GENERATION OF EFFECTIVE SAND-BLASTING TRAJECTORY FOR AN AUTONOMOUS ROBOT IN STEEL BRIDGE MAINTENANCE

by

Tian Ran REN

submitted in fulfilment of the
requirements for the degree of

Master of Engineering by Research

Faculty of Engineering
The University of Technology, Sydney

31 March 2008

CERTIFICATE OF AUTHORSHIP/ORIGINALITY

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Production Note:
Signature removed prior to publication.

Tian Ran REN

Abstract

Steel bridges are vulnerable to corrosions, which results in conditions demanding regular maintenance in terms of de-rusting and re-painting. Current practices mostly rely on human workers with manually operated sand-blasting equipment to remove the rust or paint. This approach is labour intensive, tedious and, most of all, causes health and safety hazards for the workers, due to toxic dust arising from the removed lead or asbestos-based paints. Thus, an autonomous steel bridge maintenance system is very desirable, and the motion control of a robotic arm is identified as a key system requirement.

This thesis is concerned with studies on algorithms for generating an effective trajectory to be followed by an industrial robot arm used in sand-blasting. It is crucial in the context of productivity that the motion of the arm should follow a trajectory that aims to maximise the coverage of the blasted area and minimise the arm movements. The problem is challenging due to the changing environment underneath the bridge and the risk of colliding with obstacles. Furthermore, the trajectory generation process is complicated because of the many requirements imposed, such as minimum arm travel distance, minimum number of turns and minimum time to complete the blasting.

The problem is tackled in this research by beginning with an assignment of the blasting area, where a hexagonal coverage pattern is adopted to allocate blasting targets. The sequencing of blasting spots on the blasting surface, constituting the path to be

followed by the blasting nozzle, is determined through the use of a genetic algorithm as a sequence-finder for its applicability and flexibility in many engineering design problems. The order of blasting spots (that is, the path of nozzle) is then transformed to robot joint angles, that is, trajectory, by a genetic algorithm amended inverse kinematics approach. Furthermore, a method based on three-dimensional force-fields is used to safeguard the robot against collisions with obstacles. The resultant trajectory, in the form of a series of joint angles commands are fed to a Denso VM-6083D-W industrial robot for sand-blasting.

The effectiveness of the generated trajectory is verified by simulations and experiments. It is shown that trajectories can be derived for blasting surfaces with satisfactory coverage. The developed method is further demonstrated in generating trajectories for a number of blasting surfaces of different sizes, to the extent of the work space, at various locations and orientations surrounding the robot arm. An experiment is conducted, on a mock-up robotic blasting system, by driving the robot arm in accordance with a generated trajectory.

Acknowledgements

The first person to whom I would like to give my sincere thanks is my respected supervisor, Associate Professor Dikai Liu. I feel very thankful to him for his wise guidance, patience and perseverance through my Master's candidature, and also for his providing me with this opportunity to continue my postgraduate study.

I would like to thank my co-supervisor, Dr Shoudong Huang, for his many valuable discussions shared with me, his encouragement and enlightenment given to me during some most difficult times.

I must thank Professor Gamini Dissanayake, Director of the Centre of Excellence for Autonomous Systems at the University of Technology, Sydney (UTS), who has provided me with the opportunity to pursue studies in robotics. Here, I would also like to thank Professor Hung Nguyen and Ms Phyllis Agius for their kindness in providing me with administration assistances. Thanks also to Pat Skinner for proofreading this thesis.

I often feel grateful to Dr Ngai Ming Kwok for his generous assistance in the process of my study. It is he who shared his efforts and guided me through my research course. I must say that I have spent a very exciting time during my Master's candidature at UTS because of the great friendship built up between my colleagues. In particular, I would like to thank Tom Chotiprayanakul for his help in conducting experiments, Dalong Wang for his friendship, Gavin Paul, Nathan Kinchner and Matthew Clifton for

their insightful discussions.

I would like to thank my families as it is their financial and spiritual support that have made it possible for me to continue my study at UTS. I thank my mother, for it is her great love that supports me through the most difficult times. I thank my father, for it is his humorous encouragement and faith in me that prompt me to look beyond the obvious and to keep the creative spark burning. I also thank my husband, Nengguang, for his tolerating my bad temper during times of struggle, and always sharing my hardships in life and work. It is also my family's encouragement that helps me overcome many difficulties and continue my study with great determination and strength.

Finally, in a word, I should thank all of those who have helped me. It is they and their help that encouraged me to complete my candidature. I am very happy for having those people who are so kind to me. They really do great favour not only to my paper writing but also to my daily study. It has been my great honour to meet them and work together with them.

To my father and mother

Contents

Abstract ii

Acknowledgements iv

1 Introduction 1

1.1 Steel Bridge Maintenance 3

1.2 Automatic Sand-blasting System Overview 4

1.3 Motivation 7

1.4 Objectives 9

1.5 Methodology 10

1.6 Publication 11

1.7 Thesis Outline 11

2 Background and Related Work 13

2.1 Industrial Robots in Infrastructure Maintenance 14

2.2 Trajectory Generation 15

2.2.1 Coverage Problem 16

2.2.2 Tool Path Planning for Robotic Spray Painting 18

2.3 Robot Manipulator Control 24

2.4 Genetic Algorithms in Path Planning 26

2.5 Summary 28

3 Trajectory Generation for Robotic Sand-blasting **29**

3.1 General Framework of Trajectory Planning 30

3.2 Blasting Nozzle Model and a Hexagon-based Topology Pattern for Complete Coverage 32

3.2.1 Blasting Nozzle Model 32

3.2.2 Hexagon-based Topology Pattern for Complete Coverage of a Given Surface 35

3.2.3 Editing of Disc Location on the Boundary 37

3.3 Generate the Sequencing of Blasting Spots 40

3.3.1 Design of Objective Function 40

3.3.2 Genetic Algorithm-based Path-Searching 41

3.4 Robot Motion Planning 43

3.4.1 Genetic-Algorithm-amended Inverse Kinematics for Determining Robot Configurations 44

3.4.2 Collision Avoidance 48

3.5 Summary 52

4 Simulation and Experimental Results **54**

4.1 Overview of Simulations and Experiments 55

4.1.1 Physical Environment 55

4.1.2 Simulated Environment 55

4.1.3 Experimental System Architecture 59

4.2 Simulations 60

4.2.1 Coverage of Irregularly Shaped Surfaces Using Hexagon-based Pattern 63

4.2.2 Editing of Blasting Spots 63

4.2.3 Sequencing Blasting Spots 63

4.2.4 Generation of Robot Arm Trajectory without Collision Avoidance Functionality 68

4.2.5 Generation of Robot Arm Trajectory with Collision Avoidance Functionality 77
4.3 Experiment 90
4.4 Summary 98

5 Conclusion **99**

Bibliography **102**

List of Figures

1.1 An example steel bridge structure (a), spots of corrosion underneath the steel bridge (b) and (c). 3

1.2 A sealed sand-blasting cell (scaffold) (a), protection gear (b) and manual sand blasting process (c). 5

1.3 Laboratory setup. 6

1.4 Overview of the sand-blasting system. 7

1.5 General framework of the sand-blasting system. 8

2.1 Sponsored coverage calculation-basic model [21]. 18

2.2 Different turn-off situations [21]. 19

2.3 A spray painting path [13]. 20

2.4 Properties of the elliptical spray area [26]. 21

2.5 The number of turns is the main factor in the cost of covering a region along different sweep directions. 23

2.6 Complete coverage paths. The dark rectangles are stationary obstacles [32]. 24

3.1 The optimal trajectory planner. 31

3.2 The nozzle model. 33

3.3 Nozzle Boride T159. 33

3.4 Nozzle characteristics with different stream length, (a) blast radius, (b) blast area. 34

3.5 The process of generating hexagon topology. 36
3.6 Covering a free-form surface using a hexagon topology. 37
3.7 Discs classified into three groups. 38
3.8 Best fitting line. 39
3.9 Boundary editing method. 39
3.10 Parameters of DENSO VM-6083D-W robot arm 46
3.11 Process of generating joint angular velocities 49
3.12 Parameters of D_{min} and D_{max} ellipsoid (a) and a robot arm covered by D_{min} (b) 51
3.13 Flowchart for generating collision-free trajectory. 53

4.1 Typical structure below a steel bridge. 56
4.2 Mock-up structure mimicking the environment underneath the steel bridge. 56
4.3 Dimension of the I-beam. 57
4.4 Emulated simulation environment underneath the bridge deck. 57
4.5 Dimension of different surfaces, (a) part of Surface 1, (b) part of Surface 2, (c) part of Surface 3. 58
4.6 Components of the industrial robot arm: (a) the DENSO VM-6083D arm, (b) motor controller, (c) teach pendant. 59
4.7 Robotic sand-blasting system architecture. 61
4.8 Software configuration of the robotic sand-blasting system. 62
4.9 Test for coverage of irregular shaped surfaces, (a) trapezoidal, (b) circular, (c) arbitrary shape. 64
4.10 Results of editing the blasting spot locations, (a) overall edited result, (b) bottom boundary, (c) top-left boundary. 65
4.11 Sequenced blasting spots, (a) minimising travel distance, (b) minimising travel distance and the number of turns. 67

4.12 The robot travels from the initial position to the first target blasting spot, (a) initial position, (b) and (c) intermediate positions and (d) blasting start position. 69
4.13 The robot motion while blasting a small area after following the planned path, (a) start of blasting, (a) and (b) intermediate positions and (c) end of blasting position. 70
4.14 Joints angle from the initial robot position to the first blasting spot. . . 72
4.15 The joint angles traced while the robot travels along the generated path. 72
4.16 Traces of joint movements with movement time extended. 73
4.17 The robot motion on Surface 1, following the generated path. 74
4.18 The robot motion on Surface 2, following the generated path. 75
4.19 The robot motion on Surface 3, following the generated path. 76
4.20 Comparison of sequencing results due to the presence of obstacles (a) without obstacle, (b) with obstacle. 78
4.21 The simulation result of the robot conducting sand-blasting of Surface 1 by following the generated path in the complex steel bridge environment 80
4.22 Distance between the force field ellipsoid and the I-beam while blasting on Surface 1. 81
4.23 The simulation result of the robot conducting sand-blasting of Surface 2 by following the generated path in the complex steel bridge environment 82
4.24 Distance between the force field ellipsoid and the I-beam while blasting on Surface 2. 83
4.25 The simulation result of the robot conducting sand-blasting of Surface 3 by following the generated path in the complex steel bridge environment 84
4.26 Distance between the force field ellipsoid and the I-beam while blasting on Surface 3. 85
4.27 The robot blasting a part of Surface 1 in the environment with obstacles. (1) 86

4.28 The robot blasting a part of Surface 1 in the environment with obstacles. (2) 87
4.29 The robot blasting a part of Surface 2 in the environment with obstacles. (1) 88
4.30 The robot blasting a part of Surface 2 in the environment with obstacles. (2) 89
4.31 The experimental results for robot path planning (1). 91
4.32 The experimental results for robot path planning (2). 92
4.33 The experimental results for robot path planning (3) 93
4.34 The experimental results for robot path planning (4) 94
4.35 Comparing between simulation and experimental results for robot path planning (1) 96
4.36 Comparing between simulation and experimental results for robot path planning (2) 97

List of Tables

3.1 Parameters of the nozzle (Boride T159). 35

4.1 Parameters of the genetic algorithm used in the simulations 66
4.2 Summary of blasting spot sequencing results 68
4.3 Comparison of blasting spot sequencing results (with obstacles) 79

Chapter 1

Introduction

Steel bridges need to be repainted for a number of reasons. One important reason concerns protecting the bridge from corrosion. The application of coatings of paint could protect a steel bridge, particularly from corrosion, through its life span. Wynch Bridge, the first iron bridge in England, was built over the River Tees in 1741 without a coating of paint. The bridge collapsed after 60 years because of the corrosion of support chains [1]. A cast-iron bridge, at Coalbrookdale in Shropshire, England, was not painted when it was built in 1779. It still exists today after paint was applied in 1788 and repainted regularly [2].

The common steel bridge painting process involves sand-blasting the steel surface to remove the current coating and rust, and then repainting the bridge to protect its surface from further corrosion. The procedure in sand-blasting usually relies on manual operation of the blast nozzles by human workers in a sealed environment, which is associated with health and safety risks. The toxic lead/asbestos contaminated old paint is a significant issue in causing harm to the worker's health [3]. Due to the health and safety risks associated with the sand-blasting processes, an automatic robotic system for sand-blasting is in great demand.

Industrial robots usually consist of a mechanical arm controlled by a computer which is able to perform tasks that are dull, dirty and/or dangerous. Robots increase production capability, improve product quality, and lower production costs. Thus, it is also economically valuable to build a robotic sand-blasting system to perform the required task automatically.

Trajectory and motion planning of the robot end-effector is one of the key issues in an automatic robotic system. In most manufacturing applications, robotic arms are deployed for repetitive operations in which trajectory planning can be conducted off-line. The environment in a work cell can be structured as obstacle-free, such that trajectory planning could be simplified. However, these favourable conditions are not always true for sand-blasting a steel bridge, and thus engineering challenges are foreshadowed.

This study is part of a joint project between the University of Technology, Sydney (UTS) and the Roads and Traffic Authority (RTA) of New South Wales, Australia. The project is aimed at developing an automatic robotic system for steel bridge maintenance, particularly, in removing paint from bridge structures using sand-blasting. This thesis addresses the problem of trajectory planning for the robot arm used in the maintenance system.

The rest of this chapter is organised as follows. A brief account of steel bridge corrosion condition and the basic method of removing the rust and paint are given in Section 1.1. Section 1.2 contains an overview of the sand-blasting system. Then the motivation is presented in Section 1.3. Objectives of this study are given in Section 1.4. The methodology adopted is presented in Section 1.5 and a publication title based on the outcome of this work is given in Section 1.6. The rest of the thesis is outlined in Section 1.7.

(a)

(b)

(c)

Figure 1.1: An example steel bridge structure (a), spots of corrosion underneath the steel bridge (b) and (c).

1.1 Steel Bridge Maintenance

In Australia, there is a large number of steel bridges at critical levels of deterioration. They need to be maintained regularly for the sake of prolonging their service. A typical environment underneath a steel bridge is illustrated in Figure 1.1(a). Figures 1.1(b) and 1.1(c) show the rusted patches commonly found.

Maintaining corroded steel bridges presents a major challenge in terms of monetary involvement, occupational health and safety issues and the demand for advanced

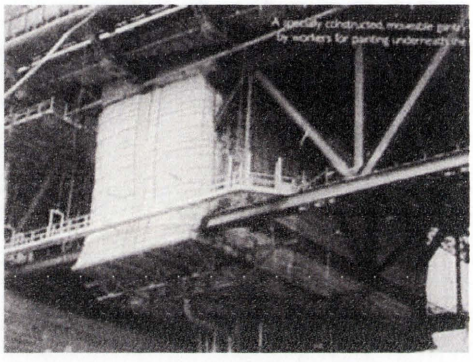
technologies. The program taken to maintain the steel bridge in practice is a very complicated and expensive process. This is because there is not only a large percentage of such steel bridges requiring maintenance, but also the problem of protecting the workers in this kind of hazardous environment which is full of dust containing lead/asbestos.

The first step in repainting the structure of a steel bridge is to remove its current coating by conducting a high-pressure sand-blasting. The second step is to repaint the processed surface within a few hours. The process must be executed in a sealed environment (see Figure 1.2(a)), because the blasting operation produces asbestos-based dust which will undoubtedly pollute the surrounding environment otherwise. In the context of the workers' safety and health, they must be protected with suitable clothing and breathing apparatus (see Figures 1.2(b) and 1.2(c)). Consequently the bridge maintenance work carried out by human workers becomes time consuming and also extremely physically demanding.

Several semi-automated sand-blasting systems have been developed in the past few years [4]. A common feature is that the choice of a trajectory for sand-blasting relies upon human control and the resultant performance critically depends on the operator's skill. On the other hand, the toxic lead/asbestos-contaminated particles cause significant health issues when cleaning a steel bridge. Furthermore, due to the health and safety risks associated with the sand-blasting process, an autonomous robotic arm system working in such environments is highly desirable.

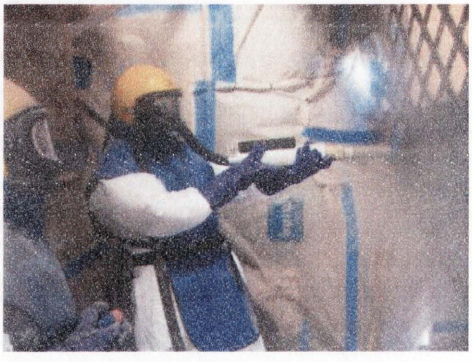
1.2 Automatic Sand-blasting System Overview

Since 2006, we have been developing an robotic system for steel bridge maintenance, supported by the Centre of Excellence for Autonomous Systems (CAS), the Roads and Traffic Authority (RTA) and the University of Technology, Sydney (UTS). As shown



(a)

(b)



(c)

Figure 1.2: A sealed sand-blasting cell (scaffold) (a), protection gear (b) and manual sand blasting process (c).

in Figure 1.3, the robotic system consists of an industrial robot, a moving platform, a control computer and a sensor unit. In practice, the system will be deployed in an environment underneath the bridges to be maintained. Tracks will be placed along the channels on the scaffold underneath the bridge deck. A robot arm equipped with a sand-blasting nozzle is mounted on a base platform. A laser range-finder is attached on the end-effector and is used to identify the surrounding environment. The sensed data is then converted to represent the environment by a set of point clouds as shown in Figure 1.4 [5].

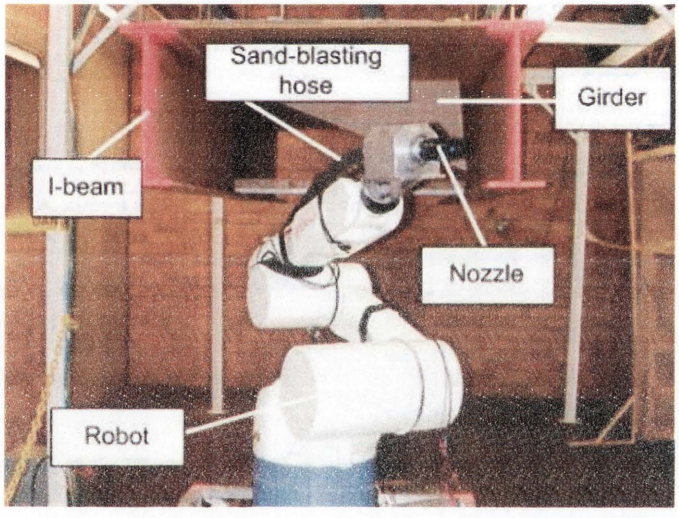


Figure 1.3: Laboratory setup.

When blasting is conducted, the nozzle is mounted on the end-effector in place of the laser range-finder. The nozzle is fed with pressurised air mixed with sand grit and its motion is governed by a generated trajectory. The process of sensing and blasting is repeated until the desired surface is completely cleaned. The architecture of the robotic steel bridge maintenance system is shown in Figure 1.5. It consists of (1) mapping of a maintenance environment. (2) identification of material types, in order to protect non-steel parts in a bridge. (3) surface reconstruction which generates the desired blasting

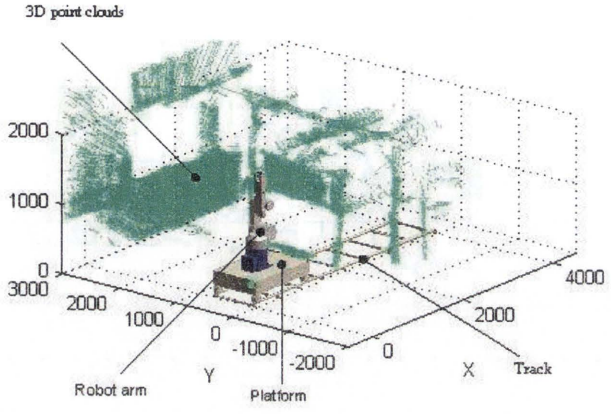


Figure 1.4: Overview of the sand-blasting system.

surface based on point clouds obtained from mapping. (4) path and motion planning which generates efficient paths automatically for a robot arm to move from one position to other position following the planned paths (5) collision avoidance which ensures the robot to conduct blasting task with collision-free path in the complex environment under steel bridges (6) a human-robot interface and (7) a robotic sand-blasting system control.

1.3 Motivation

In order to effectively conduct steel bridge maintenance, it is required to develop a fully automated robotic sand-blasting system with an effective path planning approach. Unlike most industrial applications of robotic manipulators in a pre-determined and obstacle-free workspace, a number of challenges in path and motion planning needed to be met in this research, and hence, have motivate this work.

These challenges are highlighted as follows:

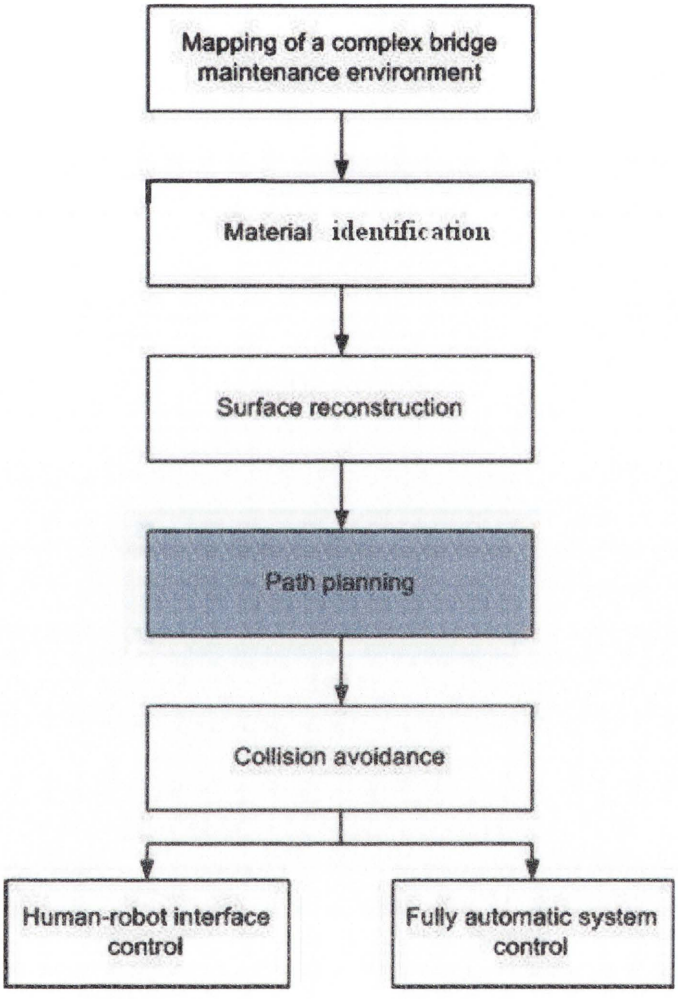


Figure 1.5: General framework of the sand-blasting system.

1. The path followed by the robot arm must be automatically generated for a particular bridge surface obtained by sensing the environment.

2. The planning algorithm should be sufficiently generic and cater for changing environments found in steel bridges.

3. With a fixed position of the robot, the blasting area should be completely covered by arm travel.

4. The powerful high-pressure sand-blasting streaming is not allowed to target outside the surface to be blasted.

5. Collision-free trajectories are mandatory because the robot moves in a complex structured environment.

6. Generation of the trajectory should accommodate objectives imposed on the processes such as minimum travel distance, minimum number of turns and minimum time to complete the blasting on a surface.

1.4 Objectives

To address the challenges presented in Section 1.3, this research aims to:

1. Formulate an appropriate topology for completely covering the blasting area without gaps.

2. Develop a boundary editing algorithm which could ensure that the blasted areas are maintained within the designated surface boundaries.

3. Formulate an efficient algorithm which can find the collision-free trajectory in complex bridge maintenance environments.

4. Characterise the behaviours of evolutionary optimisation approaches for robotic path planning.

5. Demonstrate and evaluate the effectiveness of the proposed approaches and algorithms through simulations and experiments.

1.5 Methodology

Through theoretical and practical research, this project aims to design and implement efficient robot end-effecter path/motion planning methodologies for sand-blasting irregular surfaces. In practice, steel bridges have complex structures which may contain a large number of irregular surfaces. Under the constraints of coverage and robot pose selection, the robotic path planning problem contains conflicting requirements, hence it demands a compromised approach. Enhancements to the path planning process will also be attempted with the use of evolutionary computation algorithms.

For robot motion planning/pose selection, inverse-kinematics is adopted to calculate the robot joint angles based on the desirable end-effector configuration. When singularity occurs a genetic algorithm is employed to find the joint pose that produce the least position and orientation error of the end-effector from desirable configurations. For safety issues, a collision avoidance algorithm is adopted to obtain collision-free trajectories.

The focuses of this thesis are summarised below:

1. *Incorporation of Hexagonal Topology-based Pattern*
   In order to maximally cover the blasting surface by the sand stream, the construction of a hexagonal coverage pattern is developed.

2. *Development of an Editing Algorithm for Free-form Surfaces*
   The editing algorithm is developed to automatically organise the surface boundary to make sure that the area to be blasted is not outside the blasting spots, while

no area which should not be blasted appears inside the boundary.

3. *Implementation of a Genetic Algorithm for Robot Arm Trajectory Generation*
   A specific requirement for the robot arm trajectory in sand-blasting is that blasted areas should not be blasted again. This leads to the development of a genetic algorithm implementation with emphasis on repairing chromosomes in order to mitigate any duplicated blasting spots.

4. *Simulation and Experimental Studies*
   Simulations and experiments are conducted to verify the effectiveness of the robotic path and motion planning methods.

1.6 Publication

Outcomes of the studies carried out are disseminated in the following publication.

T. R. Ren, N. M. Kwok, D. K. Liu and S. D. Huang, "Path Planning for a Robotic Arm Sand-blasting System," *Proceedings of the IEEE International Conference on Information and Automation*, June 2008, Hunan, China (accepted March 2008, to appear).

1.7 Thesis Outline

After presenting the steel bridge maintenance problem, the motivation, the objective and the focus of this work, the rest of this thesis is arranged as follows. The relevant background to the application of a robotic system to perform sand-blasting will be reviewed in Chapter 2; topics on robot arm trajectory planning and its optimisation are also included. Chapter 3 contains the core development in this thesis. After presenting a practical sand-blasting model, a hexagon-based topology pattern is developed to completely cover the area to be blasted. A boundary editing algorithm is designed to organise the blasting spot on the boundary. A genetic algorithm is employed to obtain

the sequencing of blasting spot. The preferred trajectory is generated by the genetic algorithm modified inverse kinematics method. A collision avoidance algorithm is adopted to ensure the trajectory is collision-free. Simulation and experimental results will be presented in Chapter 4 to illustrate the effectiveness of the proposed methods. These tests include path planning, using: 1) different settings of the trajectory generation algorithm, 2) various shapes of irregular surfaces, 3) changed surface size and 4) real-life robot arm motion control. Finally, a conclusion is drawn in Chapter 5.

Chapter 2

Background and Related Work

Robotic systems have been attracting considerable attention in various application areas. For instance, robotic manipulators are very suitable for the automatic maintenance of infrastructure such as steel bridges. In order to ensure its satisfactory performance, it is crucial to generate an effective trajectory for the manipulator to follow. Hence, a rationalised trajectory planning strategy is required, such that the motion of the arm can be controlled by deriving a sand-blasting sequencing of joint angle commands. Furthermore, real-world constraints imposed by the problem domain, such as minimum travel distance, joint angle change and obstacle avoidance need to be properly addressed. To this end, optimisation techniques are frequently incorporated to design the desired trajectory. The automatic robotic system for steel bridge maintenance to be developed in this thesis falls within the scope of the robot system described above. In this chapter, previous related work in these areas is reviewed and serves as a background for the subsequent developments.

Section 2.1 of this chapter will provide examples of different types of robotic paint removal systems under development and practical milestones achieved. Section 2.2 will provide information regarding related research work in trajectory generation. In Section 2.3 the control of robot manipulator is briefly presented. In Section 2.4, the application

of genetic algorithms in solving optimisation problems are reviewed.

2.1 Industrial Robots in Infrastructure Maintenance

There have been substantial advancements in using automated and robotic systems in infrastructure construction over recent decades. Systems have been developed [3][4][6][7][8] to improve quality, productivity, safety and to reduce cost. Robotic systems are envisioned to be able to conduct surface processing without very complex robot motions [9].

In Japan, a surface-finishing [10] robot was developed for wall painting in 1989. Researchers from the University of Texas at Austin developed an automated machine system for use in the petrochemical industry, which was designed to blast and paint large-diameter steel storage tanks. This prototype design was completed in 1990. A surface cleaning (current paint removal) system has been developed by several other research groups. The National Aeronautics and Space Administration/United Technology project has embarked on the development of an advanced stripping system - U.S. Navy/High Pressure Water Jet Paint Removal and Recovery System. This system is based on high-pressure water-jet cleaning to clean large ships, barges, floating dry-docks and other vessels [10]. These robotic paint/removal systems are built for simple and large structures. However, the processes of generating a robot manipulator trajectory for complex bridge structures have not been widely addressed. Furthermore, the above systems are designed to work in open environments. On the contrary, the blasting system developed in this work is to be deployed in a closed environment where obstacles may be found.

An automated robotic system for rust/corrosion removal for bridges was developed

in the Construction Automation and Robotics Laboratory at North Carolina State University [11] jointly with the Federal Highway Administration and the North Carolina Department of Transportation. The removal system consisted of a four degree-of-freedom robotic arm equipped with a sand-blasting nozzle, a bridge inspection crane (Peeper crane) with a crane boom, and a containment collection system mounted on an actuated platform. An actuated platform was also built for positioning the robotic arm. This system was tele-operated.

The robotic system control in [12] is one important element of robotic removal systems. The human-robot interface provides a great advantage to the control of the robot system. In manual operation, relying on the visual images from camera and sensor data from ultrasonic distance sensors, a human operator could control the positioning by using joysticks. The goal of this automated operation strategy is to perform the paint removal work by pointing the end-effector at a desired position. It is obvious that this robotic paint removal system is a semi-automated sand-blasting system, which relies upon human control. The quality of the sand-blasting, therefore, depends on the skill of the human operator.

2.2 Trajectory Generation

Moon and Bernold [4] used two basic strategies for generating robotic trajectories catering for the demand on complete coverage of the desired processing surfaces. One is in the form of a staircase and the other is a spiral confined in a window frame. Different travel approaches lead to significant variations in the total distances travelled by the robotic arm. Based on their optimisation objectives, the staircaseś blasting path generally performs better than spiral form. On the other hand, when regularly shaped surfaces are not available, such as on steel bridges, these strategies cannot be directly applied. Thus, there is motivation to build a fully automated robotic sand-blasting

system, which could generate the path and motion of the robot end-effector in for steel surfaces processing task.

In a context equivalent to sand blasting, a considerable amount of research work has been conducted on coverage issues addressed in sensor network designs. The trajectory generation problem on tool path planning for robotic spraying has also been studied (see [12][13][14]). The surface quality is assessed by the paint deposition thickness arising from the application of spray painting, which is more demanding than the coverage problem. In the following section, a review of these related research works tackling the coverage problem will be provided.

2.2.1 Coverage Problem

The coverage problem is widely studied in several domains, for example, in sensor network. The need for coverage generally answers the question about the quality of sensor service (surveillance). The coverage area of a sensor could be modelled as a disc. The objectives of both a sensor network and sand-blasting are to cover the largest area by using a minimum number of individual areas denoted as discs. The principle to decide the sand-blasting positions is conceptually equivalent to that used in sensor network.

Huang and Tseng [15] presented an algorithm to decide whether every point in a given service area is covered by at least one sensor. In their work, they indicated the relationship between two sensing ranges. Based on this method, the insufficient coverage problem could be solved and the overlap of sensing ranges mitigated. Zhang and Hou [16] addressed the maintenance of sensing coverage and the connectivity problem. They solved this problem by minimising the number of sensor nodes belonging to a wireless sensor network. They also proved that the sensor communication range should be at least twice as large as its sensing range. Minimising energy consumption and prolonging the system lifetime have been considered by many researchers in wireless

ad hoc networks studies. In the article by Xu *et al.ś* [17], in order to reduce energy consumption, they considered a geographical adaptive fidelity (GAF) algorithm in *ad hoc* wireless networks. It is employed to conserve energy by dividing the whole region where the nodes are distributed into rectangular grids. After dividing, in the transmission range of each sensor, the distance between any pair of nodes in adjacent grids was maximised. Chen *et al.* [18] presented an algorithm SPAN, which is used to decide if a node should be operating or standing-by on the basis of the connectivity among its neighbours. GAF and SPAN are both required to perform local neighbourhood discovery. The coverage problem in sand-blasting tasks is also concerned with the process of organising the location of each neighbourhood blasting spot.

In dealing with the sensing coverage issues, several centralised and distributed algorithms have been proposed. Cerpa and Estrin [19] addressed the ASCENT algorithm in their article which could automatically configure sensor network topologies. The finding of the maximal number of covers in a sensor network has been presented by Slijepcevic *et al.* [20]. In their paper, the monitored area can be covered by a set of nodes; a heuristic solution for solving the coverage problem is provided. They demonstrated that the proposed algorithm approaches the upper bound of the solution in most cases.

Tian *et al.*[21] presented an algorithm which uses the concept of the *sponsored area* to meet the complete coverage requirement. The main objective of this approach is to minimize the number of working sensors, and maintain the original sensing coverage, as shown in Figure 2.1. A sensor node will calculate its sponsored area while receiving a packet from one of its working neighbours. Here, the sponsored area is defined as the maximal sector covered by the neighbour. The node will turn itself off while the union of all the sponsored areas of a sensor node, as shown in Figure 2.2. Since the area where the sand stream hits the bridge surface is relatively much smaller than the region to be covered, hundreds of sand-blasting spots need to be organised on each blasting surface. To this end, computational burdens may impose difficulties in the above algorithms.

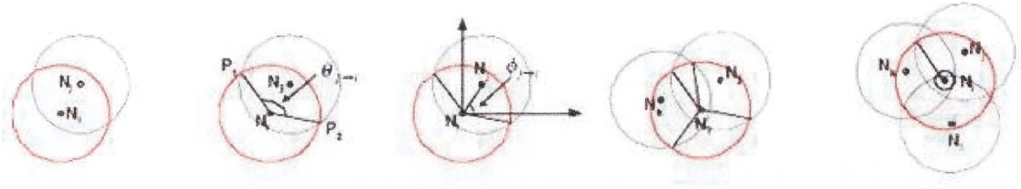


Figure 2.1: Sponsored coverage calculation-basic model [21].

Ye *et al.* [22] proposed the Probing Environment and Adaptive Sleeping (PEAS) method, a probing-based distributed density approach, for robust sensing coverage. PEAS can build a long-lived sensor network which lasts longer than that of individual nodes. The probing range can be adjusted to achieve different levels of coverage redundancy. The algorithm guarantees that the distance between any pair of working nodes is at least within the probing range, but does not ensure that the coverage area of a sleeping node is completely covered by working nodes. For a sand-blasting trajectory, completed coverage is one of the most important quality requirements.

2.2.2 Tool Path Planning for Robotic Spray Painting

There are many research works on tool path planning for robotic spraying ([12][13][14]) . The surface quality is assessed by the paint deposition thickness arising from the application of spray painting, which is more demanding than the coverage problem. For example, in Antonio's papers [23][24], a framework for optimal path planning is drawn to solve the thickness problem of optimal painting. Asakawa and Takeuchi put forward

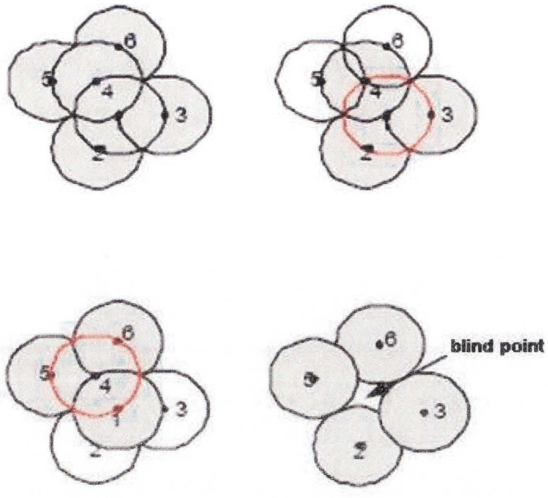


Figure 2.2: Different turn-off situations [21].

a teaching-less method for spray-painting of sculptured surfaces by an industrial robot in their article [13]. They took painting a car's bumpers as an example to present a sculptured surface. In their approach, the spray-painting gun moves along the curves interpolating the points, which are generated on the work-surface on the basis of available CAD data. After passing a spray-painting path element, the gun moves to start the position of the next tour. The spray-painting path is shown in Figure 2.3. During the painting process, the spray gun is kept perpendicular to the work-surface. However, the small radius of curvature leads to greater paint thickness at corners. At the edge of the work-piece, the shortage of the over-spray points has caused the gun to start painting before attaining the desirable speed for doing so.

Cheng *et al.* [25] pointed out that to achieve uniform paint thickness on the complex geometry of free-form surfaces is still a challenging research topic. They determined the path based on the paint thickness relationship of each neighbouring paths. The golden

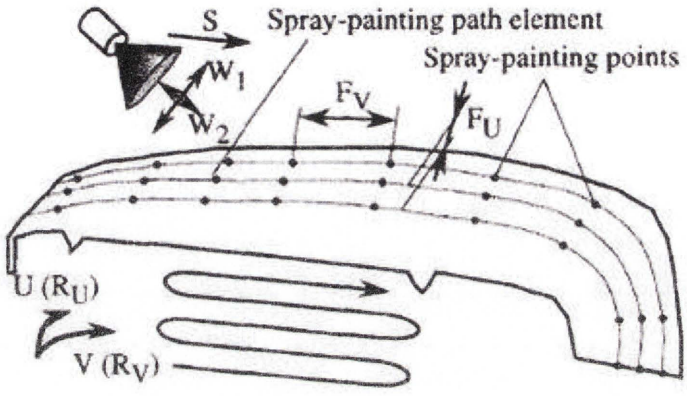
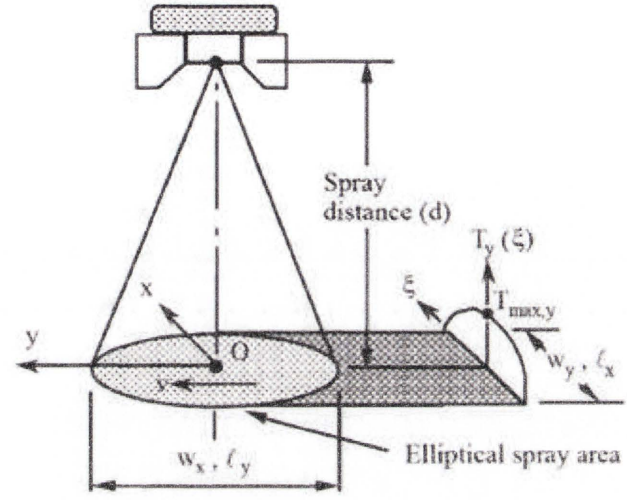


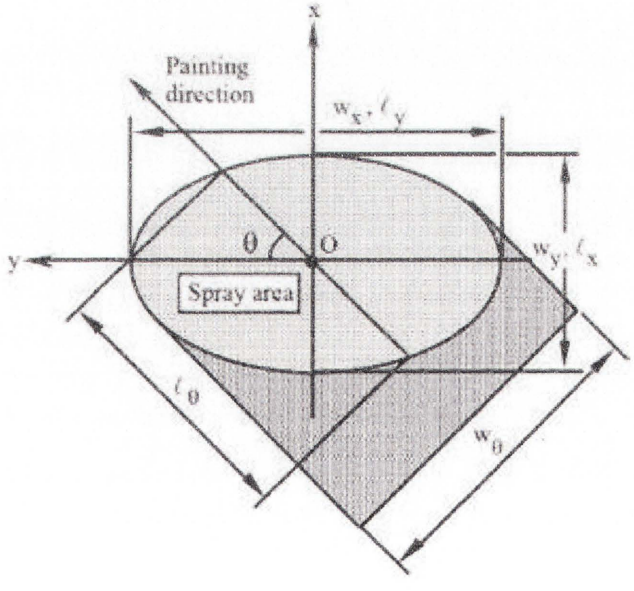
Figure 2.3: A spray painting path [13].

section method is used to find the optimal gun velocity and overlap distance by iteration. A recent research work, by Arikan *et al.* [26], mainly presents the elliptical spray areas generated by different painting strokes at different spray distances and painting velocities. The aim of their work is to present an experimental demonstration of a feasible paint flow-rate employed for elliptical paint spray applications. The authors, in their article, showed that in order to obtain a general solution and model a wider range of paint spray forms, the spray gun becomes the basic variables for predicting the thickness distribution on the painting surface in modeling the spray painting process. Spray areas are indicated as ellipses (Figure 2.4). However, the elliptical spray area is much more complicated to be calculated than the circular ones.

The tool motion must be taken into account to improve the manufacturing efficiency. The teaching of systems to generate a special path for automatic tools have been widely studied in the past decades. For example, Suh *et al.* [12] developed an Automatic Trajectory Planning System (ATPS) for painting robots. The approach to approximately represent the original free-form surface with small surfaces is adopted by



(a)



(b)

Figure 2.4: Properties of the elliptical spray area [26].

Suh *et al.* [12] in the trajectory planning stage. In [27], the approach decomposes the coverage region and determines a sequence of sub-regions, and then creates a coverage path that covers each region and moves to the next. During the process, the cost of the path generated has been taken into account the optimal line-sweep decompositions. The resultant trajectory length is approximately the total length, which is generated by the different sweep directions. However, the number of turns may cause increases in the operation times as illustrated in Figure 2.5, because the robot must slow down, make turns, and then accelerate. Thus, minimising the number of turns in proportion to the altitude of the sub-region will reduce the working time. This implies that generating a decomposition which minimises the sum of the altitudes is able to produce an optimal coverage path. The diameter of a polygon is determined by rotating the polygon and measuring the height as it rolls along the flat surface. A diameter function is generated to describe the altitude of the polygon along the sweep direction. This approach is closely related to the sand-blasting operation considered here; its design principle will therefore be adopted.

In other works such as [28][29][30][31], the authors also developed different methods to generate optimal paths for compound surfaces and free-form surfaces. Simulations were conducted based on a car's inner hood in these articles. Once the path planning is finished, it could be used for painting large numbers of such car's inner hoods. Thus, computational time in planning is not considered as a major problem.

In the mobile robot path planning domain, researchers are normally generate a robot trajectory on the basis of the pre-generated landscape. Gou *et al.* [32] studied complete coverage control for non-holonomic mobile robots in a dynamic environment by employing a neural network approach. They used the centre of a circle to indicate a mobile robot, the radius of the circle was used to represent the robot coverage region by its end-effector. The positions of stationary obstacles are described by coordinates in a pre-existing map of its operating environment. The speed and position of moving

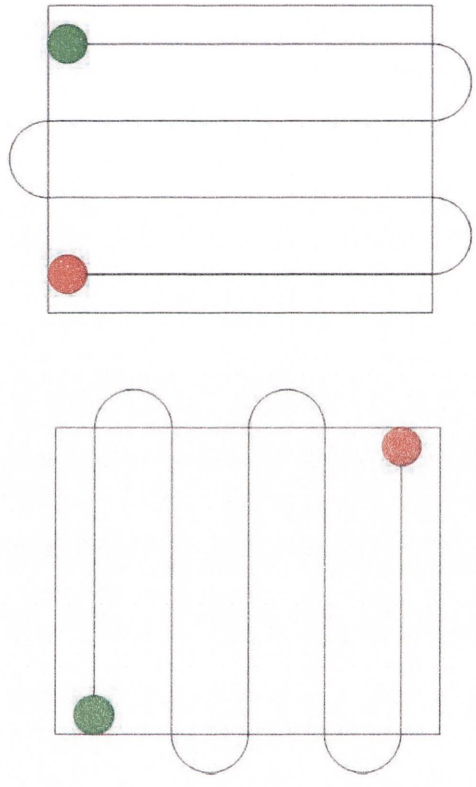


Figure 2.5: The number of turns is the main factor in the cost of covering a region along different sweep directions.

obstacles are detected by the robot's on-board sensor that is used to update the map. Afterwards, the robot path is automatically generated based on the location and landscape. As shown in Figure 2.6, the authors presented the completely covered path within the bounded region without covering a point twice. Although this method focuses on mobile robots, the design principle can be used in path planning for sand-blasting.

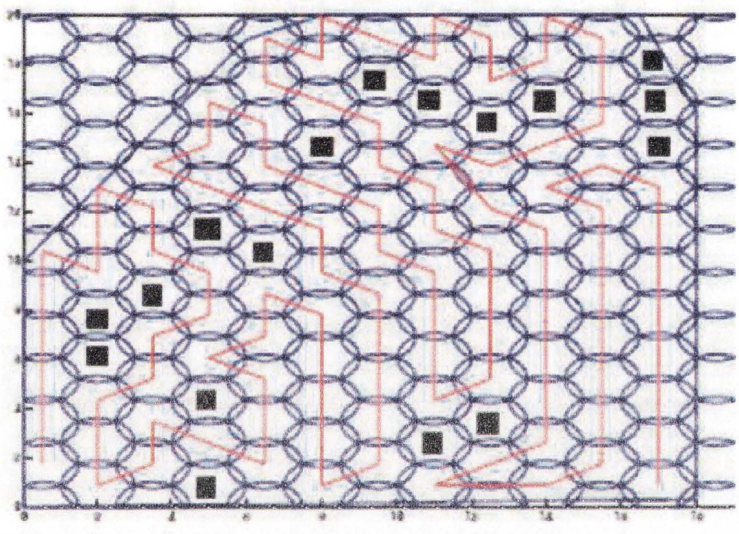


Figure 2.6: Complete coverage paths. The dark rectangles are stationary obstacles [32].

2.3 Robot Manipulator Control

The control of robot manipulators requires mapping from the end-effector positions to joint angles. An articulate figure is often modelled as a set of rigid segments connected with joints. Thus, altering joint angles can cause varying configurations. It is straightforward to calculate the configuration by given joint angles (forward kinematics). However, obtaining the joint angles for the desired configuration is more involved (inverse kinematics). Inverse kinematics is a common technique in robot manipulator

research [33]. In robotics, however, the greatest concern is with the functionality of manipulators, especially facing with practical tasks. Since the inverse kinematics problem is of special importance to manipulator control, researchers have proposed several different solution approaches. Korein and Badler [34] investigated methods for kinematic chain positioning, especially in the context of joint limits and redundant degrees of freedom.

Girard and Maciejewski [35] adopted an approach from robotics. To calculated the pseudo inverse of the Jacobian matrix that relates the increment of the joint angles to the displacement of the end-effector in space. Energy constraints were used by Witkin *et al.* [36] for calculating the position and orientation of the robot manipulator. Constraints are satisfied if and only if the energy function is zero. An isoenergy line is defined based on the joint angle space, on which the energy function takes identical values. Under this physical interpretation of the energy function, the method by Witkin *et al.* [36] searches the path from the initial configuration to the target configuration which is, at any point, perpendicular to the isoenergy lines. Instead of associating energy functions with constraints, Barzel and Barr [37] introduced deviation functions, which measure the deviation of two constrained parts. They discussed a variety of constraints such as point-to-point, point-to-nail, etc. and their associated deviation functions. A segment in a system of rigid bodies is subjected to both external forces, such as gravity, and constraint forces, which force the deviations to zero whenever they are found to be positive. Constraint forces are solved from a set of dynamic differential equations that require all deviations go to zero exponentially in a certain amount of time. An approach based on physical modelling and interpretation was also adopted by Witkin and Welch [38] on non-rigid bodies whose deformations are influenced by a number of parameters. To apply this method to articulated figures, a joint would be considered as a point-to-point constraint and added to the system as an algebraic equation. This poses some practical problems that render such solutions inappropriate to highly articulated figures. First, it is not unusual to have several dozen joints in a highly articulated figure, adding to the

number of constraint equations substantially. Second, a joint of an articulated figure is meant to be an absolute constraint: it should not compete with any constraint that relates a point on a segment of the figure to a point in space. Such competition often leads to numerical instability.

The computation of the reachable workspace of an end-effector could employ the inverse kinematics [39]. The author believed that the determinations of joint angles are less important than the answer to the questions of whether a spatial configuration could be achieved or what the overall shape of the work-space looks like. Due to the collision avoidance requirement, inverse kinematics can be augmented to achieve collision detection. Combining spatial and joint constraints into collision-avoidance motion planning is a fundamental goal in robotics, and a variety of exact and heuristic approaches exist. The problem is that the complexity grows exponentially with the number of degrees of freedom, making the process to obtain exact solutions less efficient.

2.4 Genetic Algorithms in Path Planning

Genetic algorithms (GA), as a search-based optimisation method, have been successfully employed in a wide domain of engineering applications. In particular, the GA is very attractive in obtaining near-optimal solutions when the solution scape is discontinuous such that gradient-based approaches cannot be directly applied.

Genetic algorithm mimics the evolution process of living species. Holland [40] modelled natural adaption in the algorithm that was motivated by the principle of "survival of the fittest", as stated in Charles Darwin's evolution theory. Goldberg further developed the algorithm for search and optimal engineering designs [41]. By coding potential solutions as a population of chromosomes, and operating through selection, crossover and mutation, the Schemata theory ensures that the fitness of the overall population

improves over iterations.

Advantageous characteristics of genetic algorithms include:

1. Coding of potential solutions enables the GA to be used for various problems.
2. Guided random search permits the GA to obtain global solutions through iterations.
3. In contrast to classical methods, GA uses a population of agents in the search for solutions. Therefore, it is suitable for parallel processing.

Applications of GAs in engineering have been widely reported in the literature. For example, Powell [42] has used a GA to solve design problems with nonlinear constraints. The optimal design of space structures was also realised by using the algorithm [43]. Furthermore, a bi-level optimisation problem was solved by using a modified GA [44].

In the context of robot manipulator motion control, the GA was applied to design a trajectory to be followed by a nozzle in the spray-forming process [28]. A trajectory was generated by using the GA for steering a six degree-of-freedom robot arm to perform three-dimensional cutting of work-pieces (see [45]). Liao [46] also used the GA to plan a path for a robot arm in metal bending.

The control of robot manipulators could be realised in the form of task (Cartesian) and joint spaces. The task space is frequently adopted to represent the location of the work-piece. On the other hand, joint space is concerned with the specification of the desirable joint angles with which the robot arm is configured. The genetic algorithm, because of its flexibility, has been applied in these domains. For instance, the GA was applied to design a trajectory in task space for a two-link robotic arm [47]. In [48], motion planning in the joint space was successfully implemented for a manipulator with three degrees-of-freedom. In a challenging environment with static and dynamic obstacles present in the work-space, the algorithm was able to derive a collision-free path for

a robot arm [49].

It is anticipated that, in the complex steel bridge environment, challenges in path planning would be frequently encountered. Because of its flexibility and effectiveness, the genetic algorithm is adopted in this study to generate a trajectory followed by the robotic arm in an automatic sand-blasting process.

2.5 Summary

This chapter has presented some research works related to path and motion planning. Due to the quality requirements of sand-blasting, the coverage problem has become an issue in trajectory planning. Thus, related research work about coverage algorithms on sensor networks is presented. This thesis is concerned with obtaining an effective blasting trajectory, therefore several optimal trajectory planning algorithms for robotic spray painting are reviewed. Furthermore, the related work in the manipulator control and optimization method are provided as well in this chapter. In the following chapter, the methodologies to solve practical robotic sand-blasting path planning problem will be proposed.

Chapter 3

Trajectory Generation for Robotic Sand-blasting

In Chapter 2, research work related to trajectory planning and coverage was described. In this chapter, the approach to devising a satisfactory sand-blasting trajectory will be presented. With regard to blasting coverage, a hexagon-based blasting pattern is adopted to provide a near-optimal solution, while reducing the un-blasted area. In real-world situations, surfaces to be blasted are rarely available in regular geometric shapes. A boundary editing procedure is therefore proposed which ensures that the blasted areas are maintained within the designated surface boundaries. A sequencing of sand-blasting spots will then be determined, which the robot end-effector is going to follow. Without *a priori* knowledge on feasible sequencing, a genetic algorithm is used as a solution searcher. A modified initialisation procedure is used and results in an efficiency improvement with a reduction in the number of iterations needed to obtain a feasible solution. Furthermore, a solution method for the inverse kinematic problem and a pose selection algorithm is employed to transform the robotic end-effector from Cartesian space coordinates into its corresponding joint configurations while carrying out the blasting task. Finally, a collision avoidance algorithm is employed to generate a collision-free trajectory.

In the rest of this chapter, Section 3.1 provides the general framework of optimal trajectory planning. Section 3.2 is concerned with the complete coverage problem. The design of the blasting trajectory is presented in Section 3.3. In Section 3.4, two approaches for robot motion control are proposed. The method of generating the collision-free trajectory is also demonstrated. Finally, a summary is given in Section 3.5.

3.1 General Framework of Trajectory Planning

A general framework for sand-blasting robot trajectory planning is formulated. Trajectory planning should take the following aspects into account: the nozzle position and orientation, collision avoidance and robot arm configuration constraints, *etc.* Figure 3.1 shows the architecture and flowchart of the robot trajectory planner.

It is assumed that knowledge of the environment, in the form of a point cloud, is available from the sensing and map building stage. The blasting nozzle model and robot arm model are given. Furthermore, the surface to be blasted has been defined, for example, by operator intervention or an automatic algorithm. The planner takes the following steps to plan a trajectory for the sand-blasting robot.

1. The extracted surface is firstly represented by blasting spots, in the form of discs, in a hexagonal pattern. The size of each disc is determined by the nozzle model.

2. A boundary editing routine is then conducted to ensure that no discs are located outside the surface. Note that the surface normally takes an irregular shape.

3. The path is generated by the path planning algorithm, taking into account short travel distances and the number of turnings.

4. While planning, the possibility of colliding with obstacles is considered. This step produces the satisfactory end-effector path.

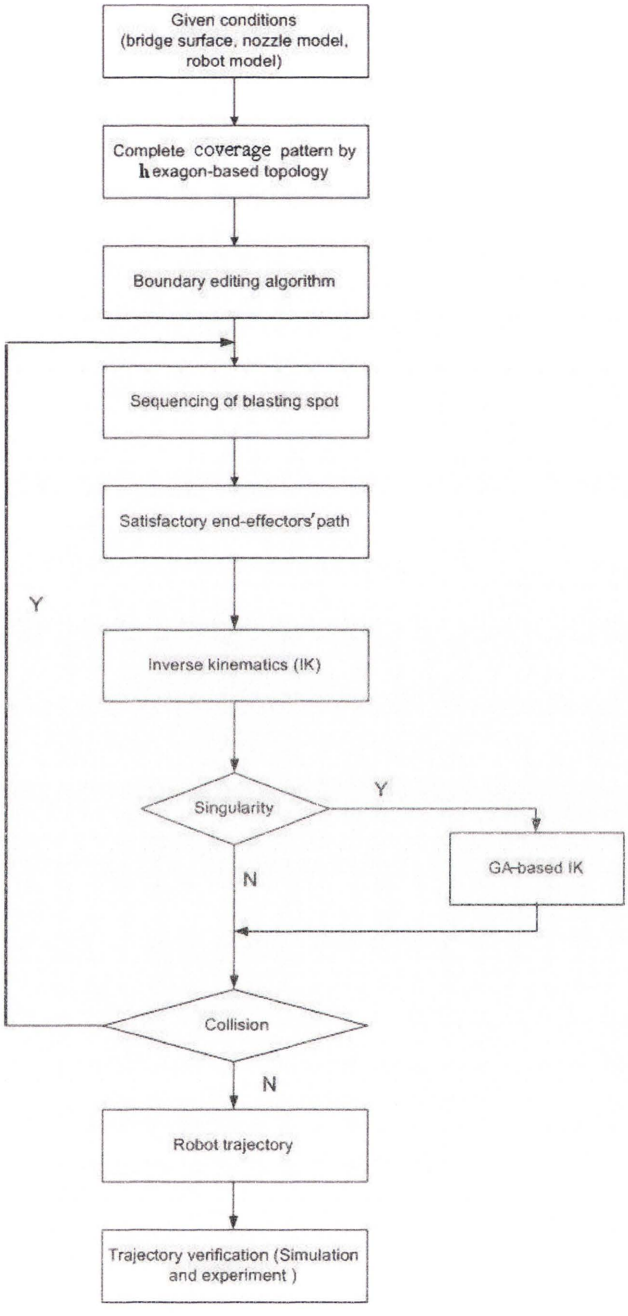


Figure 3.1: The optimal trajectory planner.

5. Based on the determined path, joint angle commands are derived by using the inverse kinematic approach. In case singularities occur, a genetic algorithm-based routine is invoked to search for feasible joint movements.

6. A satisfactory sequence of joint angle commands for the robot arm will be issued to drive the arm to conduct sand-blasting.

3.2 Blasting Nozzle Model and a Hexagon-based Topology Pattern for Complete Coverage

3.2.1 Blasting Nozzle Model

The path planning for complete coverage of an area to be blasted depends critically on the model of the blasting nozzle. The blasting nozzle can be modelled as a spray cone as shown in Figure 3.2, whereby a sand stream is emitted from the tool radially with a fan angle. A circular blasting pattern (denoted as a disc) is formed when the spray cone hits the bridge surface. The distance from the nozzle to the bridge surface is the stream length d_n. The radius of the blasting pattern is R_c, which is related to the stream length and the type of nozzle. The blasting direction or nozzle orientation can be varied from 45° to 90° is parallel to the nozzle axis and aligned with the normal of the surface.

Based on the specifications of the nozzle (Boride T159) shown in Figure 3.3, the blasting area, in the form of a circular disc, is given by

$$S = \pi R_c^2, \; R_c = d_n \frac{d_{max} - d_{min}}{2L} + \frac{d_{max}}{2} \tag{3.1}$$

where d_s is the stream length from the nozzle to the surface, L is the length of the cone within the nozzle with minimum and maximum diameters of d_{min} and d_{max}, respectively. The nozzle used in this work has three sizes, namely, small, medium and large

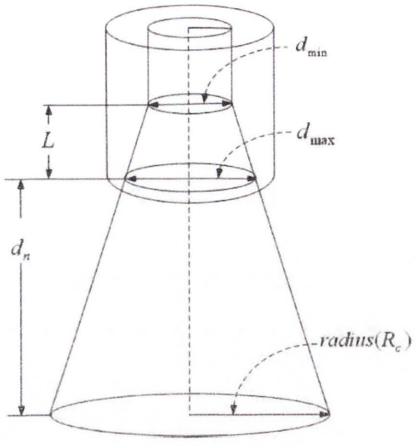


Figure 3.2: The nozzle model.

Figure 3.3: Nozzle Boride T159.

size, as the blasting task demands. The parameters of Nozzle-Boride T159 are given in Table 3.1. The size of the disc also affects the blast coverage. The blast radius and areas with different stream length are illustrated in Figure 3.4(a) and Figure 3.4(b). It is clear that d_s affects the size of the disc, the longer distance, the larger blasted area. However, due to safety and coverage quality requirements, d_s is chosen between $100mm$ and $500mm$.

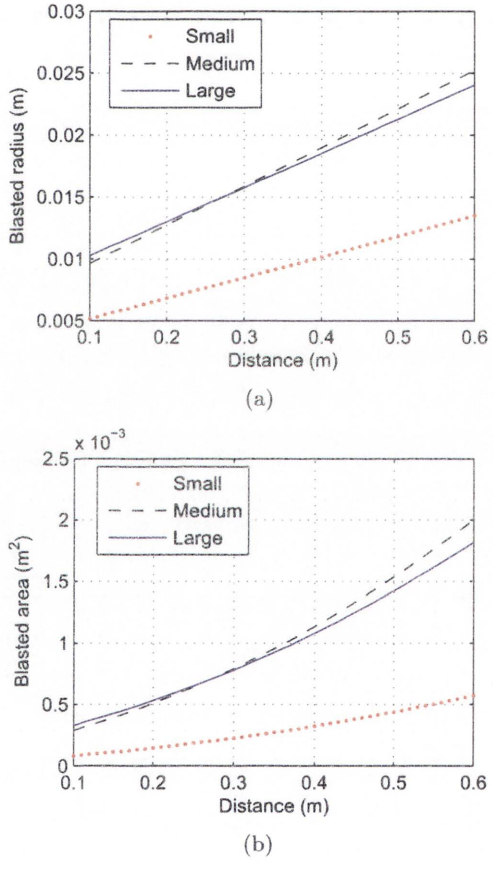


Figure 3.4: Nozzle characteristics with different stream length, (a) blast radius, (b) blast area.

Due to the need for determining an efficient sand-blasting trajectory, the process is

Table 3.1: Parameters of the nozzle (Boride T159).

Parameters(mm)	Small Size	Medium Size	Large Size
d_{max}	7	13	15
d_{min}	4.5	6	7
L	75	112	145

simplified by keeping the stream length constant. Thus, blasting discs with the same radius are chosen. The hexagon-based topology is adopted to cover the identified blasting area, which aims to minimise the *overlap* area without *"blank"* areas in the free-form surface.

3.2.2 Hexagon-based Topology Pattern for Complete Coverage of a Given Surface

The hexagon-based topology is selected to cover the desired blasting surface because, compare with square based, triangle based and other topologies, it is an efficient topology that is able to cover a large area without leaving uncovered areas. In this section, the procedure adopted to generate the hexagon-based topology pattern for covering given surfaces with a disc of radius R_c is presented in Figure 3.5. Then, the development of a boundary editing algorithm for free-form surfaces is presented. It is used to confine the blasting disc locations within the surface boundary. The procedure of generating this pattern is described below.

1. Since a hexagon-based topology is chosen for complete region coverage, the blasting region is firstly packed in the boundary region with discs of radius R_c, where R_c is obtained from the nozzle model.

2. The process then searches for a point P which is located on the near-centre of the blasting surface as an initial sampling point and a point Q that is $\sqrt{3}R_c$ to P is also selected.

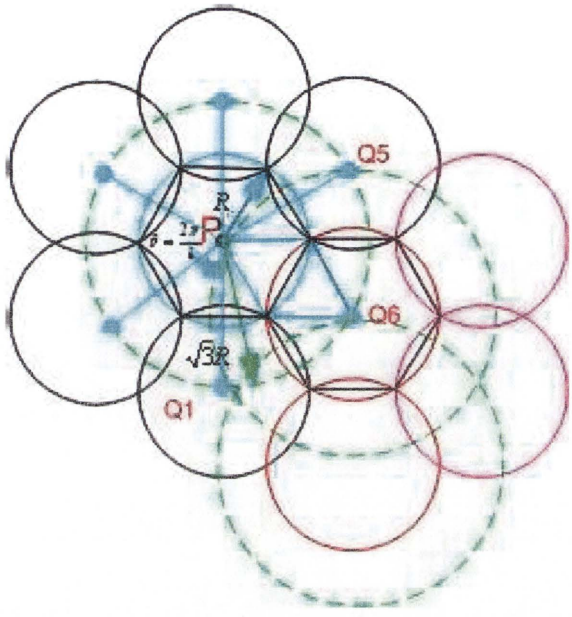


Figure 3.5: The process of generating hexagon topology.

3. Construct a circle with R_c as its radius at the near-centre of the desired blasting surface. Choose six points from Q_1 to Q_6 on the perimeter, with an angle $\pi/3$ apart, the starting point Q_i, $i = 1, \cdots, 6$ as its centre. Draw discs with radius R_c centred at Q_i.

4. Taking Q_1 to Q_6 as the centre, repeat the process, until the desired blasting-surface is completely covered. It is shown that the disk pattern in Fig. 3.6 has the near minimum number of disks to cover an irregular surface.

Figure 3.6 shows clearly that there are many blasting points located outside the boundary and on the boundary. The flow of high-pressure sand from the sand-blasting nozzle is very powerful, and could easily chip the wooden scaffold. Therefore, the region outside the identified blasting surface should not be blasted. In the following section, the methodology for adjusting the boundary blasting points to relocate them into the blasting surface will be presented.

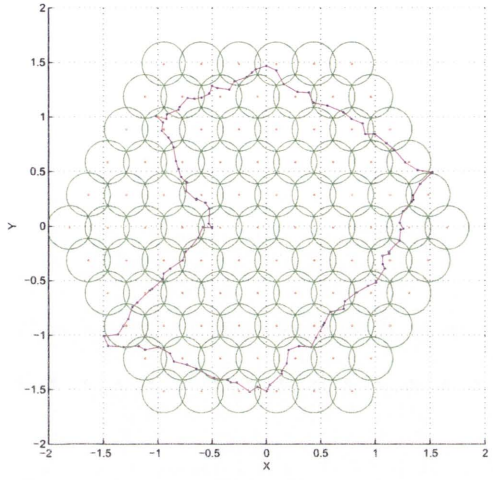


Figure 3.6: Covering a free-form surface using a hexagon topology.

3.2.3 Editing of Disc Location on the Boundary

Here, the boundary editing algorithm is used to ensure that there is no blasting disc beyond the boundary of the area to be blasted. The boundary region has been completely packed with discs of radius R_c, obtained according to the hexagon topology scheme. The first step in solving the editing problem is to identify the discs on the boundary and picot of the blasting region. A straightforward approach to the problem would be to classify the discs into different groups based on their distinctive characteristics. For each disc i, let

$$E_i = \begin{cases} 1 & \text{if it is inside the boundary} \\ 2 & \text{if it is outside the boundary} \\ 3 & \text{if it is on the boundary} \end{cases} \tag{3.2}$$

Where E_i indicates the position of each disc. Checking the distance from each discs centre to boundary points, which is denoted as D_p . Checking distance between discs

centre to near desired surface centre, which is denoted as D_s. Indexing the nearest boundary points to each disc, which is indicated as D_n. If $D_p < D_s$ and $D_n > R_c$, the disc is inside the boundary, set $E_i = 1$, if $D_n < R_c$, the disc is outside the boundary, set $E_i = 2$ and if $D_p > D_s$ and $D_n > R_c$, the disc is on the boundary, set $E_i = 3$. Obviously, the discs outside the boundary should be erased. The circles in light colours, as depicted in Figure 3.7, are the discs to be stripped. Due to the complete coverage requirement, boundary discs cannot be removed directly. They need to be *pulled* inside the boundary to avoid over-blasting. In order to avoid over-blasting and to minimise the overlap area, the desired position for the centre of the disc on the boundary is taken as the vertex of the circle connecting the boundary.

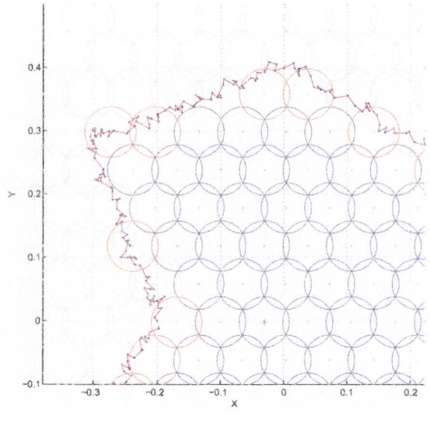


Figure 3.7: Discs classified into three groups.

The boundary line is generated from the point clouds. In practice, the linear squares fitting technique is employed to generate the best fitting lines which is the simplest and most commonly applied form of linear regression and provides a solution to the problem of finding a best straight line from a set of points. In Figure 3.8, the solid line is the best fitting line generated by the linear square fitting technique. The direction of moving the discs on the boundary is perpendicular to the best fitting line, shown in Figure 3.9,

moving P_1 to P_1'.

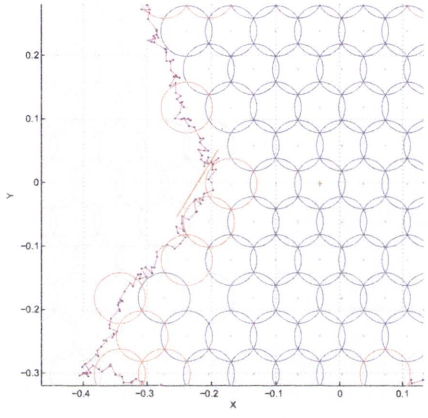


Figure 3.8: Best fitting line.

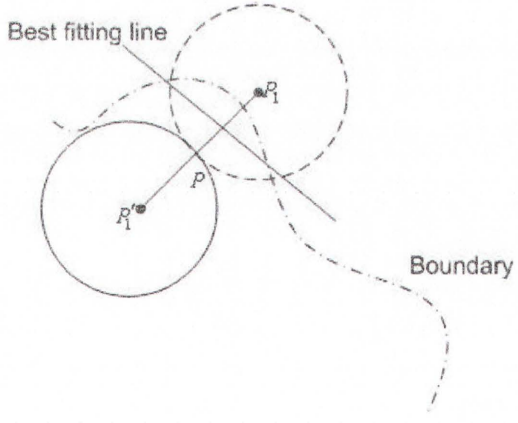


Figure 3.9: Boundary editing method.

After the whole working regions have been completely covered, the path for the robot arm can be generated based on performance criteria. In the next section, the genetic algorithm is employed to determine a blasting path.

3.3 Generate the Sequencing of Blasting Spots

A disc placement pattern has been generated with a near minimum number of discs to cover a free-form blasting area. In this section, complete coverage path planning is studied as a problem of finding the sequence of visiting the disc centres by the blasting stream. A genetic algorithm is employed and the algorithm is modified to generate paths on the blasting area.

3.3.1 Design of Objective Function

In this section, the greedy method is employed to generate a nozzle path for each step with minimal travel distance and turning angles, without collisions. Here, the greedy method is adopted because it is much more efficient then a global approach. Both nozzle sweep directions and the magnitude of turns are essential conditions that affect the sand-blasting efficiency.

The travel distance d_i can be expressed as

$$d_i = (P_i - P_{i+1}),\ i = 1, 2, \cdots, n-1 \tag{3.3}$$

where p_i and p_{i+1} are the current position and next position of discs picked randomly, and n is the number of discs.

To make a turn the robot manipulator must slow down, make turns, and then accelerate, thus, it is desirable to minimise the magnitude of the turns. Let θ_i represent the angle of a turn needed while moving from p_{i-1} to p_i then p_i to the next potential location $\hat{p}_{i+1}$. We select the next point among the six neighbours such that the cosine of the turn angle approaches unity, that is, $\cos(\theta_i) \rightarrow 1$ since $|\theta_i| \rightarrow 0$ is preferred. After the blasting stream reaches its next position, that position becomes a new current position.

The $\cos(\theta_i)$ function is further used as a penalty on d_i with the resultant travel distance modified as

$$d_i \leftarrow d_i(2 - \cos(\theta_i)) \tag{3.4}$$

Here, this objective function equation 3.4 is adopted because the trajectory is preferred with minimum travel distances and minimum turn angle.

3.3.2 Genetic Algorithm-based Path-Searching

In this section, the proposed algorithm to solve the path planning is described. Firstly, the genetic algorithm is briefly reviewed. Then a non-duplicative chromosome modification will be explained in detail.

Genetic Algorithm (GA) is an algorithm which is mainly guided by *Natural Selection* and *Evolution Theory*. Reproduction, crossover, and mutation are the basic mechanisms of GAs. Typically a GA works with coding a set of binary strings to search the discrete space. Holland [40] referred to these strings as *genotypes* and Schaffer [50] referred to them as *chromosomes* while choosing P strings each of length N as an initial population.

Chromosomes are selected from the population to be parents for operations such as crossover and mutation. According to Darwin's evolution theory, the best ones should survive and create new offspring. There are many methods for selecting the best chromosomes, for example, roulette wheel selection, Boltzman selection, tournament selection, rank selection and steady-state selection. Roulette Wheel Selection is employed to perform selection when parents are selected according to their fitness. The better the fitness is, the greater chance there is of being selected.

In 1989, Goldberg [41] described a simple GA implement with a population of encoded potential solutions. In most situations the initial population is generated randomly. After each string is evaluated and assigned, a population is created.

While doing crossover, the new set of strings is paired randomly. For each pair, crossover occurs with probability P_c. After crossover, mutation occurs. In this stage, some bits of a string are independently changed from zero to one or vice versa in the new population with probabilities. A genetic algorithm, through the above procedures, could find the best fitness values of individuals and combine them to produce individuals that offer better fitness values than their parents by Selection, Cross-over and Mutation operations. This process continues until the population converges to the best fitness value, or the number of generations is reached.

The steps for using a genetic algorithm in the path planning process are explained below. Integer coding is employed to represent the individuals and evaluate the solution based on the fitness landscape.

1. Initialisation: Let the population be P (i.e., paths, the length of chromosome equals the number of blasting discs of a desired blasting-surface). Conventionally, each chromosome is built from choosing randomly from the discs that correspond to the complete coverage path planning as

$$Chromosome = [Dsk_n],\ n \in [1, \cdots, N] \tag{3.5}$$

where Dsk indicates a disc, n is the index of the disc, N is the total number of discs. To avoid the bits (disc) of a chromosome from repeating, the repeated bits of a chromosome are repaired by randomly replacing the repeated discs with the missed ones. The complexity of the problem is related to the number of discs. For a large number of discs, more chromosomes are needed in the population. After

the generation of P chromosomes, their fitness is evaluated according to equation 3.4.

2. Selection: Chromosomes are sorted in ascending order according to their fitness value (minimum is preferred), each one has its place scaled according to its fitness function. By making multiple copies from the pool of chromosomes, the better chromosomes have more chances of being copied.

3. Crossover: New chromosomes (offspring) are produced from the chromosome pool that was created in the previous step. Crossover occurs according to the defined crossover probability which is chosen as 0.9.

4. Mutation: This procedure randomly selects a single element (the disc along the trajectory) in a chromosome and changes it. The mutation probability is set at 0.1.

5. Check for the termination criterion: If the termination criterion (no fitness improvement for 10 generations) is not satisfied then repeat from Selection step.

3.4 Robot Motion Planning

The trajectory to be followed by the robot end-effector has been generated in Section 3.3. Next, robot motion planning to realise the designed path (in Section 3.3.2) is another important element. Inverse kinematics is a nonlinear and configuration dependent problem that may have either an unique solution, no-solution, or multiple solutions. When faced with the no solution problem, a standard GA is adopted to determine the joint angles by minimising the error of position and rotation compared with the desired values. In this section, the generic-algorithm-amended inverse kinematic method will be presented. In addition, for safety operations, a collision avoidance algorithm is applied to generate a collision-free trajectory.

3.4.1 Genetic-Algorithm-amended Inverse Kinematics for Determining Robot Configurations

3.4.1.1 Inverse Kinematics (IK)

The inverse-kinematics problem is concerned with determining the robot joint angles based on the desired end-effector configuration. For sand-blasting, the robot arm, which is equipped with a blasting nozzle, has to be operated in a three-dimensional space to effectively remove the current surface coating. In the control space there are six dimensions corresponding to the joints of the arm, three of which can be described as the orientation, while the other three represent the position. Position and orientation are related to the world coordinate frame. Point clouds, which represent the environment as detected by a sensor, are initially in a sensor coordinate frame. Knowledge of the sensor position w, t, r (twist, rotation, translation) and the world coordinate frame allows these points to be converted to that coordinate by applying a rotation and a translation to the sensor derived points [51].

A homogeneous transformation matrix is used to conveniently indicate a rotation and translation between the coordinate frames.

$$\mathbf{T} = \begin{bmatrix} \mathbf{n} & \mathbf{o} & \mathbf{a} & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.6}$$

where $\mathbf{n}, \mathbf{o}, \mathbf{a}, \mathbf{p}$ denote normal, approach, orientation and position respectively. They are 3×1 orthogonal unit vectors and $\mathbf{p}$ is a 3×1 position vector.

According to the Denavit-Hartenberg convention, the relative position and orientation of two consecutive links can be described by Homogenous Transformation.

$$\mathbf{T}_{i-1}^{i}(\theta_i) = \left(\begin{array}{ccc|c} & \mathbf{R}_{i-1}^{i}(\theta_i) & & \mathbf{p}_{i-1}^{i}(\theta_i) \\ \hline 0 & 0 & 0 & 1 \end{array} \right) \tag{3.7}$$

In this equation, the relative orientation and position of the frame are described by $\mathbf{R}^i_{i-1}(\theta_i)$ and $\mathbf{p}^i_{i-1}(\theta_i)$ respectively. The parameters of these matrices can be extracted from the physical shape and configuration of the DENSO VM-6083D-W robot arm used in this project. The detailed parameters of the robot arm are shown in Figure 3.10. To calculate the position and orientation of the joints $(\mathbf{T}_{oe}(\theta_1, \theta_2, \cdots, \theta_n))$, $n \leq 6$ in respect to the base of the robot for arbitrary joint angles $[\theta_1, \theta_2, \cdots, \theta_n]$, the transformer will be:

$$\mathbf{T}_{oe}(\theta_1, \theta_2, \cdots, \theta_n)) = \prod_{i=1}^{n} T^i_{i-1}(\theta_i) = \left(\begin{array}{ccc|c} & \mathbf{R}_{oe} & & \mathbf{p}_{oe} \\ \hline 0 & 0 & 0 & 1 \end{array}\right) \tag{3.8}$$

The robotic inverse kinematics, is the problem of finding $[\theta_1, \theta_2, \cdots, \theta_n]$, from the Homogenous Transformation Matrix $\mathbf{T}_{oe}$. This problem is a mapping from the 3-dimensional task space to the joint angle space and usually has more than one solution.

The process of determining the joint angles is based on the desirable end-effector configuration by inverse kinematics (IK). When faced with the multiple solution situation, the minimum joint angle change solutions are picked from the solution group. When faced with the no-solution problem, a GA is adopted to determine the joint angles by minimising the error of position and rotation compared with the desired values.

3.4.1.2 Genetic Algorithm for Inverse Kinematics

The approach adopted to solve the IK problem using a genetic algorithm is treated as a minimisation problem. The error between the end-effector position and orientation of an individual and the desired location is defined as the measure of fitness. In the genetic algorithm, measurements of the fitness of each individual chromosome are required to

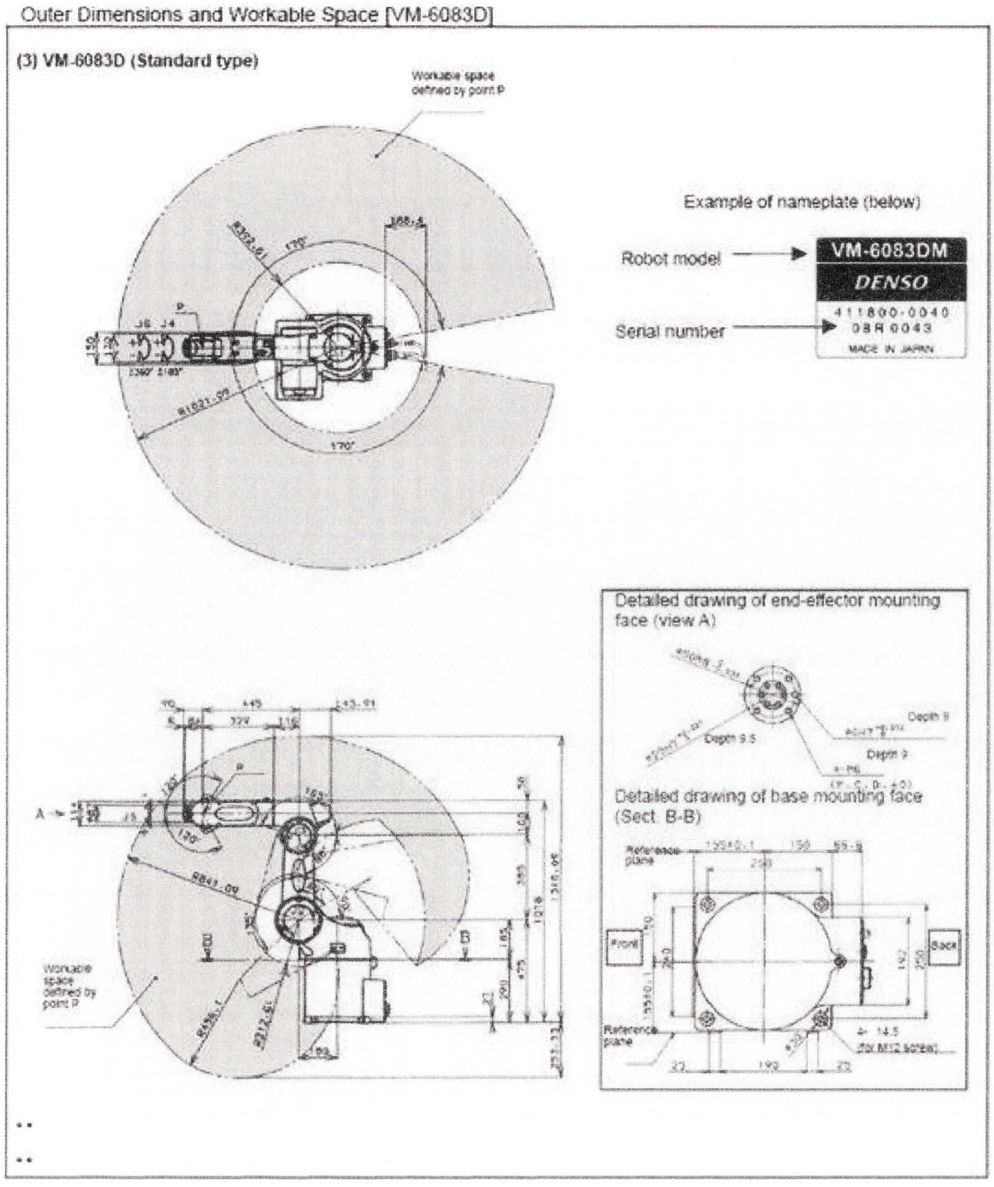


Figure 3.10: Parameters of DENSO VM-6083D-W robot arm

select the most suitable individuals for genetic operations. To measure the position error we compared the vector $\mathbf{P}_e = [p_{ex}, p_{ey}, p_{ez}]^T$ obtained from the homogeneous transformation matrix, which is the difference between the end-effector position of each individual and that of the desired position $\mathbf{P}_t = [p_{tx}, p_{ty}, p_{tz}]^T$ in the configuration space, that is,

$$P_{error} = \min\{\|\mathbf{P}_t - \mathbf{P}_e\|\} \tag{3.9}$$

The orientation error formulation is

$$O_{error} = \min\{\|\mathbf{a} - \mathbf{n}_t\|\} \tag{3.10}$$

where $\mathbf{a}$ is a unit vector which lies along the blasting stream, and is the same as the approaching vector as in equation 3.6. $\mathbf{n}_t$ is the normal vector of the desired target surface. With equation 3.9 and equation 3.10, the objective function for the minimisation can be written as:

$$F_{objective} = \alpha_p P_{error} + \alpha_o O_{error} \tag{3.11}$$

Here, α_p and α_o are the weighting factors and can be used to normalise their corresponding values.

3.4.1.3 Joint Movement

In order to create feasible joint movement after obtaining the sequencing of blast spots, m points are inserted along the previously generated trajectory. $\mathbf{T}$ is used to indicate the Homogenous Transformation matrix of interpolated points, which is 4×4 and includes the relative orientation and position.

In practice, sand-blasting is conducted by the robot arm equipped with blasting nozzle and hose. The hose will be fixed on the arm body while enabling rapid changes in robot configuration. Due to the hose management requirement and to avoid joint vibrations, the robot angular velocity is limited in practice. The algorithm for generating joint angular velocities is given below. Figure 3.11 is a flowchart of the proposed algorithm.

A description of each step is as follows:

1. The angular difference of each joint angle is calculated when the configuration changes from the current position to the next position. Here, $\theta_{angdiff}(i)$ is used to denote the angular difference of each joint, where i = $1, 2, \cdots, 6$.

2. Calculating joint angular velocity by

$$V_{current} = \frac{\theta_{angdiff}(i)}{\triangle\, t} \tag{3.12}$$

3. The current joint angular velocity is compared with the maximum allowed angular velocities where the maximum angular velocity is taken as a desired value. If the current angular velocity is smaller than the maximum allowed angular velocity, then output the resultant joint angles, otherwise go to step 4.

4. Extending the joint movement time:

$$\triangle\, t = \frac{\theta_{angdiff}(i)}{V_{limited}} \tag{3.13}$$

3.4.2 Collision Avoidance

The path or motion planning process automatically generates the path for a robot arm to move from one position to another position following the planned paths. The paths have

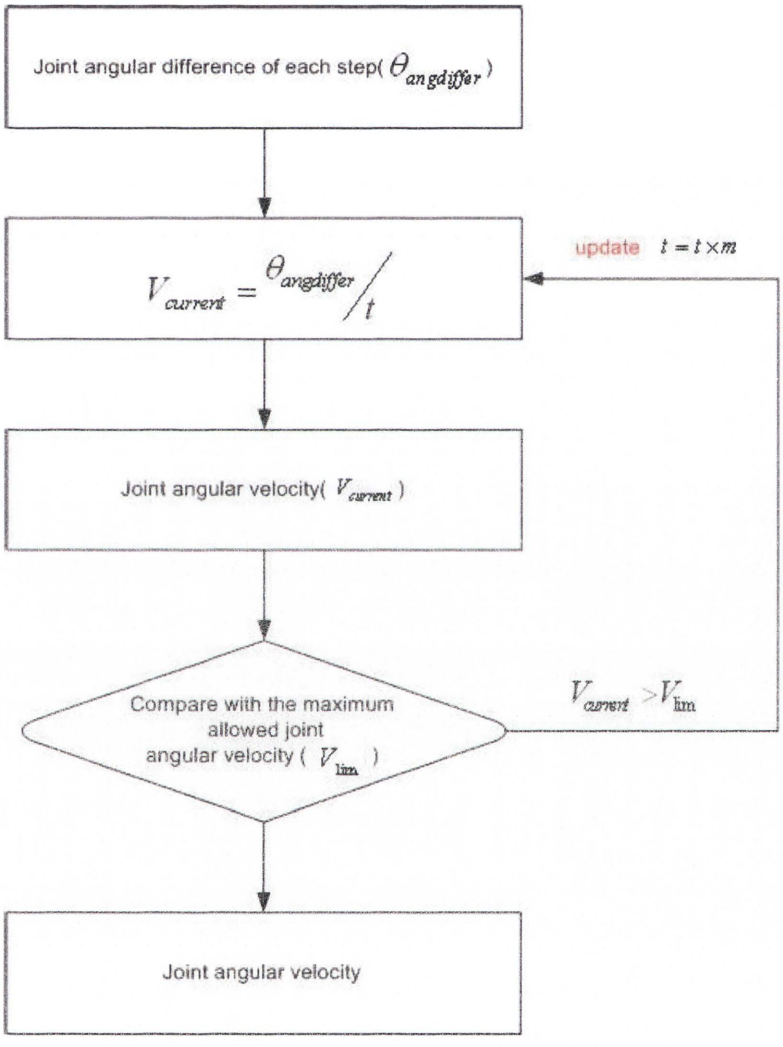


Figure 3.11: Process of generating joint angular velocities

to be collision-free. Thus, collision avoidance for the whole robot manipulator system has been considered as a very important issue, especially in the complex environment under steel bridges. In order to avoid collisions [52][53], a force field (F^2) based collision avoidance method is adopted in this research. In this method, a virtual force field is continuously generated to cover every moving part of the robotic arm. The interactions among the robot's force fields and obstacles provide a natural way for collision avoidance while the robotic arm is conducting sandblasting tasks in the complex environment of bridge maintenance. The links of the robot arm are covered by ellipsoids, as shown in Figure 3.12. Two points on a link are selected as the foci. To ensure that a whole robot arm is covered by the ellipsoid, the length of the major axis is set to $C_f = L \times K_p$, where L indicates the distance between the two foci and K_p is defined as a constant larger than 1. For any point in a three-dimensional space, a straightforward approach to the problem would be to put the obstacle into different groups based on their distances to the arm, giving

$$C_c = \begin{cases} 1, & C_f \geq R_1 + R_2 \\ 0, & C_f < R_1 + R_2 \end{cases} \tag{3.14}$$

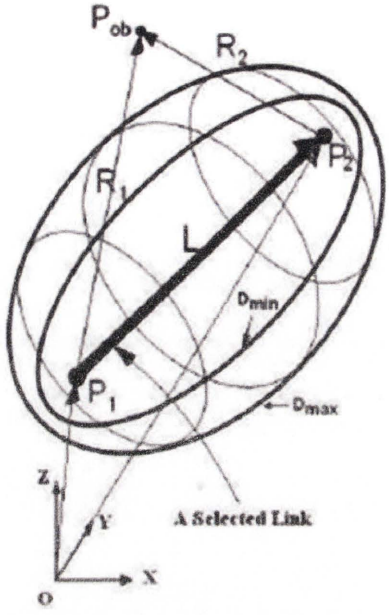
If $R_1 + R_2$ is smaller than C_f, this point is inside D_{min}. Conversely, when $R_1 + R_2$ is larger than C_f, the point is outside D_{min}.

Obviously, the obstacles outside and on the ellipsoid will not influence the motion planning. While $C_c = 1$, this function is further used as a penalty on d_i (distance between consecutive blasting points) with the resultant distance travelled modified as

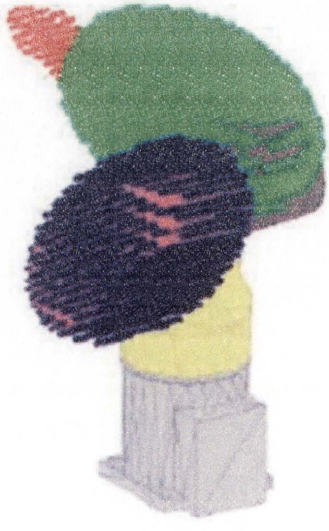
$$d_i \leftarrow d_i + c \tag{3.15}$$

where, c is a constant, indicating a potential collision.

Figure 3.13 shows the flowchart of generating a collision-free trajectory. Each step will be described as follows:



(a)



(b)

Figure 3.12: Parameters of D_{min} and D_{max} ellipsoid (a) and a robot arm covered by D_{min} (b)

1. Choose an initial blasting point from the hexagon-based topology.

2. Determine the following point, calculating each robot joint angle position in its configuration space.

3. Calculate the distance between each joint angle and selected points on the obstacle(s). According to equation 3.14, if $C_c = 1$, penalise the travel distance with equation 3.15.

4. Determine the next point with the objective of finding the shortest travel distance, according to equations 3.3 and 3.4.

5. If the distance is greater than C, then there is no feasible trajectory. Go back to step 2 to search a new trajectory. The frequency of this depends on C . If C is larger, there is less chance with a "no feasible trajectory" situation.

6. Determine the joint angle by the methods which were presented in the previous section.

3.5 Summary

In this chapter, the hexagon-based blasting pattern has been employed to satisfy the complete coverage requirement. A boundary editing procedure is also proposed to ensure that the sand-blasting discs are not beyond the designated surface boundaries. Furthermore, an optimal approach based on the genetic algorithm has been adopted to select a satisfactory trajectory which the end-effector is going to follow. In addition, a GA-based method has been employed, when solutions cannot be obtained from inverse kinematics, to transform the robot from Cartesian space coordinates into corresponding joint configurations. Finally, due to the fact that the steel bridge has a complex structure, the collision avoidance approach has been incorporated to generate a collision-free trajectory.

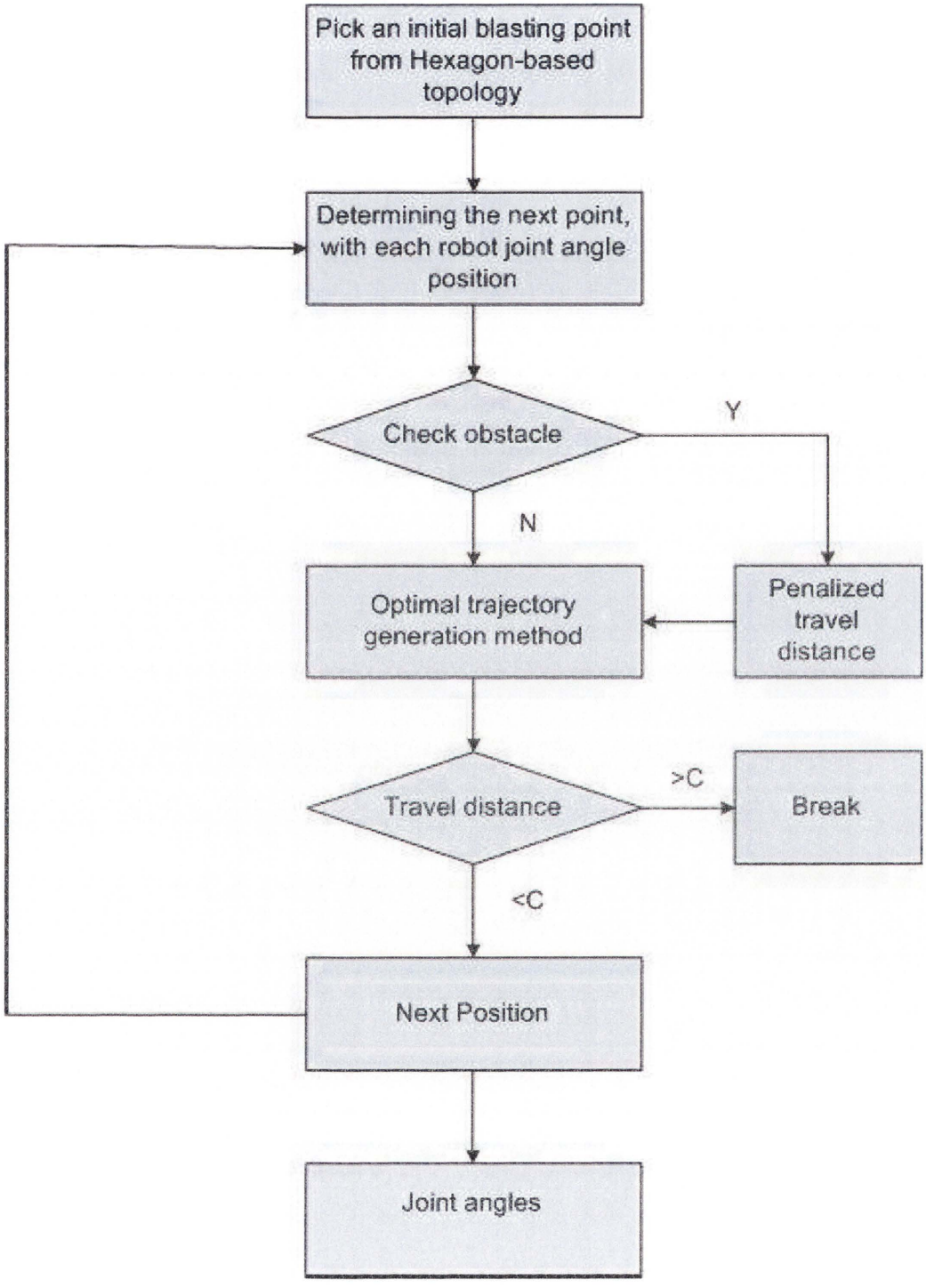


Figure 3.13: Flowchart for generating collision-free trajectory.

Chapter 4

Simulation and Experimental Results

The previous chapter presented the approaches developed for the generation of an effective trajectory followed by the robot arm in a sand-blasting task. This chapter is focused on verifying the performances of these developed methods through a series of simulations and experiments. Based on the availability of a map of the environment (from the mapping step, which has been undertaken by Paul *et al.* [5]), it is assumed that the bridge surface to be blasted is extracted. It is also assumed that the specification of the blasting nozzle is given. Simulations and an experiment are then designed to test for: 1) allocation of blasting spots on surfaces of irregular shape, 2) editing or relocation the blasting spots on the boundary of the surface to enclose them within the surface boundary, 3) sequencing the blasting spots to reduce the blasting distance and the number of turns, 4) deriving the robot arm trajectory and joint angle commands, from the sequence of blasting spots, 5) incorporating an obstacle avoidance algorithm for safe manipulation of the robot arm and 6) driving the real robot arm to follow the generated trajectory using an experimental robot system.

The rest of the chapter is arranged as follows. In Section 4.1, an overview for the

setup of the simulations and experiments is given. Simulations ranging from the allocation of blasting spots to the verification of obstacle avoidance are described in Section 4.2. The experiment conducted to illustrate the performance of the developed methods is presented in Section 4.3. Finally, a summary is provided in Section 4.4.

4.1 Overview of Simulations and Experiments

4.1.1 Physical Environment

Simulations are conducted by emulating an environment close to the experimental setup in accordance with the real-world structure commonly found below a steel bridge, as illustrated in Figure 4.1. The rusted area, surface 1, represents the region in front of the robot that is to be blasted for de-rusting. Other types of surfaces are labelled as surfaces 2 and 3. These surfaces are located on top of the robot as well as on the two sides. A mock-up structure mimicking the real-world environment has been constructed in the laboratory and is shown in Figure 4.2. The robot arm is located under the bridge structure. Other supported structures are shown in the background. It is observed that the test gear closely resembles the real bridge structure. Figure 4.3 shows the dimension of the I-beam which is used in the steel bridge.

4.1.2 Simulated Environment

Figure 4.4, illustrates that a six degree-of-freedom industrial robot is mounted on a movable platform which moves along a track underneath the bridge deck. The bridge surface is represented as a set of point clouds, represented as dots in the figure, above the robot arm. Figure 4.5 presents the dimension of different surfaces which are used in the simulation.

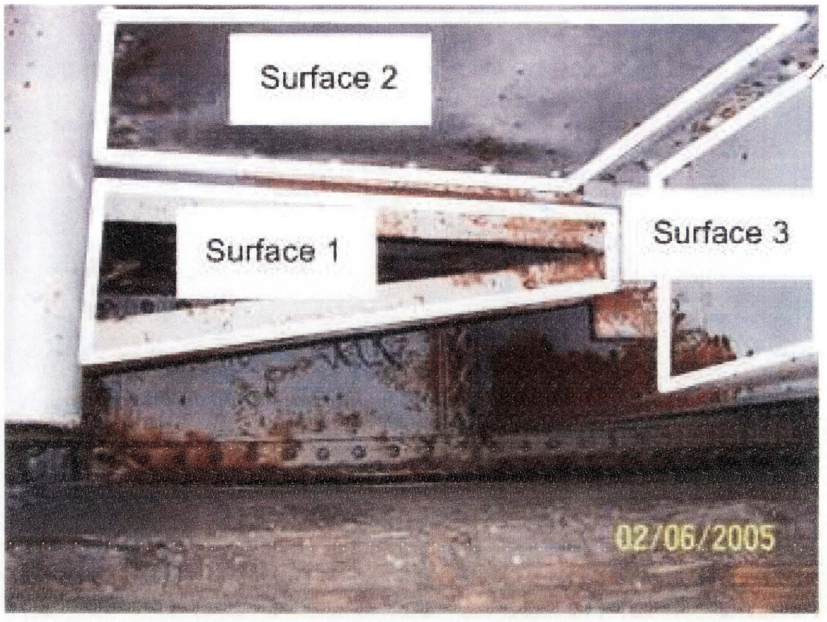


Figure 4.1: Typical structure below a steel bridge.

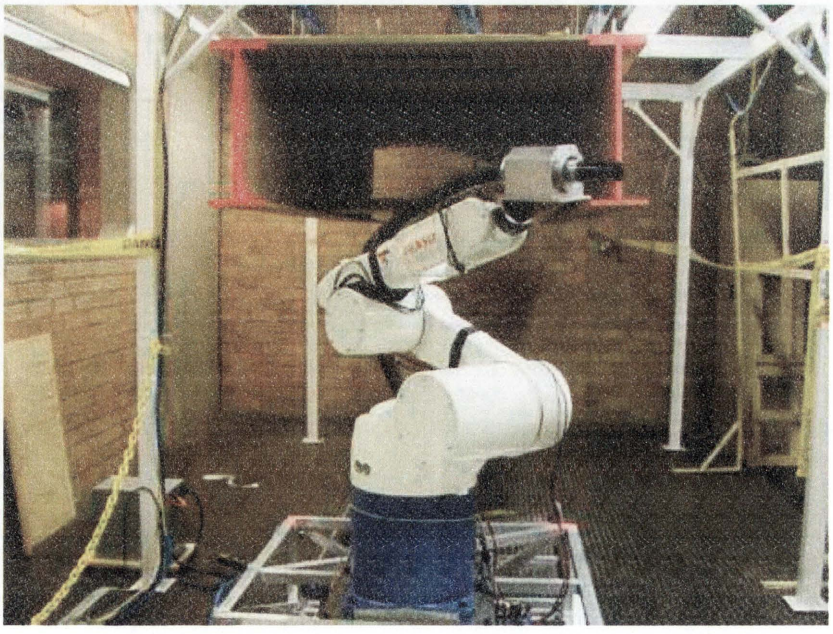


Figure 4.2: Mock-up structure mimicking the environment underneath the steel bridge.

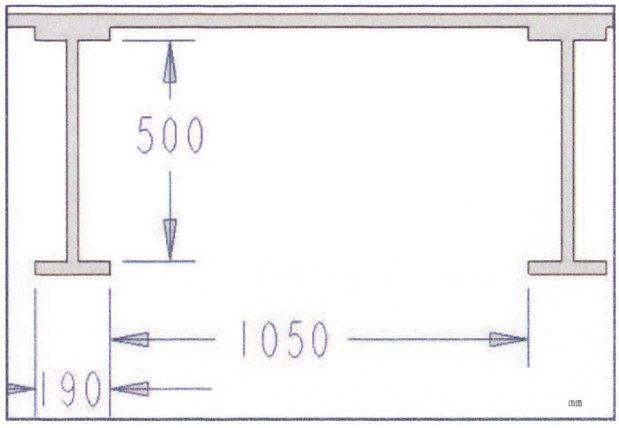


Figure 4.3: Dimension of the I-beam.

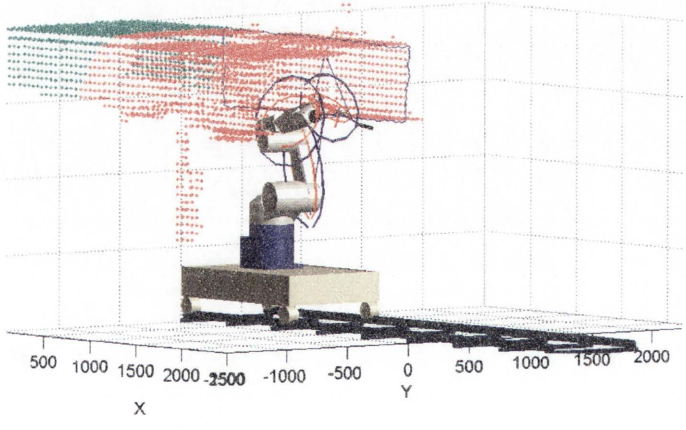


Figure 4.4: Emulated simulation environment underneath the bridge deck.

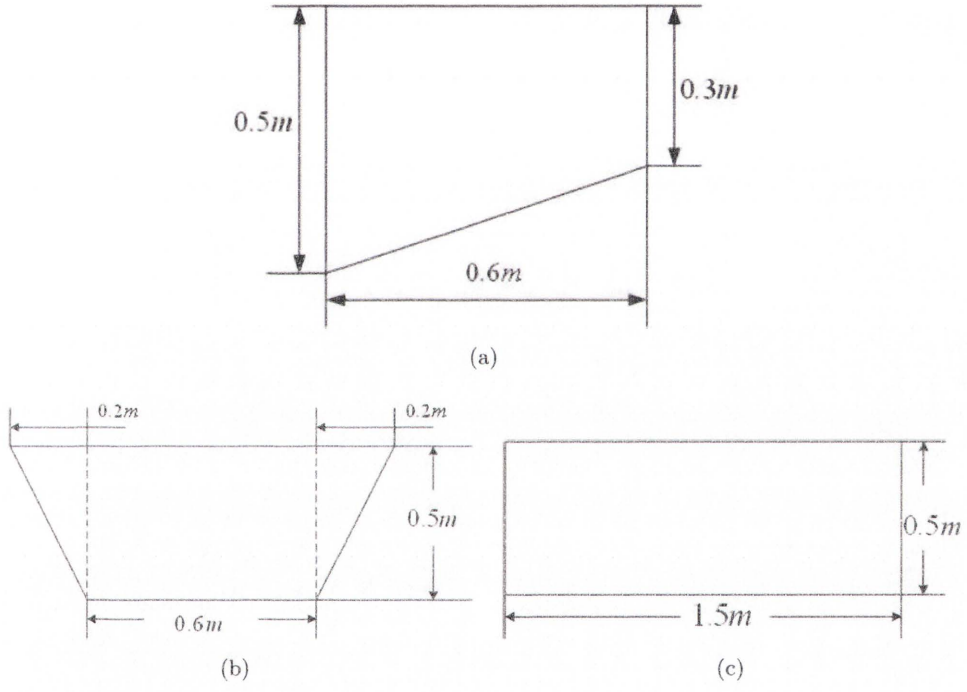
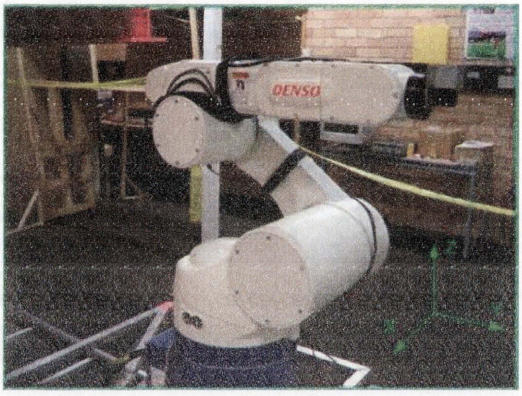


Figure 4.5: Dimension of different surfaces, (a) part of Surface 1, (b) part of Surface 2, (c) part of Surface 3.

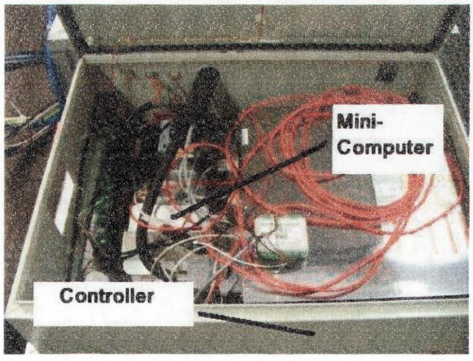
4.1.3 Experimental System Architecture

4.1.3.1 Hardware Components

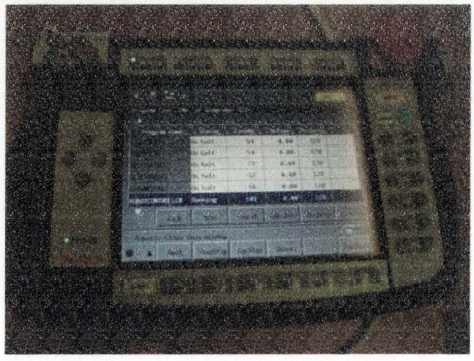
The robot arm used in the test, DENSO VM-6083D, is an industrial robotic manipulator and was supplied together with the motor control hardware and a teach pendant for manual operation. Figure 4.6(a) shows the robot arm; the motor controller and the mini-computer are shown in Figure 4.6(b), and Figure 4.6(c) is a photograph of the teach pendant.



(a)



(b)



(c)

Figure 4.6: Components of the industrial robot arm: (a) the DENSO VM-6083D arm, (b) motor controller, (c) teach pendant.

4.1.3.2 System Architecture

In addition to the industrial robot, the robotic sand-blasting system is made up of several supporting equipment components. Figure 4.7 shows the system architecture needed to construct the blasting system for experiment and real-world operation. The robot's motor is controlled by the associated controller connected to a mini-computer on the movable platform. The emergency stop switch and a teach pendant are attached for safety and manual operation. The mini-computer controls the platform motor and the robot arm. A remote computer is used for human-robot interface and is connected via a wireless network to the mini-computer.

The software configuration is depicted in Figure 4.8. On one hand, the robot controller is driven by the PAC programs shipped from the manufacturer. On the other hand, in-house interfaces have been developed in C++ dynamic libraries, to be called from user programs currently written on the MATLAB platform. These include the software implemented for trajectory generation and motion control.

By making use of the hardware and software setup presented above, simulations and experiments were conducted to verify the developed methods. In the following section, simulations for testing coverage, editing, sequencing and trajectory will be described, together with the presentation of the experiment, and the results will be discussed.

4.2 Simulations

The following simulations were conducted and results are shown. They include tests for: 1) coverage for irregularly shaped surfaces, 2) editing of blasting spots, 3) sequencing of blasting spots, 4) generation of arm trajectory and 5) incorporation of obstacle avoidance functionality.

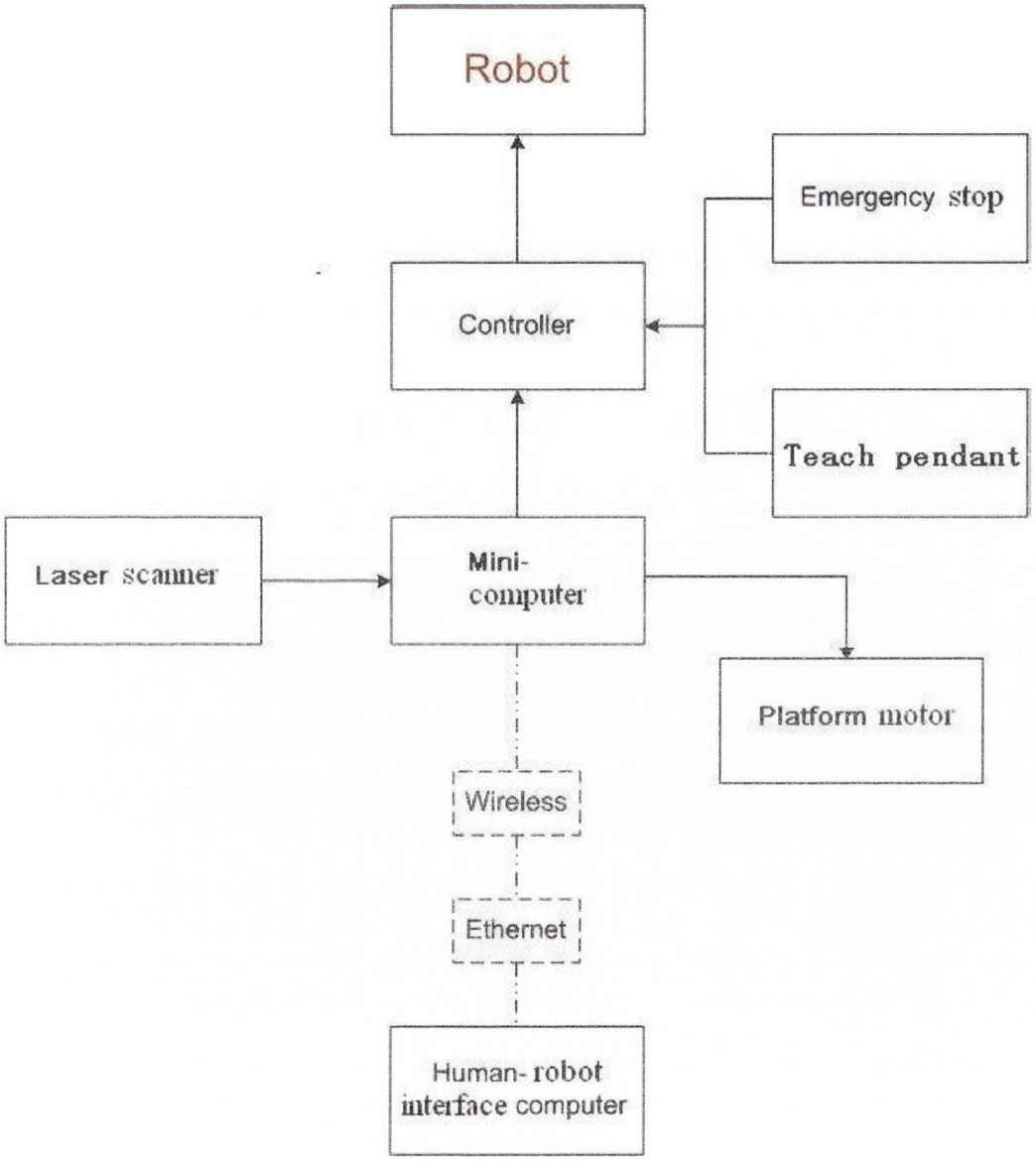


Figure 4.7: Robotic sand-blasting system architecture.

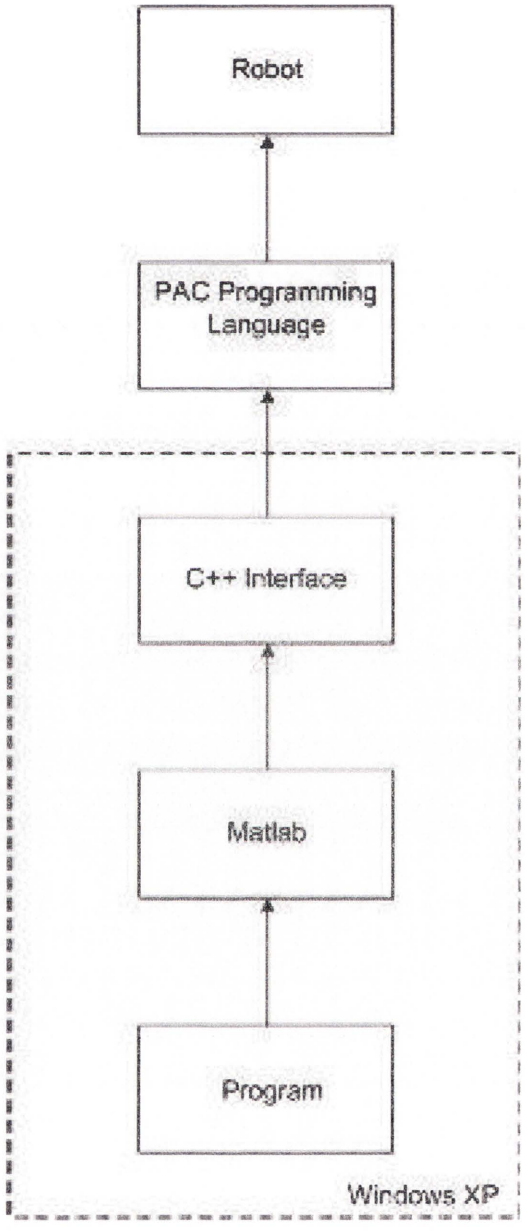


Figure 4.8: Software configuration of the robotic sand-blasting system.

4.2.1 Coverage of Irregularly Shaped Surfaces Using Hexagon-based Pattern

Assume that the environment is represented by a set of point clouds and a surface is identified for sand-blasting, including the following shapes: 1) trapezoidal, 2) circular and 3) arbitrary. Based on the given specification of the blasting nozzle, blasting spots are formed on the surface in circular discs and their radii are $0.04m$ where the nozzle is steered from the surface at $0.3m$ perpendicularly. Results are presented in Figure 4.9.

The locations of the blasting spots are indicated by circles constructed in accordance with a hexagonal pattern. The results illustrate that, irrespective of the shape of the surface to be blasted, the hexagonal coverage pattern is able to cover the surface appropriately.

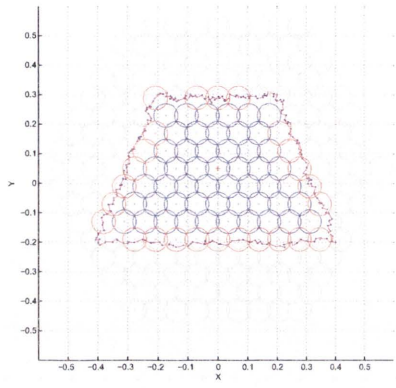
4.2.2 Editing of Blasting Spots

It is observed from the coverage results that the discs needed to be classified as within, on and out the boundary of the surfaces. As developed in the previous chapter, the procedure to relocate or editing the blasting spots on the boundary is tested here. In Figure 4.10, the results of editing regions of the blasting surface are depicted. Here, the the thin zigzag lines in the figures are the boundary of the desired blasting surface.

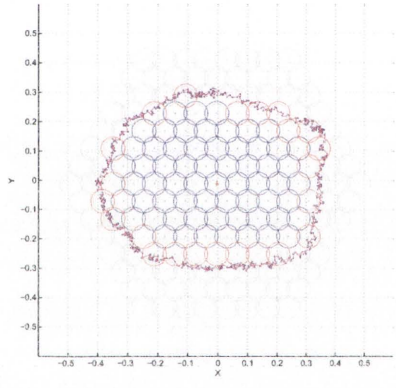
As indicated by the results, independent of the contour of the boundary, the editing procedure is able to re-locate the blasting spots within the blasting surfaces.

4.2.3 Sequencing Blasting Spots

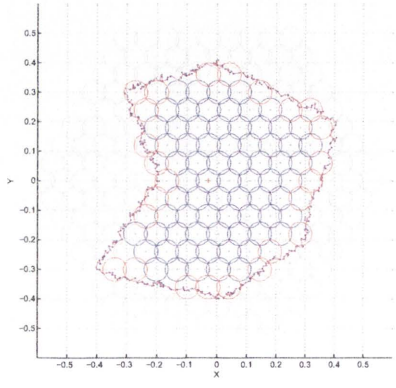
After the blasting spots have been classified and relocated within the boundary of the surface, the test for sequencing the spots for sand-blasting operation is then conducted.



(a)

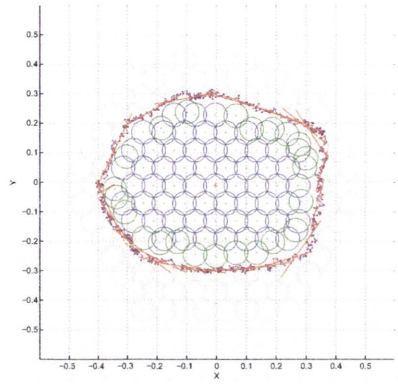


(b)

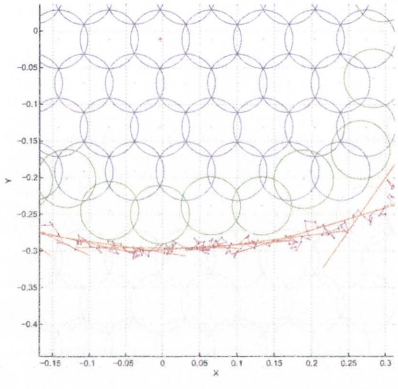


(c)

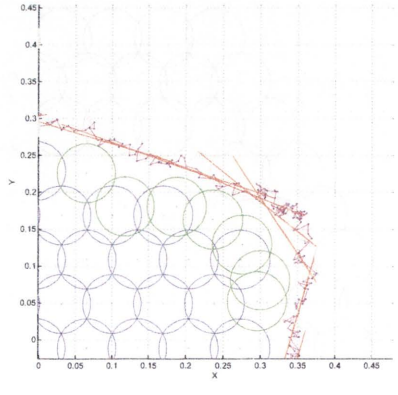
Figure 4.9: Test for coverage of irregular shaped surfaces, (a) trapezoidal, (b) circular, (c) arbitrary shape.



(a)



(b)



(c)

Figure 4.10: Results of editing the blasting spot locations, (a) overall edited result, (b) bottom boundary, (c) top-left boundary.

The sequencing is cast as an optimisation problem solved by using for which the genetic algorithm. Table 4.1 lists the parameters of the genetic algorithm used in the simulations.

Table 4.1: Parameters of the genetic algorithm used in the simulations

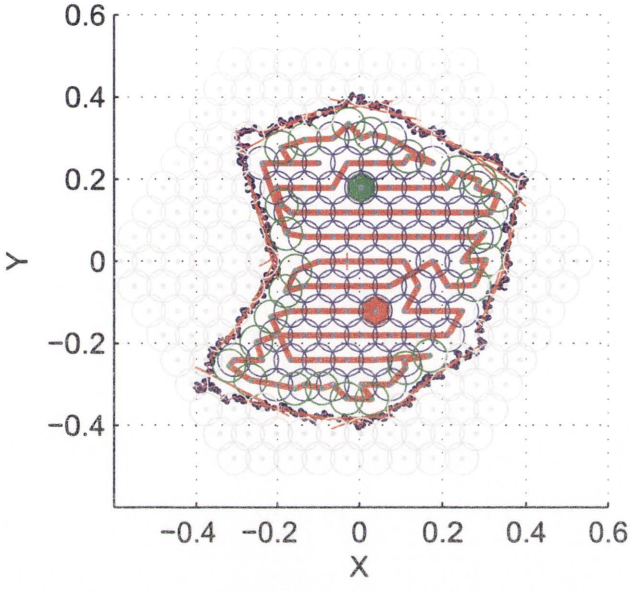
crossover probability	0.9
mutation probability	0.1
maximum generation	300
size of chromosomes	number of blasting spots

Two cases are simulated by considering the following two objectives:

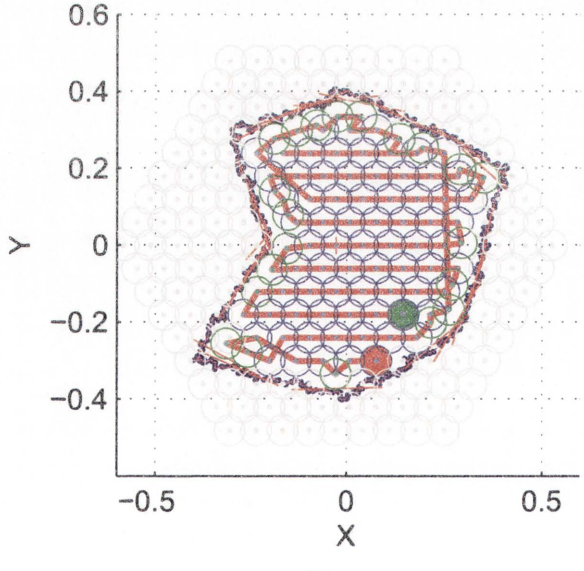
1. shortest travel distance of the blasting nozzle,

2. shortest travel distance and minimal number of turns.

Results are presented in Figure 4.11. In this figure, the starting points are marked in green and end points in red. Figure 4.11(a) presents the result (i.e., path) is obtained by minimising the travel distance of the nozzle. It is observed that the sequence generated is satisfactory, however, it is also noticed that the path has turns. A further simulation is conducted, minimising the travel distance and the number of turns. The result is depicted in Figure 4.11(b). It is noticed that the path length is similar to Figure 4.11(a) and the objective of minimising the turns is achieved. The results of the distances travelled and the numbers of turns in these two case studies are summarised in Table 4.2. It is shown that the distances travelled are comparable for the two case studies. The number of turns has been reduced when such an objective is taken into consideration in the second case.



(a)



(b)

Figure 4.11: Sequenced blasting spots, (a) minimising travel distance, (b) minimising travel distance and the number of turns.

Table 4.2: Summary of blasting spot sequencing results

Objective	Distance (m)	Number of turns
Distance	6.62	32
Distance and Turn	6.10	25

4.2.4 Generation of Robot Arm Trajectory without Collision Avoidance Functionality

The generated sequence of blasting spots is exported to the robot arm path planning algorithm to generate a trajectory of the arm. In this section, the approach adopted to transform the blasting spots from the Cartesian space into their corresponding arm joint configurations without obstacle avoidance is described. In order to resolve the singularity problem that exists in the method using inverse kinematics, the genetic algorithm is employed to identify the robot joint angles when the singularity problem occurs. As a requirement of hose management, the robot joint angle velocity is limited to below $30°/sec.$ The robot model used is the DENSO VM-6083D-W, the work-cell setup that was shown in Figure 4.4. During the simulation process, the base of the robot is assumed to be placed at $(500, 375, 370)mm.$ Nozzle length is set as $100mm$ and the stream length is set as $300mm.$ Due to safety issues, the smallest robot pose is chosen as the initial configuration, where joint angles are $(0°, -75°, 160°, 0°, 30°, 0°)$, respectively.

There are two simulation cases , including:

Case 1: *Trajectory generated by inverse kinematics only (without collision avoidance)*

A snapshot of a full sand-blasting nozzle path is given in Figure 4.12, which shows how the robot arm travels from its initial position and configuration to the first sand-blasting target on Surface 1. It is noticed that the starting point, ending point and target cell

are the same as in Figure 4.11. The computational time is approximately 37 *sec* using a computer with 2.40GHZ CPU and 1.0GB of RAM. Figure 4.13 illustrates that the robot moves by following the planned path as generated by inverse kinematics only. Here, the desired blasting surface is an irregular surface; therefore, the approximate centre of the surface is used to define the location of the surface in the global coordination. The location of the robot is $(500, 375, 370)mm$, the centre of the surface is located at $(0, 500, 1500)mm$.

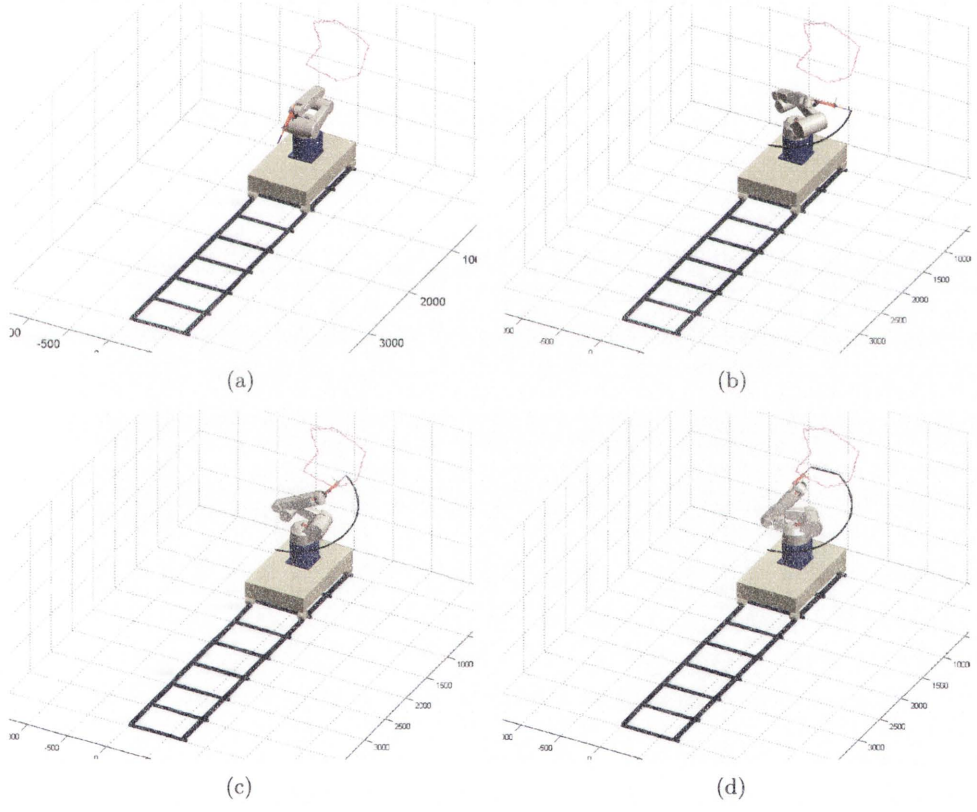


Figure 4.12: The robot travels from the initial position to the first target blasting spot, (a) initial position, (b) and (c) intermediate positions and (d) blasting start position.

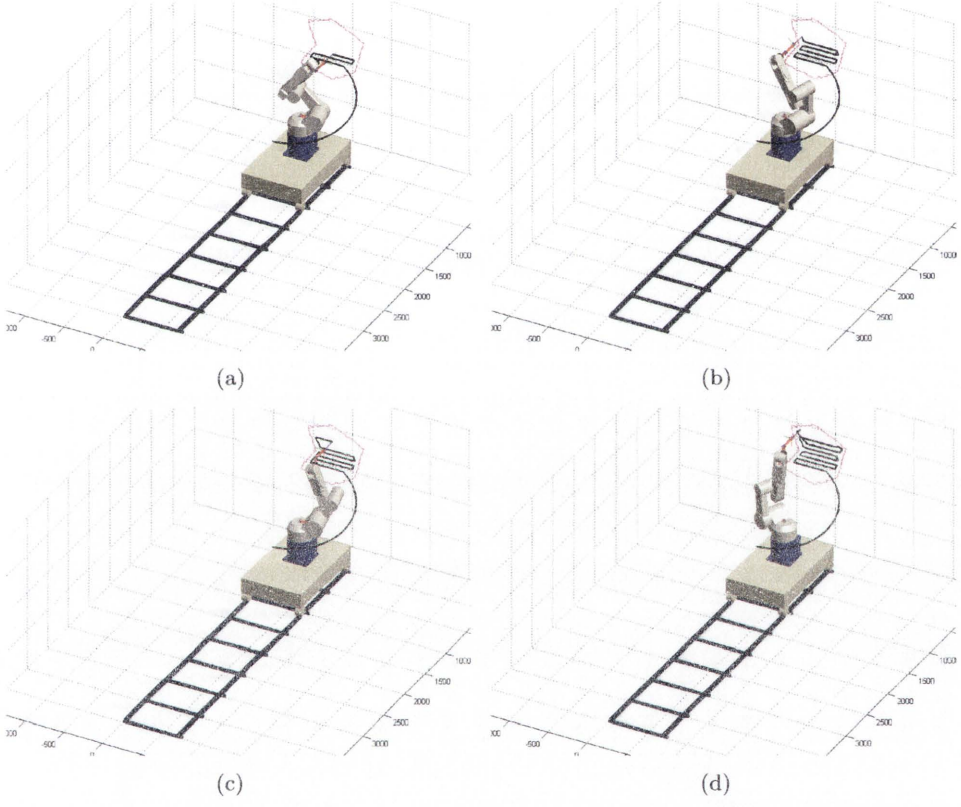


Figure 4.13: The robot motion while blasting a small area after following the planned path, (a) start of blasting, (a) and (b) intermediate positions and (c) end of blasting position.

In Figure 4.12, the initial motion traced a smooth path where there are no constraints on the nozzle's target, except in its final position. The blasting process shown in Figure 4.13 illustrates that the arm is able to follow the generated joint angles and trace the required sequence of blasting.

The traces of joint angles from the robot's initial pose to the first blasting target are shown in Figure 4.14. Due to the accurate and smooth motion requirement, 100 way-points are inserted along the movement from the start configuration to the first blasting target. Obviously, the robot arm moves smoothly in this process.

While the blasting nozzle travels along the generated path, the traces of joint angles are presented in Figure 4.15. However, joint 1 and joint 6 experienced large joint angle changes during the turns. A robot joint angular velocity control approach is employed to make sure that the angular velocity is smaller than the limited value. Figure 4.16 plots the traces of angles joint when velocity control approach is applied.

Case 2: *Trajectory generated by combining inverse kinematic and genetic algorithm (without collision avoidance)*

A snapshot of the sand-blasting nozzle path shown in Figure 4.18 and 4.19 is obtained by inverse kinematics combined with the genetic algorithm approach presented in Chapter 3. The results show that the sand-blasting process is effective on different locations on the bridge's surfaces. Figure 4.19 shows the path and robot motion for blasting surface 3 located at $(0, 1000, 1500)mm$ in the global coordinate. For this position and orientation of the surface, no singularity problem was encountered during the process of obtaining robot joint angles by inverse kinematics. The computational time is approximately $93sec.$

Figure 4.17 shows the robot sand-blasting system blasting Surface 1 located at $(500, 500, 1700)mm$ in the global coordinate. The singularity problem occurred while

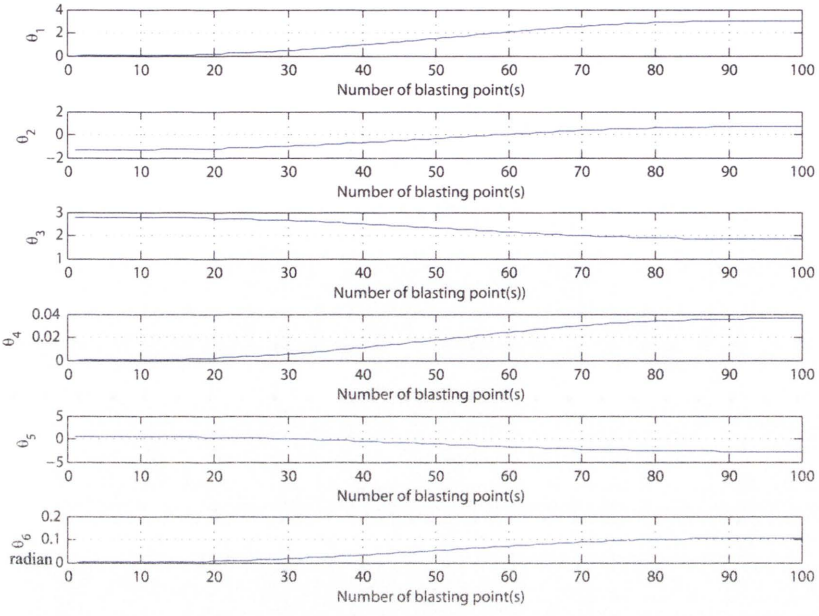


Figure 4.14: Joints angle from the initial robot position to the first blasting spot.

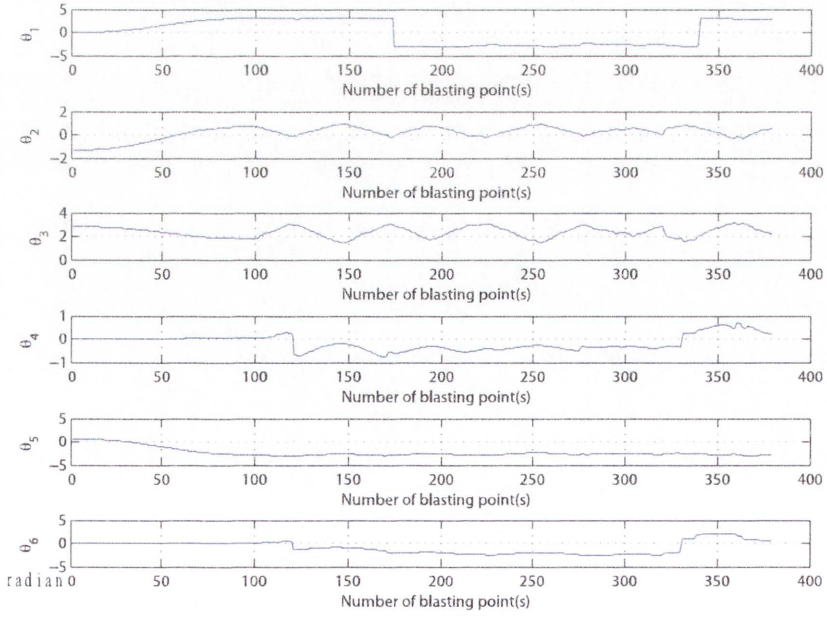


Figure 4.15: The joint angles traced while the robot travels along the generated path.

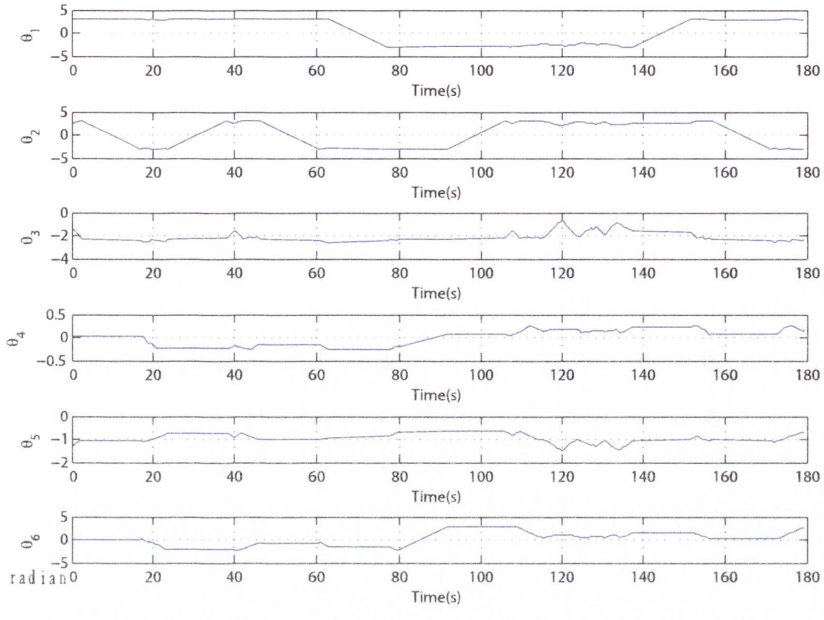


Figure 4.16: Traces of joint movements with movement time extended.

the robot was moving on the positions shown in Figure 4.17(a), 4.17(b), 4.17(c) and 4.17(d). As presented in section 3.4.1.1, the inverse kinematic is required for determining the joint angle given desired end-effector configurations in its Cartesian space. Singularity problems occur when robot axes are redundant, which means that more axes than necessary can be found to produce the same motion, or when the robot is in certain configurations that require extremely high joint rates to move at some speed in the Cartesian space. Here, the genetic algorithm based inverse kinematics approach is applied to handle the singularity problem. The error between the actual position and its desired location are $0.0253mm$, $0.0187mm$, $0.0214mm$ and $0.0281mm$, respectively. The number of errors are dependent on the number of times the singularity problem occurred. The four errors here are observed from the simulation conducted. The error in orientation is very small.

On the other hand, Figure 4.18 shows the robot sand-blasting system blasting Surface 2, located at $(0, 500, 1500)mm$ in the global coordinate. The singularity problem

occurred while the robot was moving on to the positions which are shown in Figure 4.19 while blasting Surface 3. The errors are $0.0247mm$ and $0.0189mm$, respectively. In both situations, the errors are considered acceptable in practice.

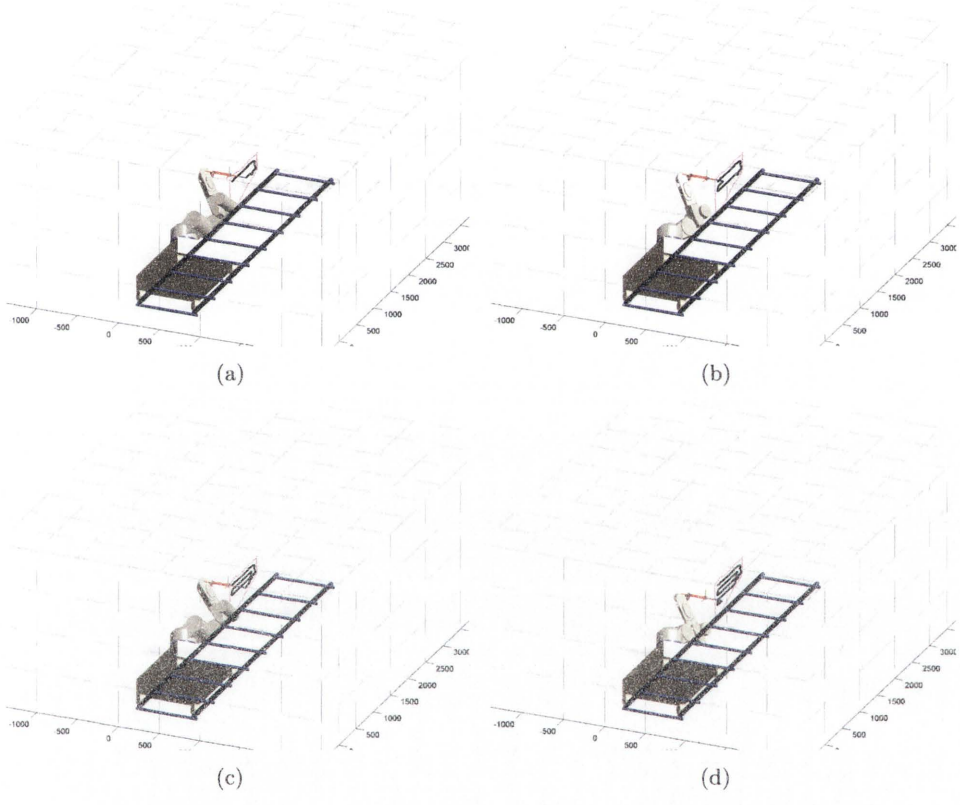
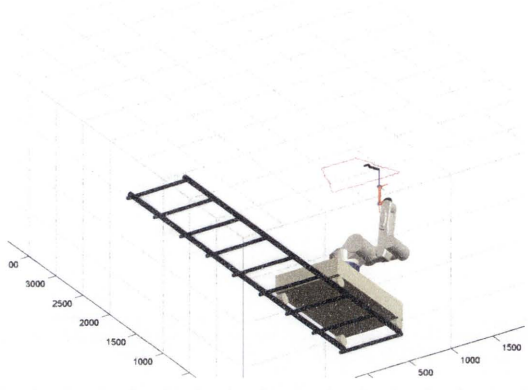
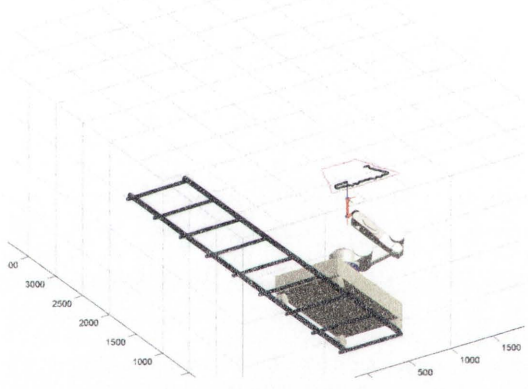


Figure 4.17: The robot motion on Surface 1, following the generated path.



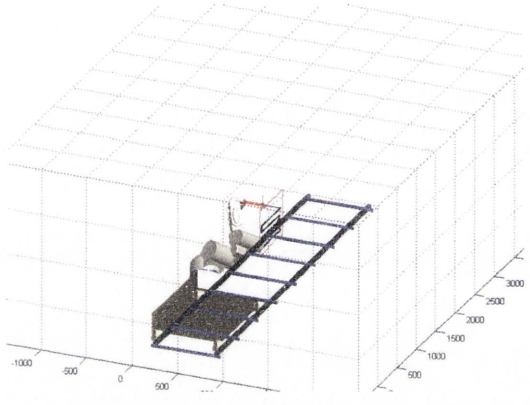
(a)



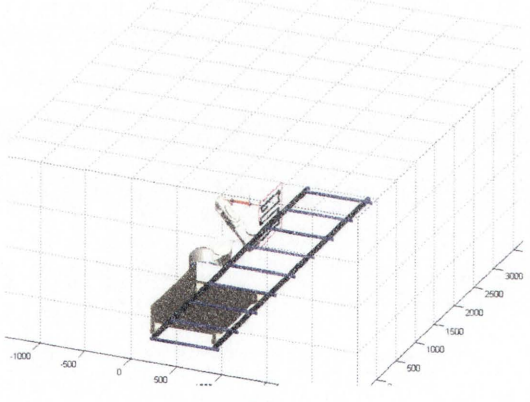
(b)

Figure 4.18: The robot motion on Surface 2, following the generated path.

These results indicate that the proposed approaches are able to conduct the blasting task on different locations and orientations. It is a generic system to be used in varied environments. While the genetic algorithm is used to alleviate the singularity problem during the blasting process, calculating one set of robotic joint angle takes around $3.1sec$ of computer time.



(a)



(b)

Figure 4.19: The robot motion on Surface 3, following the generated path.

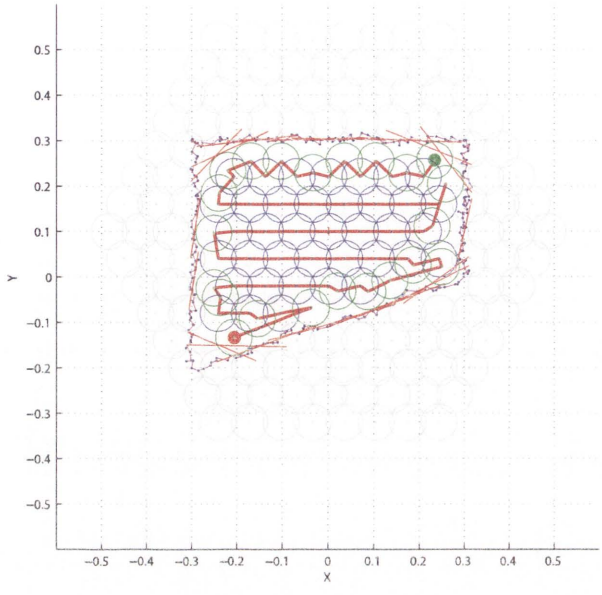
4.2.5 Generation of Robot Arm Trajectory with Collision Avoidance Functionality

4.2.5.1 Effects of Obstacles on the Sequencing of Blasting Spots

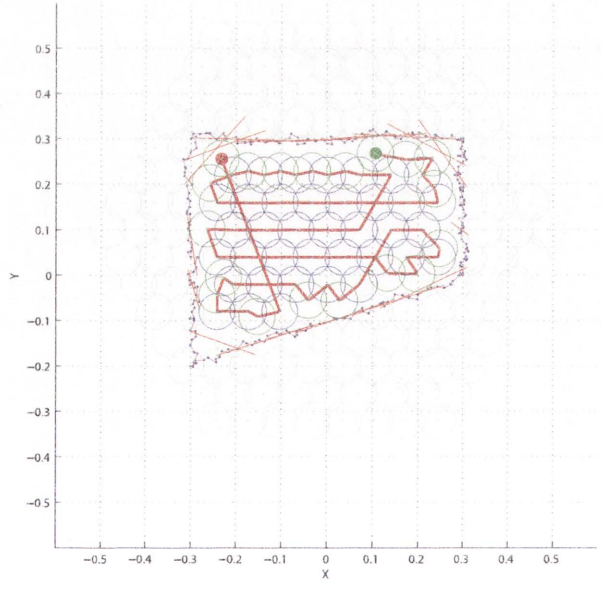
The sequencing of blasting spots in obstacle-free environments was conducted in Section 4.2.3. In this section, the sequencing is cast as an optimisation problem in an obstacle-presented environment, for which the genetic algorithm is used. Table 4.1 lists the parameters of the genetic algorithm which are used in the simulations.

The comparison is conducted between the obstacle-free environments and obstacle presented environments while generating the sequencing of blasting spots, using Surface 1 as an example. The results shown in Figure 4.20 illustrate that when determining the sequence of blasting spots without considering the presence of obstacles, the result obtained from using the shortest distance and reduced turns is better.

In Figure 4.20(a), the sequence obtained resembles that of the previous scenario and the result is considered satisfactory. On the other hand, the blasting sequence obtained when the bridge deck channel is the obstacle contains several sections of non-straight paths, see Figure 4.20(b). Here, the location of the surface (Surface 1) lies between the I-beams forming the channel. As a result, the movement of the arm is to generate constrained and the freedom a satisfactory sequence is thus limited. The results are summarised and compared in Table 4.3. It is noted that the distance travelled by the nozzle is not severely affected by the presence of obstacles. The distance travelled increased slightly from $3.32m$ to $3.52m$ only. In other words, the sequencing algorithm is able to maintain short travel distances irrespective of the obstacles. However, when obstacles are present, the nozzle has to take alternative paths to avoid collisions. Consequently, the number of turns is increased from 17 to 26 as anticipated.



(a)



(b)

Figure 4.20: Comparison of sequencing results due to the presence of obstacles (a) without obstacle, (b) with obstacle.

Table 4.3: Comparison of blasting spot sequencing results (with obstacles)

Obstacles	Distance (m)	Number of turns
Not present	3.32	17
Present	3.52	26

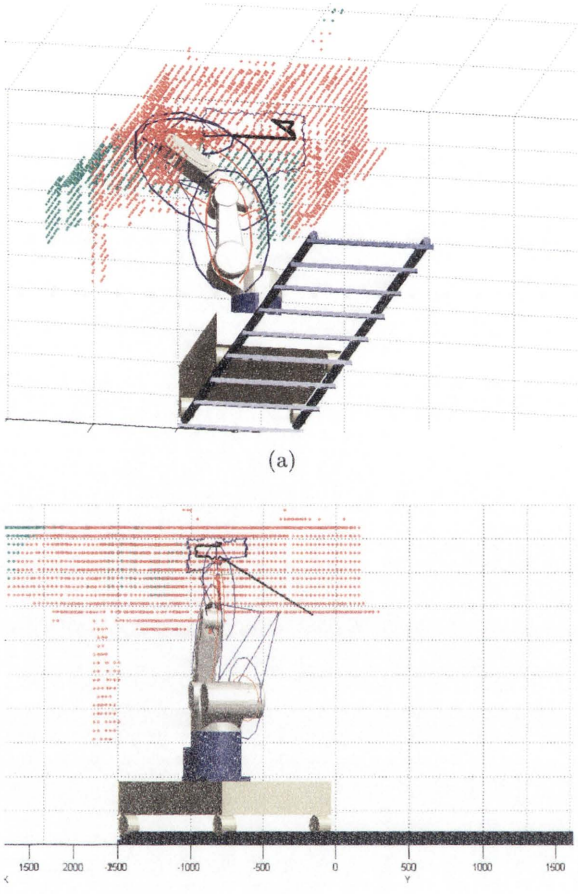
4.2.5.2 Generation of Robot Arm Trajectory in Obstacle Presented Environments

In this section, the amended inverse kinematics approach, combined with the collision avoidance methods are adopted to transform the blasting spots from the Cartesian space into robot arm joint configurations. The cases tested include:

Case 1: *Avoiding collision between the robot arm and bridge structure*
The obstacle avoidance functionality is incorporated into generating the trajectory of the robot. The results indicate that the objective of being collision-free is achieved on different types of surfaces under the complex steel bridge environment. Figures 4.21, 4.23 and 4.25 show simulation snapshots of the robot with the force-field ellipsoid defined by D_{min}, D_{max} and the nozzle path. The values of k_p in the simulations are set to be $k_p = [1.05, 1.10, 1.10]$. The computation time is $78sec$ for 800 iterations while processing Surface 1. For Surfaces 2 and 3, the computational times are $58sec$ and $1756sec$, respectively.

In Figures 4.22, 4.24 and 4.26, it is observed that the distances between the ellipsoid covering the arm and the surfaces have been kept above zero. That is, an effective avoidance of collision with obstacles, in the form of bridge surfaces, is accomplished. Notably, when blasting the surface in front of and above the arm, the distance between the force field ellipsoid and the I-beam fluctuate. This is caused by the fact that the surfaces on the side of the robot are close to the arm's path. On the other hand, for blasting the side surfaces, the distances are relatively constant (Figure. 4.26), reflecting the fact

that collision is not likely because of the openings along the length of the bridge channel.



(a)

(b)

Figure 4.21: The simulation result of the robot conducting sand-blasting of Surface 1 by following the generated path in the complex steel bridge environment

Case 2: *Avoiding collision with an obstacle in the work-space*

Simulation results on different bridge surfaces during the sand-blasting process under the complex steel bridge environments have been presented in the previous cases. In these cases, simulation results show the process of generating a collision-free trajectory while there is a static obstacle in the robot working space. For the case considered here,

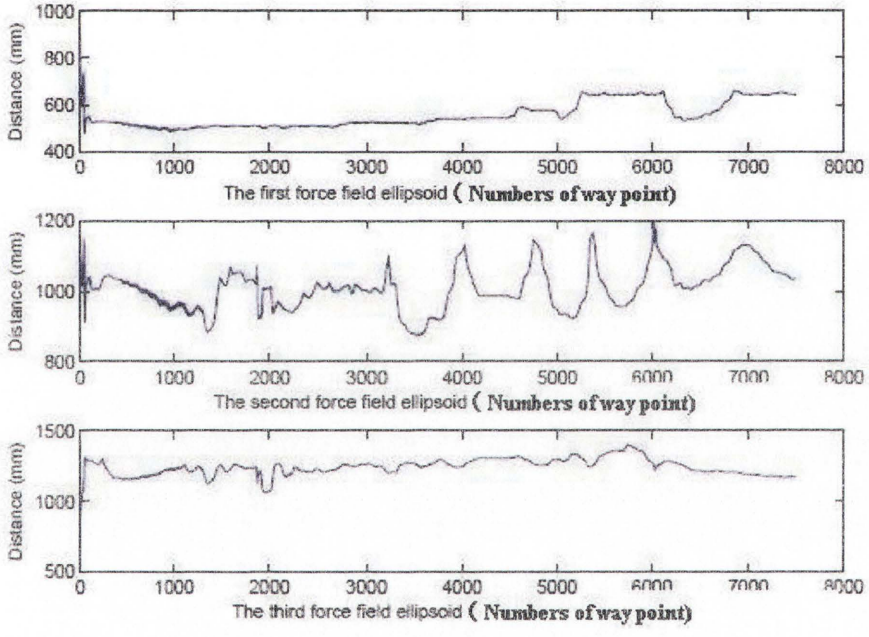
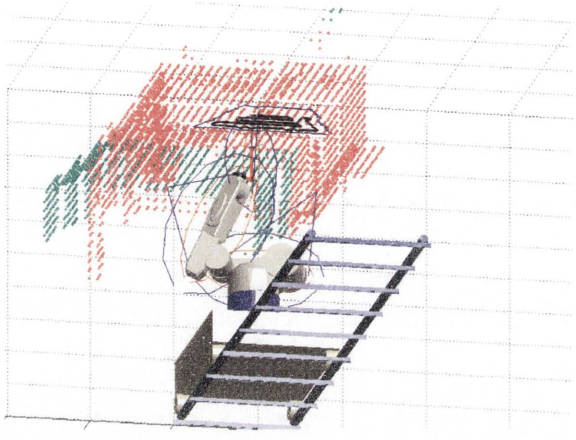
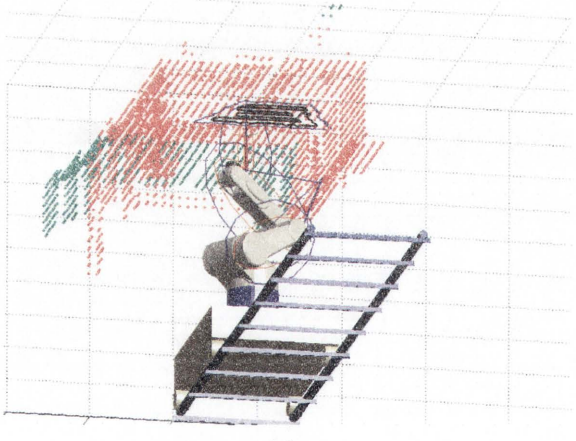


Figure 4.22: Distance between the force field ellipsoid and the I-beam while blasting on Surface 1.



(a)



(b)

Figure 4.23: The simulation result of the robot conducting sand-blasting of Surface 2 by following the generated path in the complex steel bridge environment

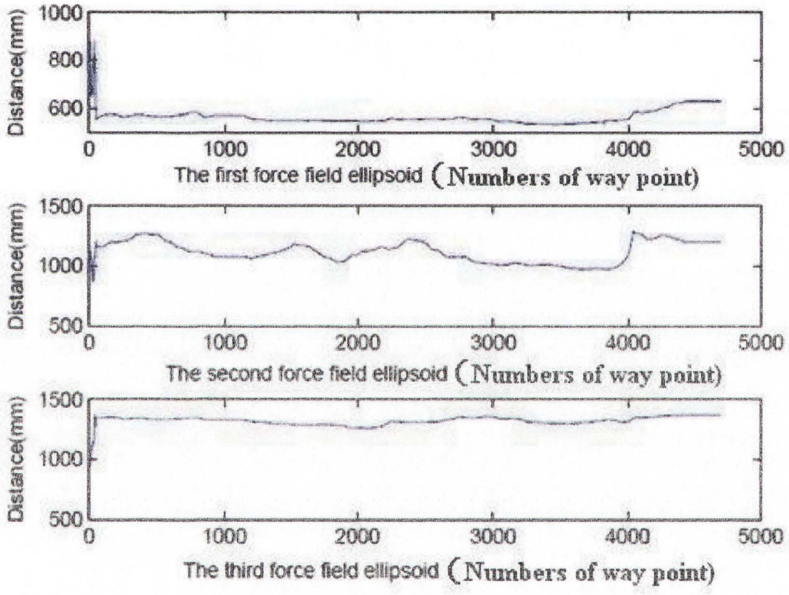
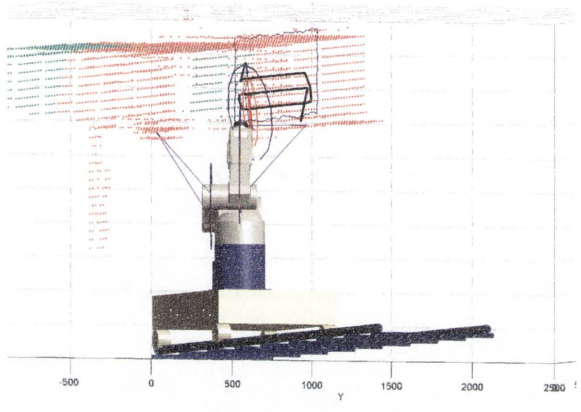
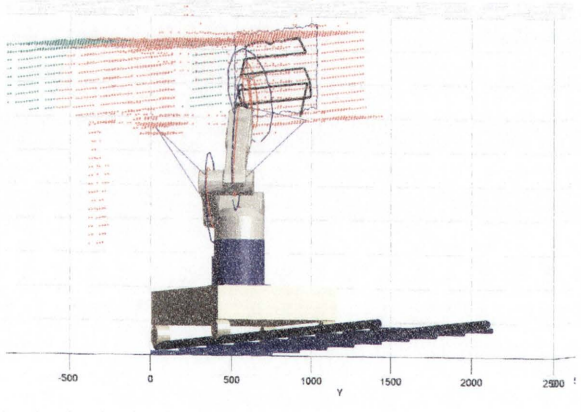


Figure 4.24: Distance between the force field ellipsoid and the I-beam while blasting on Surface 2.



(a)



(b)

Figure 4.25: The simulation result of the robot conducting sand-blasting of Surface 3 by following the generated path in the complex steel bridge environment

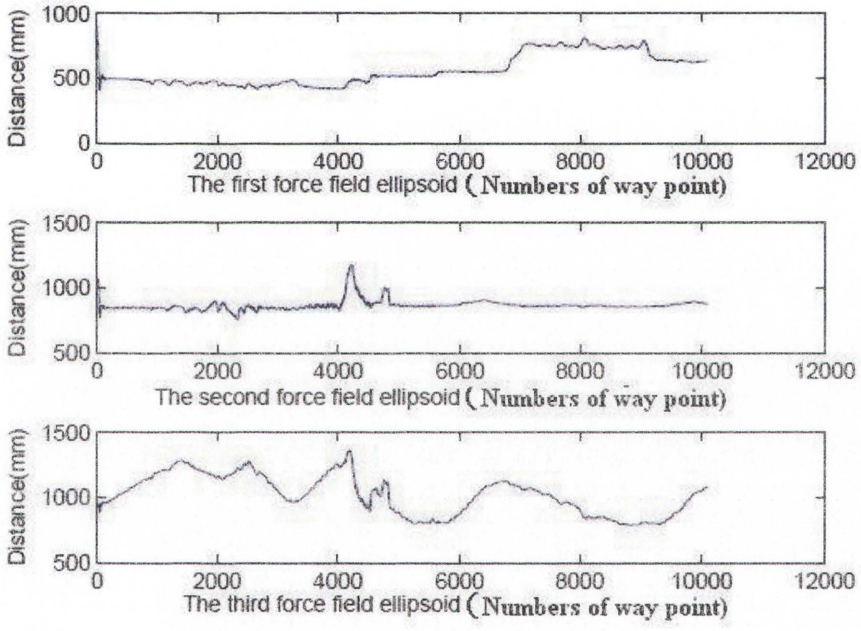


Figure 4.26: Distance between the force field ellipsoid and the I-beam while blasting on Surface 3.

Figure 4.27 shows the robot motion planing while the robot is blasting the surface on the roof of the channel. The red points are used to indicate the obstacle. Figure 4.28 is enlarged to clearly present the relationship between the obstacle and the robot arm. Figure 4.29 shows sand-blasting conducted in front of the robot. Figure 4.30 is also enlarged to clearly present the relationship between the obstacle and the robot arm while sand-blasting. These results indicate that, the collision avoidance based path planning approach successfully protects the robot from collision during the sand-blasting process.

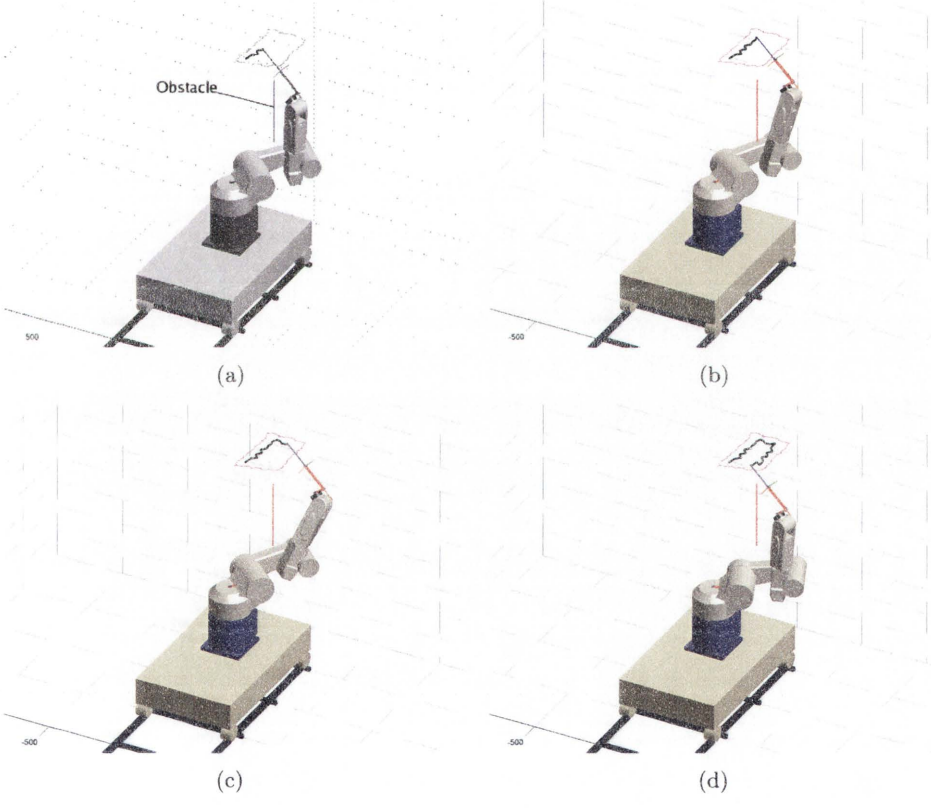


Figure 4.27: The robot blasting a part of Surface 1 in the environment with obstacles. (1)

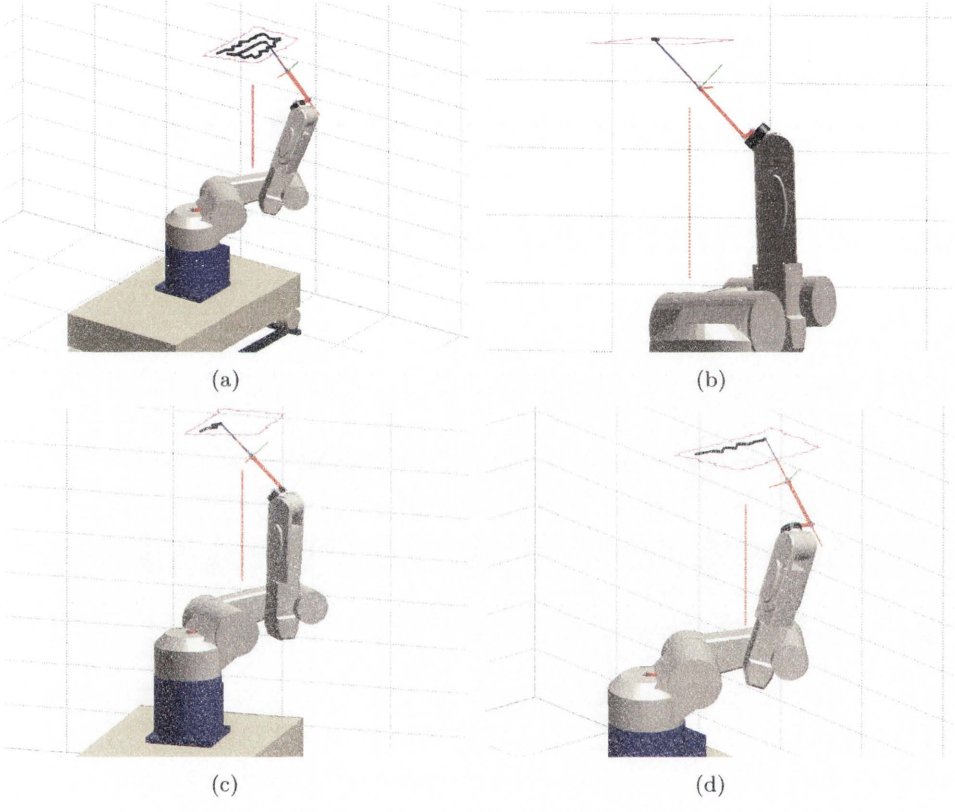


Figure 4.28: The robot blasting a part of Surface 1 in the environment with obstacles. (2)

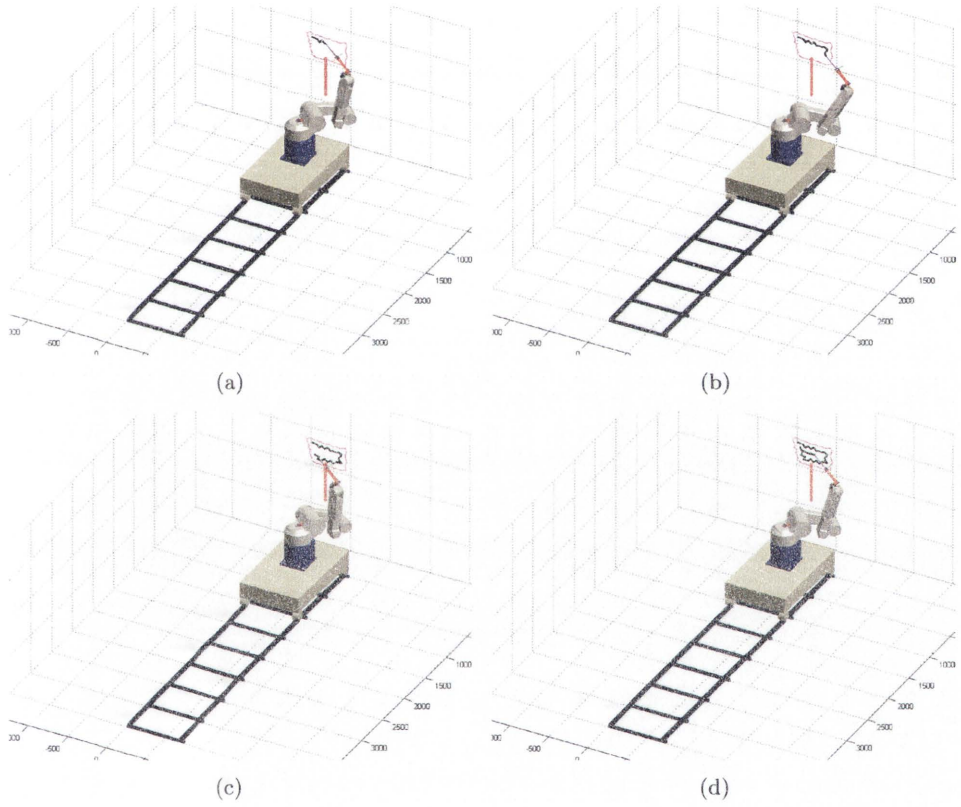


Figure 4.29: The robot blasting a part of Surface 2 in the environment with obstacles. (1)

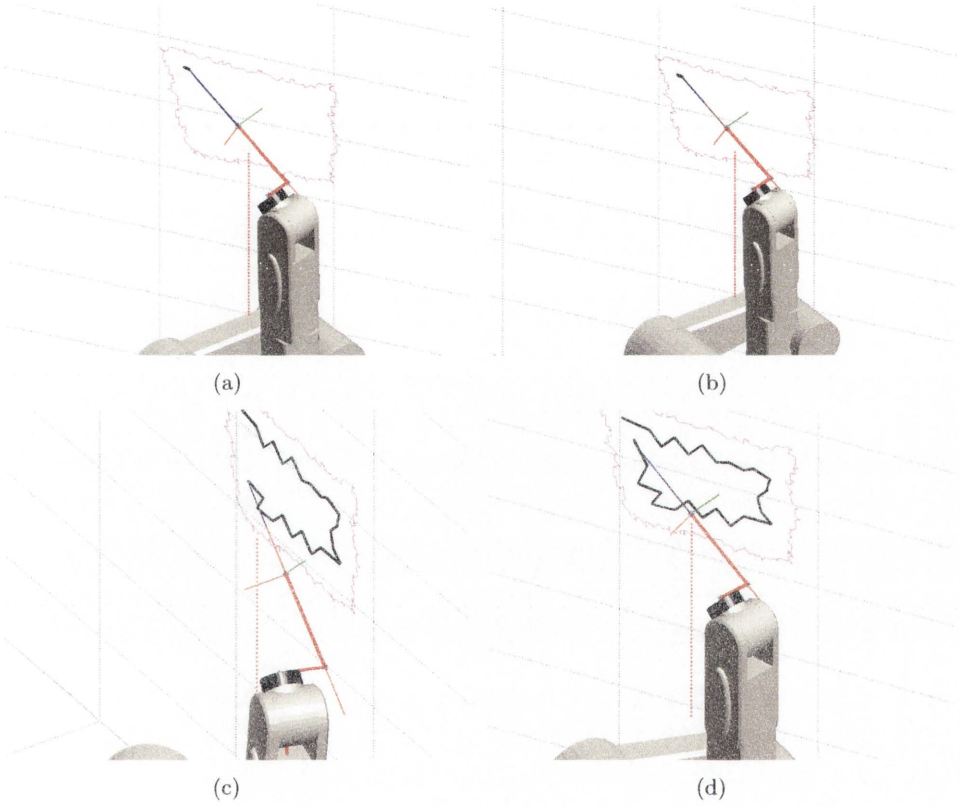
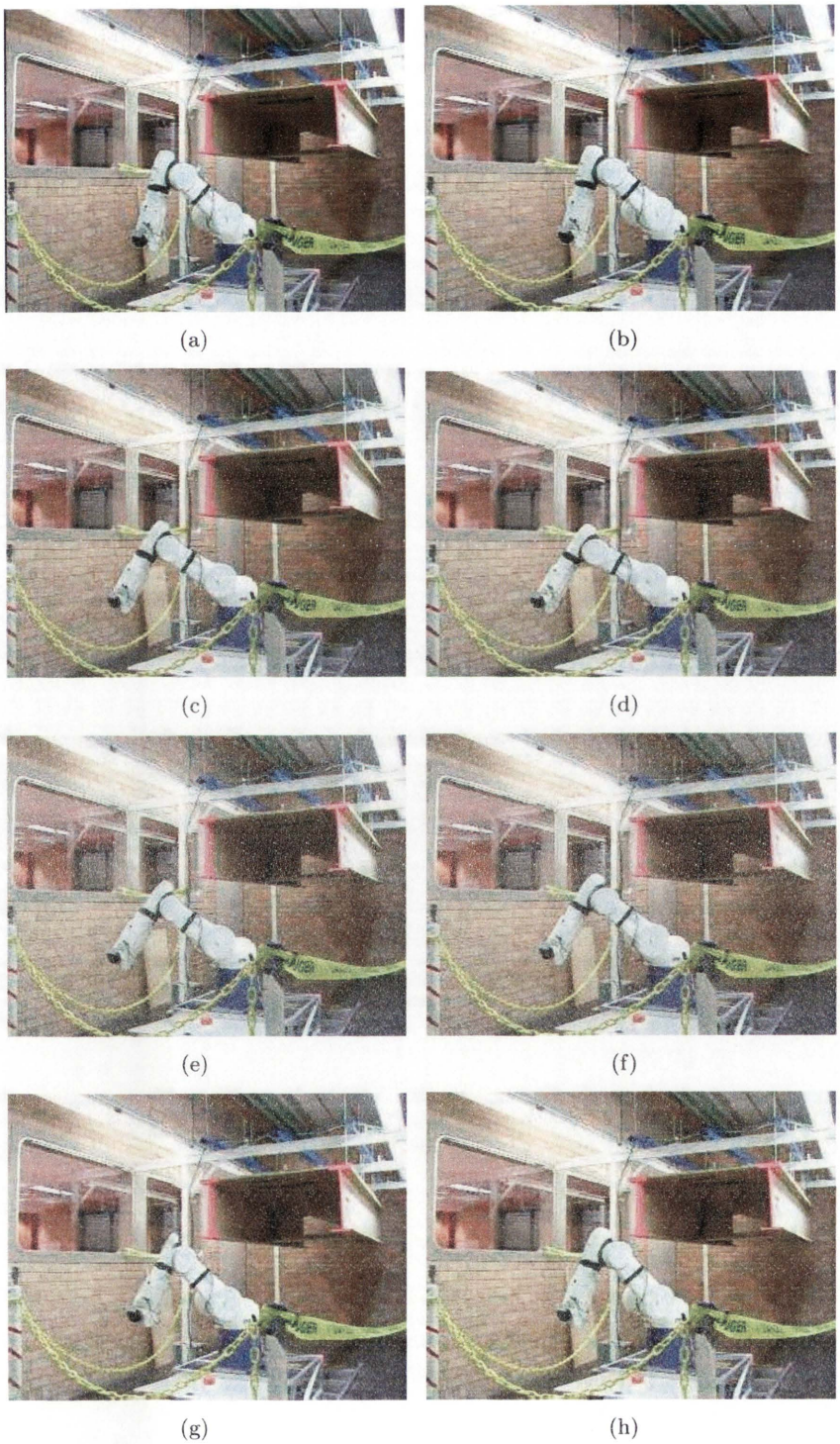


Figure 4.30: The robot blasting a part of Surface 2 in the environment with obstacles. (2)

4.3 Experiment

Based on the simulation results, an experiment was conducted in the laboratory. The robot arm motion followed the sand-blasting trajectory which was determined in this experiment using the trajectory generation approaches and joint angles generated by inverse kinematics. Experimental results are given in Figures 4.31-4.34. These figures are snapshots showing the robot motion with the pre-generated trajectory during the experiment. The question that the experimental results aims to answer in this section is how accurately the robot could follow the trajectory. The trajectory generation process consists of two elements: robot end-effector trajectory generation and robot joint angle identification. This experiment is conducted in the laboratory.



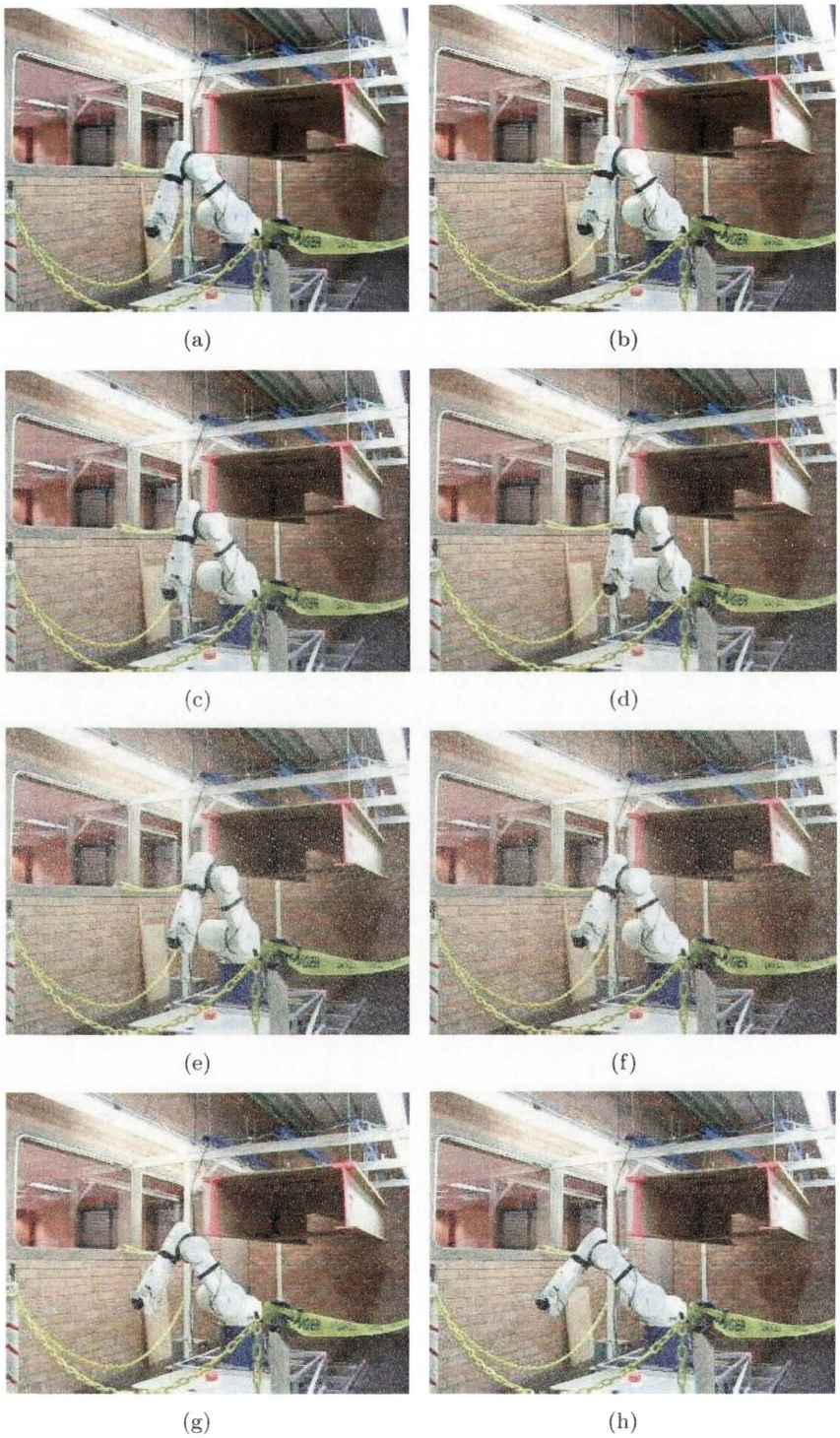
(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 4.31: The experimental results for robot path planning (1).



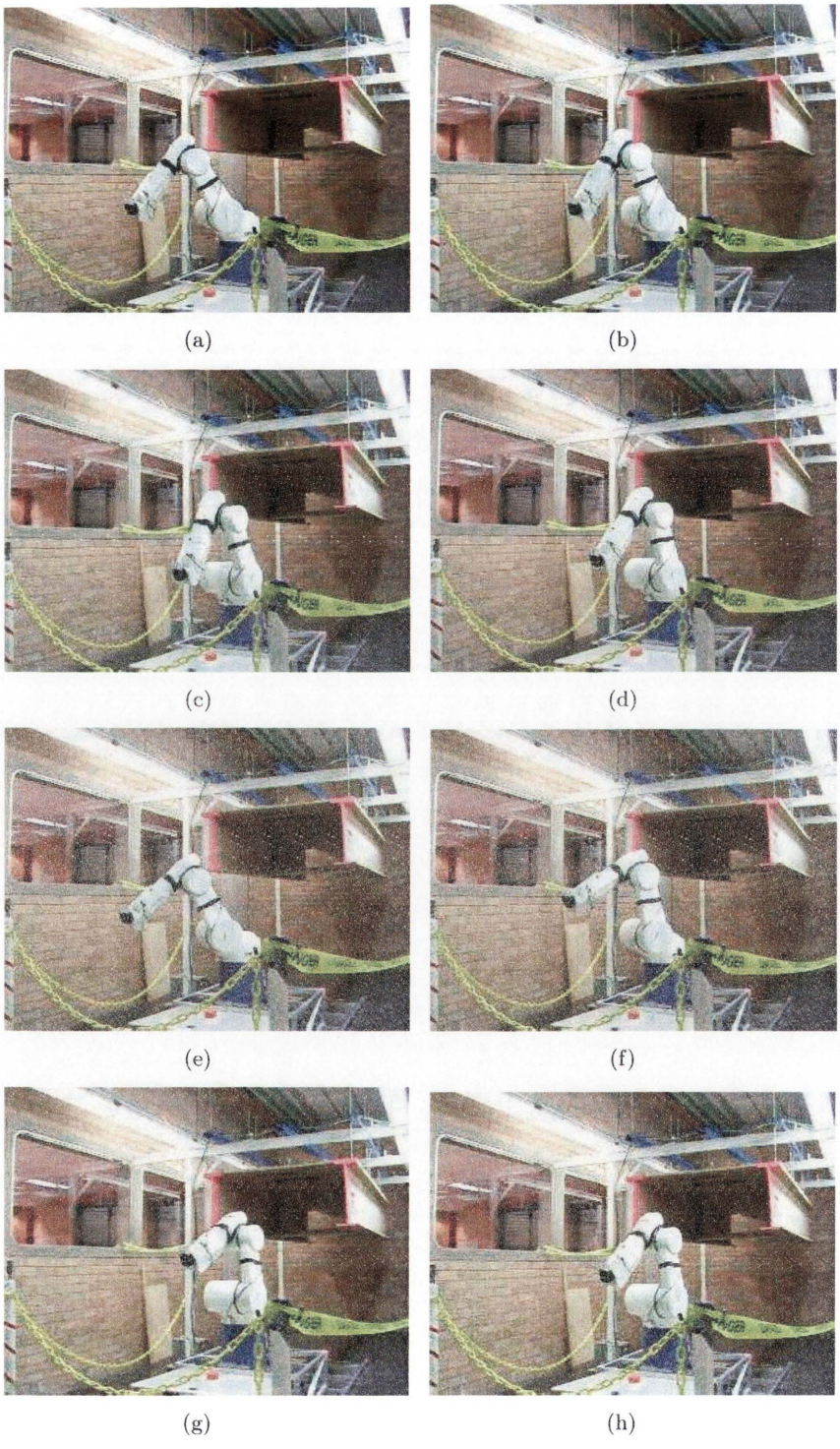
(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 4.32: The experimental results for robot path planning (2).



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 4.33: The experimental results for robot path planning (3)

(a) (b)

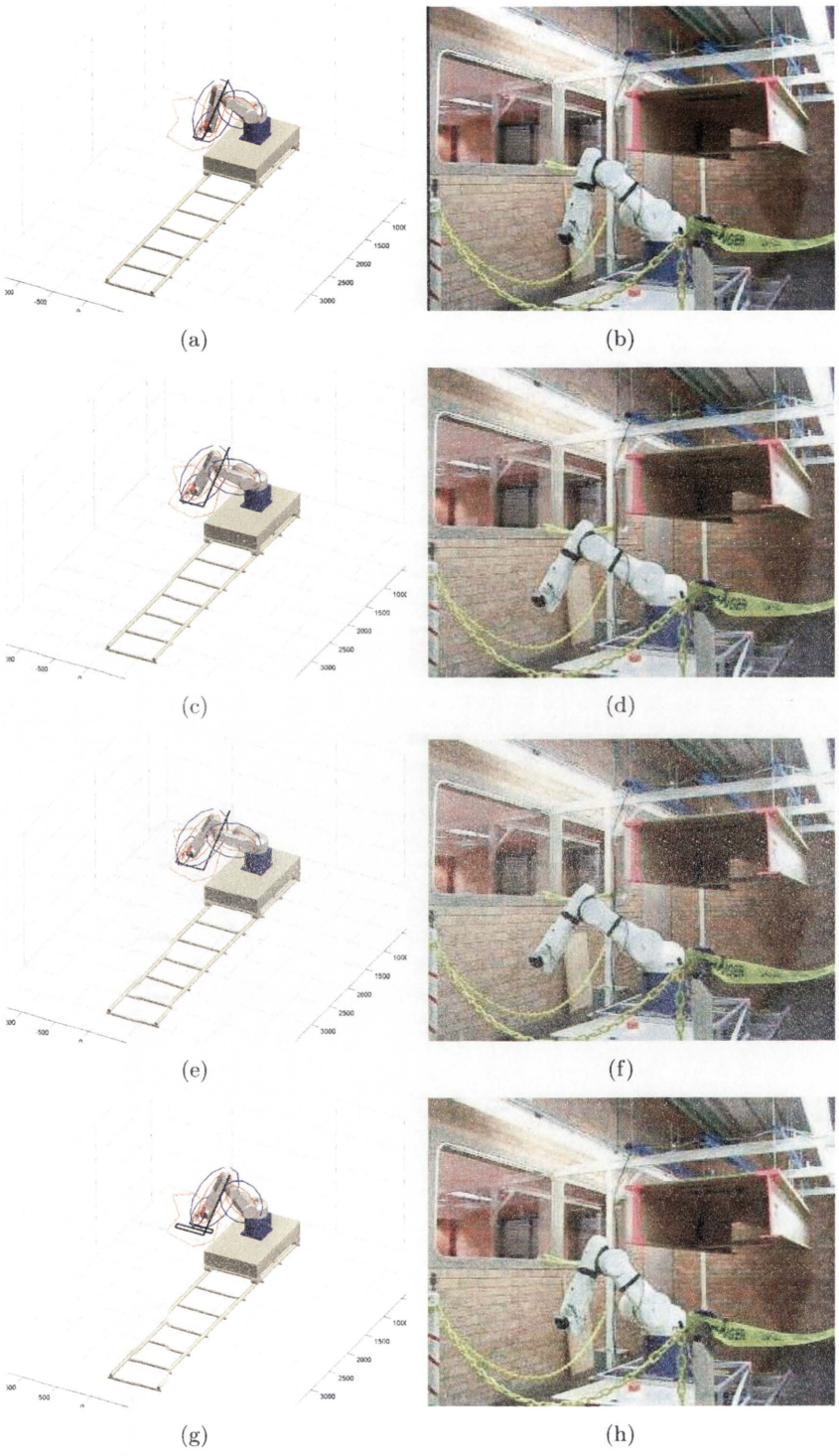
(c) (d)

(e) (f)

(g) (h)

Figure 4.34: The experimental results for robot path planning (4)

In order to protect the robot, the collision avoidance algorithm is employed while the robot is moving. Figure 4.35 and Figure 4.36 show the results. Snapshots of simulation are placed alongside the photographs taken during the arm movements. The surface to be blasted is in front of the robot. The path traced by the robot follows a parallel and zigzagged path from bottom to top. From the comparison, the experiment and simulation results are matched. Thus, this experiment result satisfies the requirement.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 4.35: Comparng between simulation and experimental results for robot path planning (1)

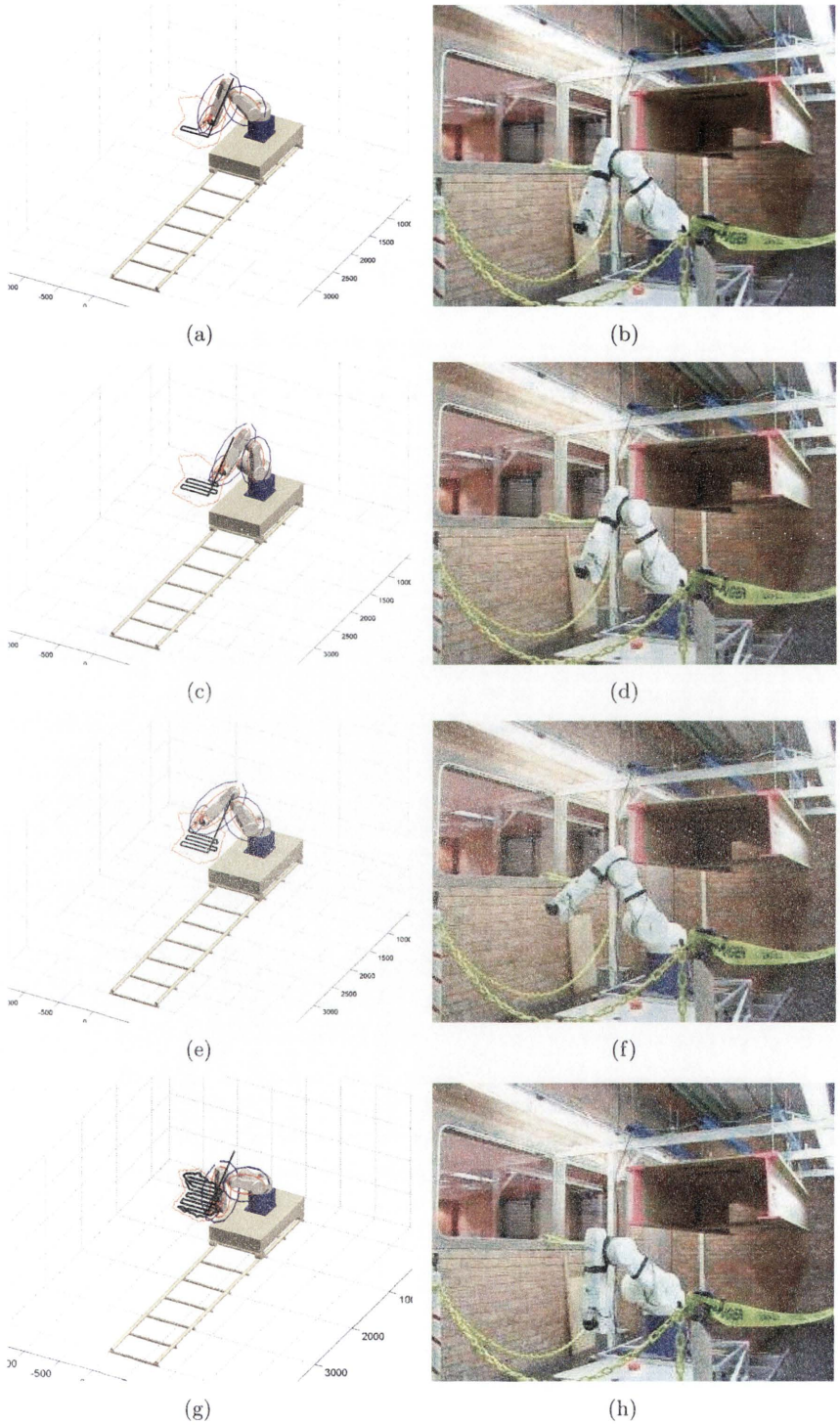


Figure 4.36: Comparing between simulation and experimental results for robot path planning (2)

4.4 Summary

This chapter has presented the simulation results, which verify that the boundary editing algorithm, the sequencing of blasting spots generation method, robot configuration transforming approaches and collision avoidance algorithm are effective. The experimental results provided in this chapter satisfy the requirement that the robot arm follows the generated trajectory accurately.

Chapter 5

Conclusion

The thesis has presented the study of generating an effective robotic trajectory to be used in a sand-blasting operation for steel bridge maintenance. The methodologies adopted have addressed the challenges faced with increasing the coverage of irregular blasting surfaces and the avoidance of collisions with obstacles. The feasibility of using the evolutionary computation technique in trajectory generation is demonstrated and the algorithm is modified as demanded by the problem domain. The overall effectiveness of the developed methods is verified by extensive simulations and experiments.

A hexagon-based topology is adopted to assign the blasting spots, to ensure that the blasted surface is fully covered while avoiding isolated patches of un-blasted areas. An editing procedure is further developed such that blasting spots are confined within the desired surface boundaries. This approach is crucial that high-pressure sand-blasting streams will not damage the surrounding structures underneath the bridge.

The trajectory generation problem is cast as the sequencing of the blasting spots that the blasting nozzle has to target in consequential steps. In this regard, a genetic algorithm is used to obtain a near-optimal solution. Since it is not desirable to repeat blasting at the same spot, the algorithm is modified it to alleviate the repetitions.

This is achieved by detecting and repairing, in the genetic terminology, the genes in a chromosome. The effectiveness of the trajectory is improved by considering objectives including short travel distances and small number of turns while manipulating the blasting nozzle. The outcome of this approach, hence, contributes to an improved productivity of sand-blasting operation.

After the sequencing of the blasting spots, the robot arm joint angles are then determined and issued to drive the arm for blasting. The transformation of task space to configuration space, i.e., from nozzle positions to arm joint angles, is conducted by the use of the inverse kinematics approach. However, due to singularities that frequently occur during the transformation, a genetic algorithm is again adopted to search for feasible joint angles when singularity occurs. This approach is considered as an attractive alternative in such demanding situations.

In practice, particularly in the complex environment underneath a steel bridge, the incorporation of obstacle avoidance functionality is necessary. To this end, a three-dimensional force-field method is used to safeguard the arm from colliding with obstacles. When collisions are encountered, the joint angles are re-selected from the pool of potential solutions maintained by the genetic algorithm. With the incorporation of collision avoidance, the robot arm can be deployed in the real-world operation environment as required.

A series of simulations and an experiment are conducted to verify the effectiveness of the generated trajectory for automatic sand-blasting. Surfaces to be blasted are determined from the sensed environment. Simulation cases include various surface locations and orientations with respect to the robot arm. These include surfaces in front, on top and on the sides of the robot arm. Large surface sizes are also chosen such that the ability to mitigate inverse kinematics singularities can be tested. Obstacles are placed

in the vicinity of the arm to test the avoidance functionality. An industrial robot manipulator, DENSO VM-6083D, is commanded in the experiment to follow a generated trajectory for sand-blasting. All results have demonstrated that the developed techniques are effective.

In summary, the methods presented in this thesis have provided feasible procedures to generate an effective robotic trajectory for sand-blasting. Directions for future research and developments may be stated as follows. Since the computational complexity increases with the size of the surface to be blasted, it is desirable to develop efficient computational methods independent of the surface size. Alternatively, methods to partition blasting surfaces into smaller sub-surfaces may be investigated such that complexity in trajectory generation could be further reduced.

Bibliography

[1] R. W. Rennison, "Civil Engineering Heritage", *Thomas Telford.* p. 98, 1996.

[2] N. Cossons,"The BP Book of Industrial Archaeology", *David and Charles.*, p. 246, 1987.

[3] S. K. Kim, J. S. Russell and K. J. Koo, "Construction Robot Path-Planning for Earthwork Operations", *J. Comput. Civ. Eng.*, 17(2), pp. 97-104, 2003.

[4] S. Moon and L. E. Bernold, "Vision-based Interactive Path Planning for Robotic Bridge Paint Removal," *J. Comput. Civil Eng.*, vol. 11, no. 2, pp. 113-120, 1997.

[5] G. Paul, D. K. Liu, N. Kirchner, S. Webb,"Safe and Efficient Autonomous Exploration Technique for 3D Mapping of a Complex Bridge Maintenance Environment", *Proc., 24th International Symposium on Automation and Robotics in Construction*, pp. 99-104, 2007.

[6] B. Akinci, M. Fischer and J. Kunz, "Automated Generation of Work Space Required by Construction Activates," *J. Constr. Eng. Manage.*, 128(4), pp. 306-316, 2002.

[7] Y. K. Cho, C. T. Hass, S. V. Sreenivasan, and K. Liapi, "Position Error Modelling for Automated Construction Manipulators", *J. Constr. Eng. Manage.*, 130(1), pp. 50-58, 2004.

[8] R. Navon and Y. Shpatnitsky, "Field Environments in Automated Monitoring of Road Construction", *J. Constr. Eng. Manage.*, 131(4), pp. 487-493, 2005.

[9] M. Skibniewski and C. Hendrickson, "Analysis of Robotic Surface Finishing Work Construction Site", *J. Constr. Eng. Manage.*, ASCE, 114(1), pp. 53-68, 1988.

[10] S. Tokioka, S. Ishigami, R. Sekiguchi, M. Wada, H. Inagaki and S. Sakai, "Robotized Painting by the Wall Surface-Finish Robot (FR-1)", *Proc., 6th int. Symp. On Robotics in Construction in San Francisco, Construction Industry*, Austin, Tex., pp. 531-538, 1989.

[11] K. M. Warne, "Investigation of Process Control Variables for an Automated Painting Machine", MS thesis, Civil Engineering Dept., Univ. of Texas at Austin, Austin, Tex, 1994.

[12] S. Suh, I. Woo and S. Noh, "Automatic trajectory Planning System (ATPS) for Spray Painting Robots", *J. Manuf. Syst.*, 10(5), pp. 396-406, 1991.

[13] N.Asakawa and Y.Takeuchi, "Teaching Less Spray-painting of Sculptured Surface by an Industrial Robot", *Proc., IEEE International Conference Robotics and Automation*, pp. 1875-1879, Apr, 1997.

[14] H. Chen, N. Xi, W. Sheng, M. Song and Y. Chen, "Automated Robot Trajectory Planning for Spray Painting of Free-form Surfaces in Automotive Manufacturing", *Proc., IEEE International Conference Robotics and Automation*, Washington, DC, Vol. 1, pp. 450-455, 2002.

[15] C. F. Huang and Y. C. Tseng, "The Coverage Problem in a Wireless Sensor Network", *Proc., of the 2nd ACM'03*, pp. 115-121, Sep, 2003.

[16] H. Zhang and J. C. Hou, "Maintaining Sensing Coverage and Connectivity in Large Sensor Networks", *Technical Report UIUCDCS-R-2003-2351*, pp. 89-124, 2005.

[17] Y. Xu, J. Heidemann and D. Estrin, "Geography-informed Energy Conservation for Ad Hoc Routing", *Proc., of ACM MOBICOM*, pp. 70-84, 2001.

[18] B. Chen, K. Jamieson, H. Balakrishnan and R. Morris "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks", *Wireless Networks*, Vol.8, no. 5, pp. 481-494, 2002.

[19] A. Cerpa and D. Estrin, "Ascent: Adaptive Self-configuring Sensor Networks Topologies", *Proc. of Infocom*, pp. 1-14, 2002.

[20] S. Slijepcevic and M. Potkonjak, "Power Efficient Organization of Wireless Sensor Networks", *Proc. IEEE International Conference Communications*, pp. 472-476, 2001.

[21] D. Tian and N. D. Georganas, "A Coverage-preserving Node Scheduling Scheme for large wireless sensor networks", *Proc First ACM International Workshop on Wireless Sensor Networks and Applications*, Georgia, GA, pp. 32-41, 2002.

[22] F. Ye, G. Zhong, S. Lu and L. Zhang, "PEAS: A Robust Energy Conserving Protocol for Long-lived Sensor Networks", *Proc. the 23nd International Conference on Distributed Computing Systems (ICDCS)*, pp. 1-10, 2003.

[23] J. K. Antonio, "Optimal Trajectory Planning for Spray Coating", *Proc. IEEE International Conference Robotics and Automation*, pp. 2570-2576,May, 1994.

[24] J. K. Antonio, R. Rbhadran and T. L. Ling, "A Frame Work for Optimal Trajectory Planning for Automated Spray Coating", *Int. J. Robot. Automat.*, Vol.12, no. 4, pp. 124-134, 1997.

[25] H. P. Cheng, N. Xi, W. H. Sheng, M. M. Song and Y. F. Chen, "CAD-based Automated Robot Trajectory Planning for Spray Painting of Free-form Surfaces", *Industrial Robot*, pp. 426-433, 2002.

[26] M. A. S. Arikan, and T. Balkan, "Modelling of Paint Flow Rate Flux for Elliptical Paint Spray by Using Experiment Paint Thickness Distributions", *Industrial Robot*, pp. 60-66, 2006.

[27] W. H. Hang, "Optimal Line-sweep-based Decompositions for Coverage Algorithms", *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 27-32, 2001.

[28] W. H. Sheng, H. P. Cheng, N. Xi and Y. F. Chen, "Tool Path Planning for Compound Surface in Spray Forming Process", *Automation Science and Engineering*, pp. 240-248, 2005.

[29] W. B. Sheng, N. Xi, H. P. Cheng, Y. F. Chen and M. M. Song, "Part Geometric Understanding for Tool Path Planning in Additive Manufacturing", *Proc., IEEE International Conference Robotics and Automation*, pp. 1515-1520, July, 2003.

[30] W. H. Sheng, H. P. Cheng, N. Xi, J. D. Tan and Y. F. Chen, "Optimal Tool Path Planning for Compound Surface in Spray Forming Process", *Proc., IEEE International Conference Robotics and Automation*, pp. 45-50, Apr. 2004.

[31] W. H. Sheng, G. Tewolde and H. P. Cheng, "Tool Path Integration for Spray Forming Processes Using a Genetic Algorithm", *Proc. IEEE, Advanced Robotic*, pp. 159-164, 2005.

[32] Y. Gou and M. Balakrishnan. "Complete Coverage Control for Nonholonomic Mobile Robot in Dynamic Environment", *Proc., IEEE International Conference Robotics and Automation*, pp. 1704-1709, 2006.

[33] R. Paul,"Robot Manipulators, Programming, and Control", *MIT Press, Cambridge. Mass*, 1981.

[34] J. Korein and N. I. Badler, "Techniques for Goal Directed Motion", *Proc. IEEE Comput. Graph*, pp. 71-81, 1982.

[35] M. Girard and A. A. Maciejewski, "Computational Modeling for The Computer Animation of Legged Figures", *ACM Comput. Graph.*, pp. 263-270, 1985.

[36] A. Witkin, K. Fleewherk and A. H. Barr, "Energy Constraints on Parameterized Models", *ACM Comput. Graph.*, pp. 225-232, 1987.

[37] R. Barzel and A. H. Barr, "A modelling System Based on Dynamic Constraints", *ACM Comput. Graph.*, pp. 179-188, 1988.

[38] A. Witkin and W. Welch,"Fast Animation and Control of no Rigid Structures", ACM Comput. Graph., pp. 243-252, 1990.

[39] T. Alameldin, M. Palis, S. Rajaskaran and N. I. Badler, "On The Complexity of Computing Reachable Workspaces for Redundant Manipulators", *Boston'90 Symposium on Advances in Intelligent Systems. Intelligent Robots and Computer Vision IX: Algorithms and Complexity,* pp. 217-225, 1990.

[40] J. Holland, "Adaptation in Natural and Artificial Systems", *University of Michigan Press*, 1975.

[41] D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning", *Addison-Wesley Publication Co.*,1989.

[42] D. Powell and M. M. Skolnick, "Using Genetic Algorithms in Engineering Design Optimization with Nonlinear Constraints", *Proc., 5th Int. conf. on Genetic Algorithms,* pp. 424-431, 1993.

[43] H. Adeli and N. T. Cheng, "Integrated Genetic Algorithms for Optimization of Space Structures", *J. Aerosp. Engre.*, ASCE, pp. 315-328, 1993.

[44] G. M. Wang, X. J. Wang, Z. P. Wan and Y. L. Chen, "Genetic Algorithms for Solving Linear Bilevel Programming", *IEEE Computer Society,* pp. 920-924, 2005.

[45] T. Shibata, T. Abe, K. Tanie and M. Nose, "Motion Planning by Genetic Algorithms for A Redundant Manipulator Using a Model of Criteria of Skilled Operators", *Intelligent Systems*, Vol. 102, pp. 171-186, 1997.

[46] X. Liao and G. G. Wang, "Evolutionary Path Planning for Robot Assisted Part Handling in Sheet Metal Bending", *Robotics and Computer Integrated Manufacturing*, Vol. 19, pp. 425-430, 2003.

[47] L. Tian and C. Collins, "An Effective Robot Trajectory Planning Method Using a Genetic Algogrithm", *Mechatronics*, Vol. 14, pp. 455-470, 2004.

[48] W. M. Yun and Y. G. Xi, "Optimum Motion Planning in Joint Space for Robots Using Genetic Algorithms", *Robotics and Autonomous Systems*, Vol. 18, pp. 373-393, 1996.

[49] H. Mahjoubi, F. Bahrami and C. Lucas, "Path Planning in an Environment with Static and Dynamic Obstacles Using Genetic Algorithm: A Simplified Search Space Approach," *Proc. 2006 IEEE Congress on Evolutionary Computation*, Vancouver, BC, Canada, July 2006, pp. 2483-2489.

[50] J. D. Schaffer, "Some Effects of Selection Procedures on Hyper Plane Sampling by Genetic Algorithms", *Genetic Algorithms and Simulated Annealing*, L. Davis. Ed. Pitman, 1987.

[51] R. P. Paul,"Robot Manipulators: Mathematics, Programming, and Control, 1st edition", *MIT Press Cambridge, MA, USA*, 1982.

[52] P. Chotiprayanakul, D.K. Liu, D. L. Wang and G. Dissanayake (2007), "A 3-dimensional Force Field Method for Robot Collision Avoidance in Complex Environments", *Proc., 24th International Symposium on Automation and Robotics in Construction* , pp. 139-145, 2007.

[53] P. Chotiprayanakul, D.K. Liu, D. L. Wang and G. Dissanayake, "Collision-free Trajectory Planning for Manipulators Using Virtual Force Based Approach", *Proc.of the International Conference on Engineering, Applied Sciences, and Technology, CD-ROM*, 2007.