Single Photon Counting.* **Wallace, Andy M, Buller, G S and Walker, A C.** Edinburgh : IEE Computing and Control Engineering Journal, 2001, Vol. 12.

16. **Faro.** Faro. [Online] December 2009. http://www.faro.com/UK.aspx.

17. *Path planning of multi-patched freeform surfaces for laser scanning.* **Seokbae, Son, Seungman, Kim and Kwan, H. Lee.** Gwangju : Springer-Verlag: International Journal of Advanced Manufacturing Technology, 2003, Vol. 22. DOi: 10.1007/s00170-002-1502-0.

18. *CAD-based path planning for 3-D line laser scanning.* **Xi, F and Shu, C.** Ontario : Elsevier: Computer Aided Design, 1999, Vol. 31.

19. Atomization and Spray Technology. *Norman Chigier.* [Online] http://www.normanchigier.com/DepositionofPaintonCarsAutomobiles-AtomizationandSprayTechnology.html.

20. *CAD-based automated robot trajectory planning for spray painting of free-form surfaces.* **Chen, Heping, et al., et al.** 5, Michigan : Emerald: Industrial Robot: An International Journal, 2002, Vol. 29. 0143-991X.

21. *Development of an optimal trajectory model for spray painting on a free surface.* **Diao, X, Zeng, S and Tam, V.** 1, s.l. : Computers & Industrial Engineering, 2009, Vol. 57.

22. *Process Simulation and Paint Thickness Measurement for Robotic Spray Painting.* **ARIKAN, M and BALKAN, T.** s.l. : CIRP Annals - Manufacturing Technology, 2001, Vol. 50.

23. *Steady-state paint flow under high centrifugal force:: atomization in spray painting.* **Kazama, S.** 4, s.l. : JSAE Review, 2003, Vol. 24.

24. *Towards Optimal Coverage of 2-Dimensional Surfaces Embedded in R^3: Choice of Start Curve.* **Atkar, Prasad N, Choset, Howie and Rizzi, Alfred A.** Las Vegas : IEEE, 2003. 0-7803-7860-1.

25. *By obersvation of painters at MM1 model makers .* www.justliketherealthing.co.uk, Glasgow : s.n., 2007.

26. *By personal observations at the games workshop in Edinburgh.* Edinburgh : s.n., 2008.

27. **Solid CAM.** Solid CAM, CAM-Modules. *Solid CAM.* [Online] Solid CAM, 11 August 2008. http://www.solidcam.com/hsm_finishing_strategies_en,15968.html.

28. **Graphics Products North America Inc.** cam-tool V3 Products. *camtool.net.* [Online] GP North America Inc., 11 August 2008. http://www.camtool.net/features.php#3D%20CAM%20ROUGHING.

29. *Optimizing tool orientations for 5-axis machining by configuration-space search method.* **Jun, Cha-Soo, Cha, Kyungduck and Lee, Yuan-Shin.** Gyeongnam : Elsevier: Computer-Aided Design, 2003, Vol. 35. 0010-4485/03.

30. *A study of the effects of cutter path strategies and orientations in milling.* **Toh, C K.** 152, Birmingham : Elsevier Journal of Materials Processing Technology, 2004.

31. *Methods to Optimize the Tool Path of the Five Axis Milling Machine Developed by the SIIT-AIT Research Group "5 Axis Thai".* **Makhanov, S, Munlin, M and Bohez, E.** Bangkok : Asian Conference on Infustrial Automation and Robotics, 2005.

32. *Ellipse-offset approach and inclined zig-zag method for multi-axis roughing of ruled surface pockets.* **Koc, Bahattin and Lee, Yuan-Shin.** 12, s.l. : Elsevier Science Ltd, Computer-Aided Design, 1998, Vol. 30.

33. *Generation of fractal toolpaths for irregular shapes of surface finish areas.* **Chen, Chao-Chang A, Juang, Yuh-Shuh and Lin, Wei-Zhi.** s.l. : Elsevier Material Processing Technology, 2002, Vol. 127.

34. *Curvilinear space-filling curves for five-axis machining.* **Anotaipaiboona,, Weerachai and Makhanovb, Stanislav.** s.l. : Elsevir, Computer-Aided Design, 2008, Vol. 40.

35. *INTERACTIVE FREEFORM CLAY MODELING SUPPORTED BY 3D SCANNING AND ROBOT MACHINING.* **Song, Y, Vergeest, J and Wiegers, T.** Salt lake City, Utah : Proceedings of DECT ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering, 2004.

36. *Clean-up tool path generation by contraction tool method for machining complex polyhedral models.* **Yongfu, Ren, Hong, Tzong Yau and Yuan-Shin, Lee.** Carolina : Elsevier Computers in Industry, 2004, Vol. 54.

37. *Three-axis machining of compound surfaces using flat and filleted endmills.* **Hwang, Ji Seon and Chang, Tien-Chien.** 8, s.l. : Elsevier Computer Aided Design, 1998, Vol. 30.

38. *Admissible tool orientation control of gouging avoidance for 5-axis complex surface machining.* **Lee, Yuan-Shin .** 7, s.l. : Elsevir, Computer-Aided Design, 1997, Vol. 29.

39. *NC machine tool path generation from CSG part representations.* **Bobrow, James E.** 2, s.l. : Elsevir, Computer Aided Design, 1985, Vol. 17.

40. *Ellipse-offset approach and inclined zig-zag method for multi-axis roughing of ruled surface pockets.* **Lee, Yuan-Shin and Koc, Bahattin.** 12, s.l. : Elsevier Computer Aided Design, 1998, Vol. 30.

41. *Periodic Global Parameterization.* **Ray, Nicolas, et al., et al.** s.l. : ACM Transactions on Graphics, 2006.

42. **Floater, Michael and Hormann, Kai.** Surface Parameterization: A Tutorial and Survey. *University of Oslo.* [Online] http://folk.uio.no/michaelf/papers/surfparam.pdf.

43. *The implementation of adaptive isoplanar tool path generation for the machining of free-form surfaces.* **Ding, A, et al., et al.** 7, Singapore : Springer London: The International Journal of Advanced Manufacturing Technology, 2004, Vol. 26. 10.1007/s00170-004-2058-y.

44. *Guide surface based tool path generation in 3-axis milling: an extension of the guide plane method.* **Kim, B H and Choi, B K.** s.l. : Elsevier, Computer Aided Design, 2000, Vol. 32.

45. *Rapid and Flexible Prototyping through Direct Machining.* **Li W., Davis T., Jensen C.G., Red W.E.,.** s.l. : Computer-Aided Design and Applications, 2004, Vol. 1.

46. *Non-Isoparametric tool path planning by maching strip evaluation for 5-axis sculptured surface machining.* **Yuan-Shin, Lee.** 7, s.l. : Elsevier: Computer Aided Design, 1998, Vol. 30.

47. *Non-Distorted Texture Mapping For Sheared Triangulated Meshes.* **Levy, Bruno and Mallet, Jean-Laurent.** s.l. : International Conference on Computer Graphics and Interactive Techniques Proceedings of the 25th annual conference on Computer graphics and interactive techniques , 1998. 0-89791-999-8 .

48. *Spatial Planning: A Configuration Space Approach.* **Lozano-Perez, Thomas.** AI Memo No 605, Boston : Massachusetts Institute of Technology, 1980.

49. **Foley, James, et al., et al.** *Computer Graphics Principles and Practice.* s.l. : Addison-Wesley Publishing Company, 1993. 0-201-12110-7.

50. **Spong, Mark W, Hutchinson, Seth and Vidyasagar, M.** *Robot Modelling and Control.* New Jersey : John Wiley & Sons, Inc, 2006. 978-0-471-64990-8.

51. **Manseur, Rachid.** *Robot Modeling and Kinematics.* Boston : Da Vinci Engineering Press, 2006. 1-58450-851-5.

52. *A Survey of Global Configuration-Space Mapping Techniques for a Single Robot in a Static Environment.* **Wise, Kevin D and Bowyer, Adrian.** 8, Bath : The International Journal of Robotic Research, Sage Publications Inc., 2000, Vol. 19.

53. **M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf.** *Compututional Geometry, Algorithms and Applications .* s.l. : Springer-Verlag, 2008. 978-3-540-77973-5.

54. *The Minkowski Sum of 2D Curved Objects.* **Wien, In-kwon Lee and Myung-soo Kim and Gershon Elber and Tu.** s.l. : Proceedings of Israel-Korea Bi-National Conference on New Themes in Computerized Geometrical Modeling, 1998.

55. *Computing Metric Topological Properties of Configuration-Space Obstacles.* **Brost, Randy C.** 170-176pp, Pittsburgh : International Conference Robotics Automation, 1989, Vol. 1.

56. *Accurate Minkowski sum approximation of polyhedral models.* **Gokul Varadhan, Dinesh Manocha.** s.l. : Academic Press Professional, Graphical Models, 2006, Vol. Special issue on PG2004. 1524-0703.

57. *Minkowski sum boundary surfaces of 3D-objects.* **Martin Peternell, Tibor Steiner.** 3, s.l. : Academic Press Professional, Graphical Models, 2007, Vol. 69.

58. *Exact and Efficient Construction of Minkowski Sums of Convex Polyhedra with Applications.* **Fogel, Efi and Halperin, Dan.** 11, s.l. : Elsevir, Computer-Aided Design , 2007, Vol. 39. 0010-4485 .

59. *A Subdivision Algorithm in Configuration Space for Findpath with Rotation.* **Brooks, Rodney A and Lozano-Perez, Tomas.** AI Memo No 605, Boston : Massachusetts Institute of Technology, 1982.

60. *Shadow algorithms for computer graphics.* **Crow, Franklin C.** 2, s.l. : ACM, ACM SIGGRAPH Computer Graphics , 1977, Vol. 11.

61. *Computing Configuration Space for a Robot on a Mesh-Of-Processors.* **Dehne, F, Hassenklover, A and Sack, J R.** 11, Ottawa : Proceedings of The International Conference on Parallel Processing, IEEE, 1989.

62. **Wikipedia.** *Wikipedia Image Bank.* [Online] 2008. http://en.wikipedia.org.

63. **Choi, Wonyun.** *Contingency Tollerant Robot Motion Planning and Control.* Stanford, California : s.n., 1993.

64. *New Heuristic Algorithm for Efficient Hierarchical Path Planning.* **Zhu, David and Latombe, Jean-Claude.** 1, Stanford : Transaction of Robotics and Automation, IEEE, 1991, Vol. 7.

65. *Morse Decompositions for Coverage Tasks.* **Acar, Ercan U, et al., et al.** s.l. : The International Journal of Robotics Research, Sage Publishing, 2002, Vol. 21.

66. *Analytical Solution for Configuration Space Obstacle Computation and Representaion.* **Zhao, C S, Farooq, M and Bayoumi, M M.** Orlando, FL, USA : Proceedings of the 1995 Industrial Electronics, Control, and Instrumentation 21st International Conference, 1995, Vol. 2. 0-7803-3026-9.

67. **Davis, Stan.** *Future Perfect: 10th Anniversary Edition.* s.l. : Basic Books; Updated edition (October 5, 1997), 1987. 978-0201327953.

68. **Garmin International.** Garmin UK. *Garmin* . [Online] September 2010. http://www.garmin.com/sites/uk/.

69. *Limits and opportunities in mass customization for "build to order" SME's.* **Svensson, C and Barford, A.** s.l. : Computers in Industry, 2002, Vol. 49.

70. *A Study of FMS Part Type Selection Approaches for Short-Term Production Planning.* **STECKE, KATHRYN and KIM, ILYONG.** s.l. : Kluwer Academic Publishers, The International Journal of Flexible Manufacturing Systems, 1988, Vol. 1.

71. *Refining and Extending the Business Model With Information Technology: Dell Computer Corporation.* **Kramer, Kennith L, Dedrick, Jason and Yamashiro, Sandra.** Irvine : Taylor Francis: The Information Society, 2000, Vol. 16. 0197-2243/00.

72. *Market Structure and the Timing of Technology Adoption with Network Externalities.* **Choi, Jay Pil and Thum, Marcel.** 2, s.l. : Elsevier Science B.V. , European Economic Review, 1997, Vol. 42.

73. *The Customer is the Company.* **Chafkin, Max.** s.l. : Inc. Magazine, 2008, Vol. June.

74. **Nike Europe.** Nike ID. *Nike ID.* [Online] http://nikeid.nike.com/.

75. **Adidas Europe.** Adidas Create Your Own. *Adidas.* [Online] http://www.adidas.com/.

76. **Hippel, Eric von.** *Democratizing Innovation.* s.l. : MIT Press, 2005. 0-262-00274-4.

77. **Threadless.** *Threadless.* [Online] 2008. http://www.threadless.com/.

78. **Winamp.** *Winamp Media Player.* [Online] 2008. http://www.winamp.com/.

79. **Microsoft.** *Microsoft Office.* [Online] Microsoft, 2008. www.microsoft.com.

80. **BBC.** *BBC.* [Online] 2008. www.bbc.co.uk.

81. **IKEA.** IKEA UK. *IKEA.* [Online] 2008. www.ikea.com/gb/en/.

82. *From Value Chain to Value Constellation: Designing Interactive Strategy.* **Normann, Richard and Ramirez, Rafael.** Harvard : Harvard Business Review, 1993, Vols. July-August.

83. *What is Strategy?* **Porter, Michael.** Boston : Harvard Business Review, 1996, Vols. November-December.

84. **DGA, Roland.** Roland DGA. [Online] September 2010. http://www.rolanddga.com/.

85. **Chua, C K, Leong, K F and Lim, C S.** *Rapid Prototyping: Principles and Applications.* s.l. : World Scientific, 2003. 981-238-117-7.

86. *Mass Customization: Implementing the Emerging Paradigm for Competative Advantage.* **Kotha, Suresh.** New York : Strategic Managment Journal, 1995, Vol. 16.

87. *CUSTOMERIZATION: THE NEXT REVOLUTION IN MASS CUSTOMIZATION.* **Wind, Jerry and Rangaswamy, Arvind.** 1, s.l. : Journal of Interactive Marketing, 2001, Vol. 15.

88. **James, Greg, et al., et al.** Image Processing. [book auth.] Randima Fernando. *GPU GEMS.* s.l. : Addison-Wesley, 2005.

89. **Catmull, E.** *A Subdivision Algorithm for Computer Display of Curved Surfaces.* Salt Lake City, Utah : Ph.D Thesis , 1974.

90. *Texture and reflection in computer generated images.* **Blinn, James F and Newell, Martin E.** 10, University of Utah, Salt Lake City : ACM, 1976 , Vol. 19. 0001-0782 .

91. *Perceptual Decomposition of Virtual Haptic Surfaces.* **Rosenberg, Louis and Adelstein, Bernard.** San Jose : Proceedings IEEE 1993 Symposium on Research Frontiers in Virtual Reality, 1993.

92. *Project FEELEX: Adding Haptic Surface to Graphics.* **Iwata, Hiroo, et al., et al.** LA, California : ACM SIGGRAPH, 2001.

93. *Spacial Occupancy Enumeration.* **Christensen, A H.** s.l. : SIGGRAPH 80 Conference Proceedings, Computer Graphics, 1980.

94. *Neighbor Finding in Image Represented by Octrees.* **Samet, H.** s.l. : Computer Vision, Graphics and Image Processing, 1989.

95. **Hoppe, Hugues, et al., et al.** *Mesh Optimization.* Redmond : Microsoft Reseach Press.

96. **Smart Technologies.** Product Information Smart Board. *Smart Technologies.* [Online] 1 September 2008. http://smarttech.com/.

97. **GxSaurav.** A designer's perspective on life. *GxSaurav: A designer's perspective on life.* [Online] October 2008. http://gxsaurav.com/mental-ray-materials-test.html.

98. **Meggs, Phillip.** *A History of Graphic Design.* s.l. : John Wiley & Sons, 1998. 978-0471291985 .

99. **Hudson, Paul.** Finishing Processes. *Blue Print Course Ware.* [Online] 2008. http://www.ider.herts.ac.uk/school/courseware/manufacture/finishing/hotdip.html.

100. **Roland.** Rolanddga. [Online] September 2010. http://www.rolanddga.com/.

101. Alternative Photography. *Liquid Light .* [Online] August 2009. http://www.alternativephotography.com/process_liquidlight.html.

102. **Reed, Martin and Jones, Sarah.** *Silver Gelatin: A User's Guide To Liquid Photographic Emulsions.* London : Argentum, 1995. 1-902538-15-3.

103. **Rudman, Tim.** *The Photographer's Toning Book: The Definative Guide.* New York : Amphoto Books, 2003. 0-8174-5465-9.

104. **James, Christopher.** *The Book of Alternative Photographic Processes.* New York : Delmar, 2002. 0-7668-2077-7.

105. **Black, Iain, et al., et al.** *EPSRC Project Proposal Optical Painting System.* Edinburgh : s.n., 2003.

106. **Epson.** *SPEL for Windows 2.0 Options, VB Guide.*

107. **EPSON ROBOT.** *Program Develop Software SPEL for windows 2.0.* s.l. : EPSON.

108. **Johnson, Brian, Skibo, Craig and Young, Mark.** *Inside Microsoft Visual Studio 2003.* s.l. : Microsoft Press, 2003. 0-7356-1874-7 .

109. *Optimal Robot Speed Trajectory by Minimization of the Actuator Motor Electromechanical Losses.* **ELEFTHERIA, S, GEORGE, S and ANASTASIOS, P.** s.l. : Journal of Intelligent and Robotic Systems, 2002, Vol. 33.

110. *Adaptive robot speed control by considering map and localization uncertainty.* **Negishi , Y, Miura , J and Shirai , Y.** s.l. : Proceedings of the 8th Int. Conf. on Intelligent Autonomous Systems, 2004.

111. **Engelberger, Joseph F.** *Robotics In Practice, Management and applications of industrial robots.* London : Kogan Page Limited, 1980. 0850383927.

112. **NASA.** NASA Robotic's. *Robot Systems.* [Online] NASA, 2010. http://prime.jsc.nasa.gov/ROV/systems.html.

113. **Leler, W m and Merry, J.** *3D with Hoops: Build interactive 3D graphics into your c++ applciatoins .* s.l. : Addison Wesley Longman Inc, 1996. 0-201-87025-8.

114. **Blanchette, Jasmin and Summerfield, Mark.** *C++ GUI Programming with Qt 3.* s.l. : Prentice Hall PTR , 2004. 0131240722.

115. *DIRECT WRITING OF DIGITAL IMAGES ONTO 3D SURFACES.* **Sung, R, et al., et al.** Long Beach : Proceedings of IDETC/CIE 2005 , ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, 2005.

116. *PATH PLANNING FOR AUTOMATED ROBOT PAINTING.* **McPherson, Finlay, Corney, Jonathan and Sung, Raymond.** Las Vegas : Proceedings of the ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, 2007.

117. **Stam, Jos.** Simulating Diffraction. *GPU GEMS.* 1, 2004, Vol. 8, 0321228324.

118. **Novosad, Justin.** Advanced High-Quality Filtering. *GPU GEMS 2.* 2005, Vol. 2, 27.

119. *Greyscale and Resolution Tradeoffs in Photographic Image Quality.* **Farrell, Joyce E.** San Jose, CA, USA : Proceedings of SPIE -- Volume 3016 Human Vision and Electronic Imaging II , 1997. 10.1117/12.274508 .

120. *Image quality metrics for halftone images.* **Mitsa, Theophano.** Chicago, IL, USA : Imaging Technologies and Applications , 1992 . 0277-786X.

121. *Trade-off between spatial resolution and gray-scale coding for letter recognition.* **MiYoung , Kwon and Legge, Gordon.** 7, Minnesota : Journal of Vision, 2010, Vol. 10.

122. *Exploring Ink Spreading.* **Emmel, Patrick and Hersch, Roger David.** Scottsdale, Arizona, USA : Proceedings of the 8th IS&T/SID Color Imaging Conference: Color Science and Engineering, 2000.

123. *WHY IS IMAGE QUALITY ASSESSMENT SO DIFFICULT?* **Wang, Zhou, Bovik, Alan C and Lu, Ligang.** Orlando : IEEE International Conference on Acoustics, Speech, & Signal Processing, 2002. 0-7803-7402-9 .

124. *Low-discrepancy point sampling of meshes for rendering.* **Quinn, Jonathan, Langbein, Frank C and Martin, Ralph R.** Prague, Czech Republic : Point-based graphics 2007 : Eurographics/IEEE VGTC Symposium proceedings, 2007. 3905673517.

125. *Stratified Point Sampling of 3D Models.* **Nehab, Diego and Shilane, Philip.** s.l. : Eurographics Symposium on Point-Based Graphics, 2004.

126. *Density-Controlled Sampling of Parametric Surfaces Using Adaptive Space-Filling Curves.* **Quinn, Jonathan A, et al., et al.** Cardiff : Geometric Modeling and Processing ,Springer Berlin / Heidelberg, 2006, Vol. 4077. 978-3-540-36711-6.

127. **Nehab, Diego and Shilane, Philip.** Stratified Point Sampling of 3D Models. *Stratified Point Sampling Write Up.* [Online] 2008. http://www.tecgraf.puc-rio.br/~diego/academic/phd/597d/sample/.

128. *Using Space Filling Curves for Multi-Dimensional Indexing.* **Lawder, J and King, P.** London : s.n., 2000. 3-540-67743-7.

129. **MIT.** Space Filling Curves. *Multi Dimensional Space Filling Curves.* [Online] MIT, September 2010. people.csal.mit.edu/jaffer/geometry/mdsfc.

130. **Wikipedia.** Space Filling Curve. *Wikipedia.* [Online] 2008. http://en.wikipedia.org/wiki/Space-filling_curve.

131. **Cardiff University.** Cardiff University Department of Computer Science. *http://www.cs.cf.ac.uk/.* [Online] 2010.

132. *Density-Controlled Sampling of Parametric Surfaces Using Adaptive Space-Filling Curves.* **Quinn, Jonathan, et al., et al.** s.l. : Proceedings Geometric Modeling and Processing, Springer, 2006. 354036711X.

133. *A Common Data Management Infrastructure for Adaptive Algorithms for PDE Solutions.* **Parashar, Manish, et al., et al.** San Jose, CA : Conference on High Performance Networking and Computing, Proceedings of the 1997 ACM/IEEE conference on Supercomputing , 1997. 0-89791-985-8 .

134. **N-Nagy, Francis and Siegler, Andras.** *Engineering Foundations Of Robotics.* London : Prentice-Hall International, 1987. 0-13-278797-0.

135. **Mckerrow, Phillip John.** *Introduction to Robotics.* Sydney : Addison-Wesley Publishing Company, 1998. 0-201-18240-8.

136. **Khatib, Oussama.** Academic Earth. *Stanford Computer Sciences.* [Online] 02 September 2010. http://see.stanford.edu/materials/aiircs223a/handout3_Kinematics-2.pdf.

137. **MapleSoft.** Maplesoft. *Maple Maths.* [Online] August 2009. http://www.maplesoft.com.

138. **Garvin, Frank.** *The Maple Book .* 2001. 1584882328.

139. **Reed, Martin and Jones, Sarah.** *Silver Gelatin: A User's Guide to Liquid Photgraphic Emulsions.* London : Aurum Press Limited, 2005. 1-902538-15-3.

140. **Oxford Dictionary.** *Oxford Compact Dictionary.* Oxford : Oxford University Press. 978-0-19-861022-9.

141. *Coverage Path Planning: The Boustrophedon Cellular Decomposition.* **Choset, Howie and Pignon, Philippe.** s.l. : Proceedings of International Conference on Field and Service Robotics, 1997.

142. **Wolfe, Rosalee.** Mapping Techniques. *Siggraph Educational Resources.* [Online] Siggraph, 1997. http://education.siggraph.org/resources/cgsource/instructional-materials/slide-sets/FrontPage/slideshow?slides=1997_mapping&show=1&type=jpg.

143. **Chua, Chee Kai, Leong, Kah Fai and Lim, Chu Sing.** *Rapid Prototyping- Principles and Applciations.* Singapore : World Scientific Publishing Co. Pte. Ltd., 2003. 981-238-120-1.

144. *Simulation System modelling for mass customization manufacturing.* **Qiao, G, Mclean, C and Riddick, F.** Gaithersburg : Proceedings of the 2002 Winter Simulation Conference, 2002.

145. *Research of the optimization methods for mass customization(MC).* **Gu, X J, et al., et al.** s.l. : Journal of Materials Processing Technology, 2002, Vol. 129.

146. *Special Issue Editorial: the what, why and how of mass customization.* **McCarthy, Ian P.** 4, s.l. : Production Planning and Control, 2004, Vol. 15. 0953-2787.

147. Adidas. *Adidas Global.* [Online] August 2008. www.adidas.com/ -.

148. *Sharpness Rules.* **Johnson, Garrett M and Fairchild, Mark D.** Scottsdale : Society for Imaging Science and Technology, 2000.

149. **InternationalTelecommunicationUnion.** *SERIES P: TELEPHONE TRANSMISSION QUALITY,TELEPHONE INSTALLATIONS, LOCAL LINE,NETWORKS.* Switzerland : TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU, 2006.

150. **Smith, Steven.** *The Scientist and Engineer's Guide to Digital Signal Processing.* s.l. : California Technical Pub, 1998. 978-0966017632 .

151. **Blender.** Blender Users Guide. *Blender.Org.* [Online] Blender.Org, October 2008. http://wiki.blender.org/index.php/BSoD/Introduction_to_Materials/part1j.

152. *NO-REFERENCE PERCEPTUAL QUALITY ASSESSMENT OF JPEG COMPRESSED IMAGES.* **Zhou, Wang, Hamid, Sheikh and Bovic, Alan.** s.l. : International Conference on Image Processing. 2002. Proceedings. , 2002. 1522-4880 .

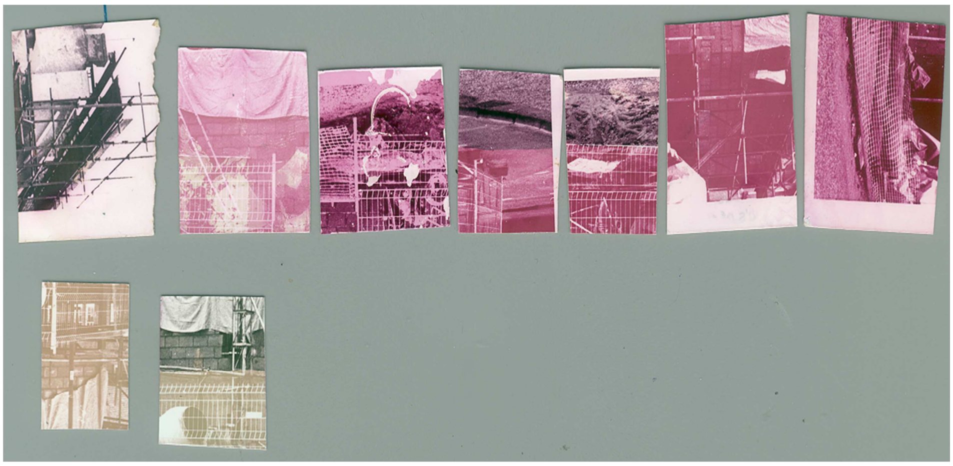
Appendix 1

Toning and Uniform Patterns

During the initial feasibility study for the system, specific applications were highlighted as areas where optical painting could be used. One application was the placement of background detail on models such as creating camouflage patterns on model tanks. This work was shown to several local enterprises, specifically highlighting the automation of the process. This would reduce the requirement for skilled hand painters for applications where other processes such as decals were not suitable. A local high-precision model company was interested in testing the optical painting system and in comparing the new system with their current hand painted components.

Their current system relied heavily on very skilled hand finishers applying paint and then relieving the finished surface with a variety of different techniques from chemical to manual sanding. The models were of a bespoke variety and detail was important. However, much of the work that was done was attempting to recreate a wood grain texture on the side of plastic components; the components being worked on were predominantly flat with defined edges. The current system was time intensive but produced high quality results, the company was forward thinking and used a selection of modern manufacturing techniques in the production of each model.

Initially work was carried out using the complete robotic tilt-table setup. However, early on, it was realised that the specific requirements of the company would not need the complexity of the complete system, therefore experiments were tried using a digital projector and the emulsion. These experiments proved to be successful so long as the exposure was increased, providing a cheaper and easier solution. This solution was viable as most of the components were flat and the high resolution of digital projectors was comparable to the requested resolution and quality. During experimentation, several components of the system were tested and the application of photographic emulsion was seen to be more difficult to the smoother resin surface and the resolution of the input image was found to be critical in achieving high-resolution results.



Initially when emulsion was applied to the surface of the smooth resin, a flaking effect was found to occur after exposure. This was caused, as the emulsion did not absorb into the surface in a similar fashion to ceramic materials. Further research found this to be a common problem and a variety of different solutions were available, though the most successful was known as *"subbing"* (139) which effectively means roughing the surface. This unfortunately was unacceptable as it would affect the quality of the finish and another solution would need to be found. Further work looked at other methods of sealing surfaces and different chemical solutions were considered. The most successful appeared to be nail varnish which sealed the emulsion after developing and produced a glossy finish similar to painted components (which could be reduced with sanding).

Initially the intention was to produce a complete system that used toning to produce the wooden shade. However, experimenting discovered that control over the colour was more difficult than initially thought; with small amounts of dye added, large changes could occur. It was then thought that using a black and white image with wood texture exposed onto a brown painted surface could yield the results that were sought, this proved to be partially true. The emulsion adhered to the painted surface significantly better and the wooden grain could be imitated. Though the depth of detail was not particularly high, it was decided that further work would be required before implementation. One of the processes that was successful, used templates to expose characters on the surface of models in matt black. This was based on the ideas used in car detailing where masking the area surrounding the detail enabled full light exposure (i.e. complete paint coverage) providing accurate character representation.