"GENETIC-BASED OPTIMISATION TECHNIQUE FOR THE DEVELOPMENT OF AUTOMATED INSPECTION AND RESTORATION SYSTEMS FOR BRIDGES"

ANNA MARIA MCCREA

MENG. MSC.

This thesis is submitted for the Degree of Doctor of Philosophy

City University

Department of Civil Engineering

August 1999

To my Parents, for bringing me up in an atmosphere of academic endeavour towards constant self-improvement.

To my Grandmother, for spinning the magic.

To Peter, for his patience.

TABLE OF CONTENTS

APPENDICES

ABSTRACT

Automation and robotics are receiving significant attention in the field of inspection and restoration of steel bridges. However, the success level of the field implementations depends on numerous technological factors. This dissertation addresses aspects of the design, development and subsequent implementation of such on-site devices. The restoration process poses a high level of health hazard and carries environmental pollution risk. For these reasons, it is high on the consideration list for automation. The varied scale and geometry of bridges are some of the limiting conditions for performing the inspection and restoration tasks. Further aspects of concern are access provisions, the diversity of tasks required in the assessment and restoration of a bridge and compatibility between the operational characteristics of the automated device, tasks layout and direction of approach. The key factors, which arise as a result of the above analysis, are access, mobility, navigation, manipulation, probe change and control.

In order to efficiently produce design alternatives, based on the industry (customers and designers) requirements, the engineering design framework is adopted. Due to the growing complexity of the required devices, new methodologies and approaches are needed. This dissertation presents a design methodology to generate alternatives for further considerations. The author's work combines: (i) research and suitability assessment of the existing enabling technologies, (ii) extensive task selection and analysis, (iii) incorporation of the industry requirements for generating the set of design criteria, and (iv) an innovative application of Genetic Algorithms.

GA is used as a tool for simultaneous optimisation of the robot's kinematic parameters, based on the criteria of collision and singularity avoidance, percentage of coverage, productivity and dexterity. Analysis and justification of a two-step approach is presented, with the former combining all the parameters, and the latter handling the chosen criteria. The methodology is then tested and verified on an existing construction robot (MPIR) from Technion. Finally, it is applied to two case studies, spherical and articulated manipulators performing a range of restoration activities on a selected bridge geometry model. A sensitivity analysis was also carried out on each case study in order to identify areas where improvements could be made.

In general, the methodology is successful in choosing the more task-suitable manipulator and optimising the ranges of its kinematic parameters. This could be extended to optimise other parameters according to a set of alternative criteria. In doing so, it can bridge over several phases of the engineering design with a single approach.

Acknowledgements

I would like to thank my supervisor, Mr. Denis Chamberlain, for organising this research work and supporting me with invaluable assistance and constant guidance throughout. Denis's inspiration, patience and friendship were crucial in helping me to complete my thesis.

I would also like to thank my second supervisor, Dr. Shamil Naoum, for his critical advice and input into the overall structure of the thesis.

My special thanks and gratitude go to Dr. Ronie Navon from Technion, Israel, my external supervisor. Dr. Navon has initiated and guided the methodology, Genetic Algorithms optimisation and the verification parts of my research. His encouragement, help and serenity have added an invaluable contribution to the results of this research.

Part of this research was financed with the aid of financial support from the British Council and the Israel Ministry of Science and Technology and this support is gratefully acknowledged.

PUBLICATIONS

Refereed Journals

Navon, R. and McCrea, A.M. (1997) Selection of Optimal Construction Robot Using Genetic Algorithm, Journal of Computing in Civil Engineering - ASCE, New York, USA:11(3), p.1-9.

McCrea, A.M. and Navon, R. (1999) GA Efficiency in Parametric Selection of Bridge Restoration Robot, Journal of Computing in Civil Engineering - ASCE, New York, USA, to be submitted in November 1999.

McCrea, A.M., Chamberlain, D.A. and Navon, R. (1999) Automated Restoration of Steel Bridges - Critical Review of Methods and Existing Technology, Automation in Construction, Elsevier, Netherlands, to be submitted in November 1999.

Refereed Conferences

McCrea, A.M. (1997) GA Performance in Parametric Selection of Bridge Restoration Robot, Proceedings of the 14th ISARC, Pittsburgh, USA.

McCrea, A.M. and Navon, R. (1996) Construction Robot Development Using genetic Algorithm Technique, Proceedings of the 13th ISARC, Tokyo, Japan.

McCrea, A.M. (1995) Health, Safety and Environmental Issues in Automated Maintenance of Bridges, Proceedings of the 4th Conference on Inspection, Appraisal, Repairs and Maintenance of Buildings and Structures, Hong Kong.

McCrea, A.M. and Chamberlain, D.A. (1995) Worker Safety and Environmental Protection in Automated Steel Bridge Restoration, Proceedings of the 12th ISARC, Warsaw, Poland.

McCrea, A.M. (1994) Productivity Analysis of Bridge Inspection Robots, Proceedings of the 11th ISARC, Brighton, UK.

McCrea, A.M. (1994) Automation Potential in Concrete Inspection, Proceedings of the 19th Conference on Our World in Concrete and Structures, Singapore.

McCrea, A.M. and Chamberlain, D.A. (1993) Towards the Development of a Bridge Inspection Robot, Proceedings of the 10th International Symposium on Automation and Robotics in Construction (ISARC), Houston, USA.

LIST OF FIGURES

LIST OF TABLES

Notation and Variable Definitions

A_{ij}	rotational part of total transformation matrix indicating the orientation of the wrist for given values of parameters
C_{3k}	series of K points representing the contours of the obstacle
C	tensor identifying boundaries of the collision envelope (here twenty)
$\theta_1,\theta_2,\theta_3$	angles of rotation of the major configuration revolute joints (RRR)
θ_1,θ_2,L_3	angle of rotation of the major configuration revolute joints and projection length of prismatic joint
$\theta_4,\theta_5,\theta_6$	angles of rotation of the revolute joints in minor configuration (last three joints
E_{3i}	trajectory of N points of M sub-paths followed by the elbow
f_1, f_2	fitness functions from first and second stage respectively
I	penalty value indicating singularity or collision
L_1, L_2	lengths of first and second links, respectively ($L_1+L_2=1$)
N, npopsiz	population size
P_{3i}	trajectory of N points of M sub-paths to be followed in the workspace
P	tensor encompassing all points along the trajectories (here twelve)
R_{ij}	rotation matrix of unit ortho-normal bases of a point within trajectory indicating the required orientation of approach
R	tensor of the required dexterity
T_{3i}	trajectory of N points of M sub-paths produced by end effector
T	tensor encompassing all the tool locations resulting from direct kinematics
t_l	task completion time
Σdist	sum of the distances between P and T
D	tensor of the rotational part of the total transformation matrix
A,B	'weighting' coefficients allocating current importance rate between dexterity and productivity
Σt_m	sum of all the times required to reach all points along P

nparam	number of parameters (groups of bits) of each individual, (nparam to match the number of values in the parmin, parmax and nposibl input arrays)
pcross	probability of single point or uniform crossover
pmutate	probability of jump mutation, typically set = 1/npopsiz
pcreep	probability of creep mutation, typically set this to (nchrome/nparam)/npopsiz
nchrom	number of chromosomes
maxgen	maximum number of generations to run by the GA
irestrt	0 for a new GA run, or for a single function evaluation = 1 for a restart continuation of a GA run
idum	initial random number seed for the GA run. Must equal a negative integer, e.g. idum=-1000
itourny	no longer used, GA presently set up for only tournament selection
ielite	0 for no elitism (best individual not necessarily replicated from one generation to the next); = 1 for elitism to be invoked (best individual replicated into next generation); elitism is recommended
iunifrm	0 for single-point crossover; = 1 for uniform crossover
iniche	0 for no niching; = 1 for niching
nchild	1 for one child per pair of parents; = 2 for two children per pair of parents
iskip	0 for normal GA run (this is standard); = number in population to look at a specific individual or set of individuals. Setting iskip-0 is only used for debugging purposes
iend	0 for normal GA run (this is standard),= number of last population member to be looked at in a set of individuals. Setting iend-0 is only used for debugging purposes and is commonly used in conjunction with iskip
nowrite	0 to write detailed mutation and parameter adjustments; = 1 to not write detailed mutation and parameter adjustments
parmin	array of the minimum allowed values of the parameters
parmax	array of the maximum allowed values of the parameters
nposibl	array of integer number of possibilities per parameter. For optimal code efficiency set nposibl=2**n, i.e. 2, 4, 8, 16, 32, 64, etc.

nichflg	array of 1/0 flags for whether or not niching occurs on a particular parameter. Set to 0 for no niching on a parameter, set to 1 for niching to operate on parameter. The default value is 1, but the implementation of niching is still controlled by the flag iniche.
microga	0 for normal conventional GA operation; = 1 for micro-GA operation (this will automatically reset some of the other input flags)
best	the best fitness of the generation
child	the floating point parameter array of the children
creep	+1 or -1, indicates which direction parameter creeps
del	square root of del2
del2	sum of the squares of the normalised multidimensional distance between member j and all other members of the population
delta	del/nparam
diffrac	fraction of total number of bits which are different between the best and the rest of the micro-GA population. Population convergence arbitrarily set as diffrac<0.05
fbar	average fitness of population
fitness	array of fitnesses of the parents
firsum	sum of the fitnesses of the parents
g0	lower bound values of the parameter array to be optimised. The number of parameters in the array should match the dimension set in the above parameter statement
g1	the increment by which the parameter array is increased from the lower bound values in the g0 array. The minimum parameter value is g0 and the maximum parameter value equals g0+g1*(2**g2-1), i.e. g1 is the incremental value between min and max.
ig2	array of the number of bits per parameter, i.e. the number of possible values per parameter. For example, ig2=2 is equivalent to 4 (=2**2) possibilities, ig2=4 is equivalent to 16 (=2**4) possibilities.
ig2sum	sum of the number of possibilities of ig2 array.
ibest	binary array of chromosomes of the best individual.
ichild	binary array of chromosomes of the children.

icount	counter of number of different bits between best individual and other members of micro-GA population.
icross	the crossover point in single-point crossover
indmax	maximum # of individuals allowed, i.e. max population size
iparent	binary array of chromosomes of the parents
istart	the generation to be started from
Ibest	the member in the population with the best fitness
ielite	a counter which tracks the number of bits of an individual which match those of the best individual
iend	used in conjunction with iend for debugging
jstart	used in conjunction with iskip for debugging
kount	a counter which controls how frequently the restart file is written
kountmx	the maximum value of kount before a new restart file is written; presently set to write every fifth generation. Increasing this value will reduce I/O time requirements and reduce wear and tear on your storage device
kelite	kelite set to unity when jelite=nchrome, indicates that the best parent was replicated amongst the children
mate1	the number of the population member chosen as mate1
mate2	the number of the population member chosen as mate2
nchrmax	maximum # of chromosomes (binary bits) per individual
nchrome	number of chromosomes (binary bits) of each individual
ncreep	# of creep mutations which occurred during reproduction
nmutate	# of jump mutations which occurred during reproduction
nparmax	maximum # of parameters which the chromosomes make up
npossum	sum of the number of possible values of all parameters
paramav	the average of each parameter in the population
paramsm	the sum of each parameter in the population
parent	the floating point parameter array of the parents
pardel	array of the difference between parmax and parmin
rand	the value of the current random number
sigshar	floating point equivalent of nparam
sumshar	the scaling factor to be applied to the fitness of each individual based on a triangular sharing function
disttlj	sum of the distances between end effector and all path points
tool	co-ordinates of the tool

elbow	co-ordinates of the elbow
ll	penalty function
devv	standard deviation
path	path co-ordinates
collis	corners of the collision envelope's co-ordinates
ptx, pty, ptz	intersection co-ordinates in x,y,z between the collision plane and elbow or tool
distance	sum of dist
dist	distance between the direct kinematics calculated position of the tool and path points
den	distance between elbow and tool co-ordinates
rotat	required rotational matrix for RRR (or RRP) with RPR and RPY wrists
dext	calculated current rotational matrix for RRR (or RRP) with RPR and RPY wrists
sumttl	sum of all times to reach all the path points
t,t1,t2	time to reach path points
partfit1	part of the fitness function due to dexterity
partfit2	part of the fitness function due to time

DECLARATION

I grant power of discretion to the university library to allow this thesis to be copied in whole or in part without further reference to me. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.

CHAPTER 1 - INTRODUCTION

The thesis presents a new approach to the optimisation of a construction robot, applying the Genetic Algorithm (GA) method, as the optimisation tool and using the engineering design (ED) process, as the framework. This new approach is a combination of (i) fresh insight into linking of the particular phases of the engineering design process, (ii) careful analysis and selection of parameters from a list of previously assembled engineering requirements and embedding them into a candidate robot's description and representation, (iii) immediate, innovative evaluation of the candidate solutions and (iv) use of the advanced knowledge of the genetic algorithm behaviour in computations. The combination of these four features yields a powerful, new method of GA application as the search and selection mechanism. The primary limitation of this framework is the availability of computing resources, which is attempted to be overcome by using the developments within the ways of behaviour and operating of GA itself.

This particular approach represents an attempt to devise a design methodology, which can be used across a broad range of application areas. This thesis explores the use of GA in two applications: robot's kinematic parameters optimisation, and configuration (RRP and RRR) optimisation and, later, suggesting the course of future work for continuing the development of this methodology.

The introductory chapter first presents the author's motivation for pursuing the line of research and then, a formal problem statement. It next presents the aims, objectives and the hypotheses of the research. The following two sub-sections outline the scope of the research and the approach (methodology) devised. The organisation of the thesis concludes this chapter.

1.1 Motivation

The recent motivation for this work comes from the author's experience in the field of bridge design, maintenance techniques and corresponding equipment. It is the author's opinion that a rigorous evaluation of alternatives is not used in the design of the overwhelming majority of partially or fully automated systems conceived for restoration of bridges. This can be directly linked to poor understanding and inadequate application of the methods and procedures of the engineering design process. Even in the best studies of robot design, the resulting systems are limited in scope, considering only a few out of large number of possible alternative configurations and evaluated for limited applications. As the tasks assumed for robots grow in complexity, together with the variety in the design and geometry of the structural environment (bridges in particular), due to the innovative construction techniques and materials, the need to design better automated systems will increase. To help in the design of these systems, which in some cases have no precedent from which to work, a new methodology for achieving an improved design, is required.

1.2 Problem Statement

1.2.1 Rationale

A vast and costly inspection, repair and maintenance requirement exists for steel structures in Europe. To carry out these operations on structures such as tall buildings, storage tanks, chimneys and long span bridges, can be costly and hazardous. Therefore, the above activities are an important target for automation and robotics research. Establishing the need initiates the engineering design process. Supported by the framework of ED, the following actions follow: (i) understanding of the problem, (ii) generating potential solutions (concepts) and (iii) evaluating solutions. For the purpose of this research, steel bridges are selected as the target structures.

To avoid building expensive and abortive trial hardware, an engineering design process is adopted, initially using elements of Quality Function Deployment (QFD) [Ullman, 1997] to determine engineering requirements (parameters and criteria), then employing Genetic Algorithm (GA) as the parameter optimisation technique, followed by method verification, simulation and experimentation. The development of an effective remotely operated automated inspection and restoration facility, focused on bridges, is aimed at in this thesis, indicating the high automation potential of these activities. Existing technology is transferred and adapted from industries, such as defence and nuclear, where their hazardous environment has led to significantly more advanced handling technology than that evolved in the construction industry. The sub-system technologies include: (i) a modular access and waste handling system, (ii) a flexible lightweight robotic manipulator, (iii) a manipulator carriage, and (iv) manipulator compatible tools for inspection, surface preparation (recyclable materials) and painting.

1.2.2 Gap of Knowledge

Workers employed in the steel structures' restoration industry suffer from high physical strain and are exposed to toxic dust and dangerous materials. The restoration process itself carries also a high risk of environmental pollution.

Whilst the sub-systems technology exists, which could improve the safety of the process, its adoption and integration awaits further investigation. Similar problems face other industries, which deal with hazardous or difficult to access environments. Expensive, purpose built access systems are commonly employed rather than low cost, reconfigurable, high performance modular systems, appropriate for flexible automation applications. Also, despite the similarity with tasks in other industries, advantage has not yet been taken of robotic manipulators, which can meet the requirements for tool delivery and handling. At the same time, methodology is also needed which can quickly identify the best solution for the required task, in terms of configuration and kinematic parameters, without having to choose from the vast range of techniques and methods potentially available for use in every stage of the ED. This methodology has not only to benefit from the formalised approach of the engineering design process, but also combine and accelerate the conceptual and embodiment (evaluation) design. The author initially pursued her investigations in the domain of conceptual design of the robot for bridge restoration. However, in her design task, the key design variables have been identified (by using elements of QFD and collaborating with the industry experts and practitioners), so a more precise characterisation of the problem is achieved. This corresponds to "parametric design at a system level of abstraction" [Gelsey, et al., 1998]. Within the parametric design, a choice of the optimisation tool is required, as well as, its ingenious application, which optimises the robot's task-focused performance. There are several optimisation procedures, such as enumeration, machine learning, or artificial intelligence, which would lead to global optimum, but they require excessive computing capacity, time to set them up, and careful monitoring of the optimising process. Therefore, author's attention has concentrated on Genetic Algorithms, as they are capable of exploring vast search spaces, arriving at optimal values of simultaneously optimised numerous parameters under, sometimes, competing objectives. The development of the GA application is however, a problem-particular affair and the most successful utilisation and performance require a fresh insight and in-depth, unique issue analysis. The GA use in task related optimisation of robot's kinematic parameters still has not been explored or utilised.

1.3 Research Objectives

The aim of this research is to arrive at the optimal configuration type and optimal values of kinematic parameters of the restoration robot for steel bridges. The optimisation is carried out for the specified range of tasks, under a set of pre-determined criteria. The above aim will be achieved through the following objectives:

i. Clarify and apply the engineering design process to the robot's selection and explain the positioning of the optimisation method within ED with the correlation between the expected outcome of consecutive stages and the application integrity.

ii. Use the QFD method to assemble a list of customers' and manufacturers' requirements, completing benchmarking and generate engineering requirements.

iii. Critically analyse the restoration tasks of large span steel bridges, through combining bridge geometry with the specification of restoration techniques.

iv. Review existing enabling technologies for possible application in the automated device.

v. List, assess and grade the choice of criteria (emerging from engineering requirements) for the optimal design of an inspection and maintenance robot. Analyse and establish design parameters, based on the criteria selection.

vi. Analyse and apply Genetic Algorithm (GA) as an optimisation tool, appropriately representing robot's parameters and employing effective evaluation methods, based on the criteria.

vii. Computerise the optimisation procedure with FORTRAN.

viii. Test the results on the verified case study of an automated system and based on the success of the verification, apply the method to the bridge inspection and restoration AF (automated facility).

ix. Confirm the optimisation outcome using robot simulation software (GRASP).

x. Anticipate the continuation to the method.

1.4 Hypothesis

The main hypothesis of the research is:

There is a need to develop the methodology for selection of the optimal robot for inspection and restoration of variety of steel bridges. In the optimisation phases, it is feasible and beneficial to employ Genetic Algorithm (GA), as a technique for generating and evaluating automation system concepts through task-based optimisation of parameters considering: (i) alternative discrete values of design kinematic parameters, within a specified range, for given configuration and (ii) simultaneous changes in values of a number of parameters under specified conditions (criteria).

The main hypothesis is further expressed as more detailed sub-hypotheses:

i. The proposed methodology for selecting and optimising an automated device spans over several phases of the engineering design process and therefore, only requires a single framework to carry the out the design into an advanced stage.

ii. Effective, task-based evaluation within the GA results in a more accurate and versatile device.

iii. Possibilities exist to match the existing enabling technologies with the results of the optimisation, which can leads to assembly of the proposed automated facility (AF) from existing components.

1.5 Scope of Research

Although, the ED framework is being employed as the framework for this research, certain practical aspects are not applicable. The QFD method is only indicatively introduced in order to identify and rank customer requirements - see Section 3.3.1 and Table 3.1. Steel bridges are classified, grouped and a benchmark library assembled. However, the GA experiments are set for one standard workspace, selected out of the library. NDT and surface preparation methods are fully investigated, with one tool path selected. The main tool handling configurations are brought together and evaluated with two of them subject to GA optimisation. From the result of customers' requirement survey and competition benchmarking, the list of engineering requirements emerges, but again, only a limited amount of this is translated to the criteria/parameter relationship, serving as constraints in the GA model. These restrictions are inevitable and do not limit possible widespread applications. The potential is highlighted in the recommendation for further studies, Section 11.5.

1.6 Research Methodology

The approach to the research is based on the following steps:

i. Site visits to Kessock Bridge, while under restoration, with interviews to highlight the characteristics of the process, access problems, health hazard and environmental impact, while targeting the process for automation.

ii. Interviews with potential clients (Local Authorities and bridge restoration companies) to assemble a list of customer requirements for the system together with ranking.

iii. Participating in five Symposia on Robotic and Automation in Construction between 1993-1997, attending International Conference on Remote Techniques for Hazardous Environments and International Conference on Inspection, Appraisal, Repairs and Maintenance of Buildings and Structures, as well as an exhibition on Industrial Maintenance 97 in Ahoy-Rotterdam, see a list of publications in the beginning of the thesis.

iv. An extensive literature survey, carried out in areas of:
   1. Design methodology (Ch.2),
   2. Design and optimisation methods for the engineering systems (Ch.2),
   3. Genetic Algorithm as optimisation tool and their potential and application to engineering design (Ch.2),
   4. Available computing tools - FORTRAN (Ch.8) and GRASP (Ch.10).

v. An extensive availability survey (competition benchmarking) was carried out to assess and evaluate:
   1. Inspection and restoration process and their automation potential (Ch.5),
   2. Existing automation and robotics technology (Ch. 6 and Appendix C.).

vi. Developing the GA driver in FORTRAN, incorporating the variables and constraints.

vii. Verification of the method and carrying out the experiments.

viii. Setting up the simulation model in GRASP (robotic simulation software) in order to further confirm the results obtained using GA.

The research methods are outlined with their inter-relationships in **Figure 1.1.**

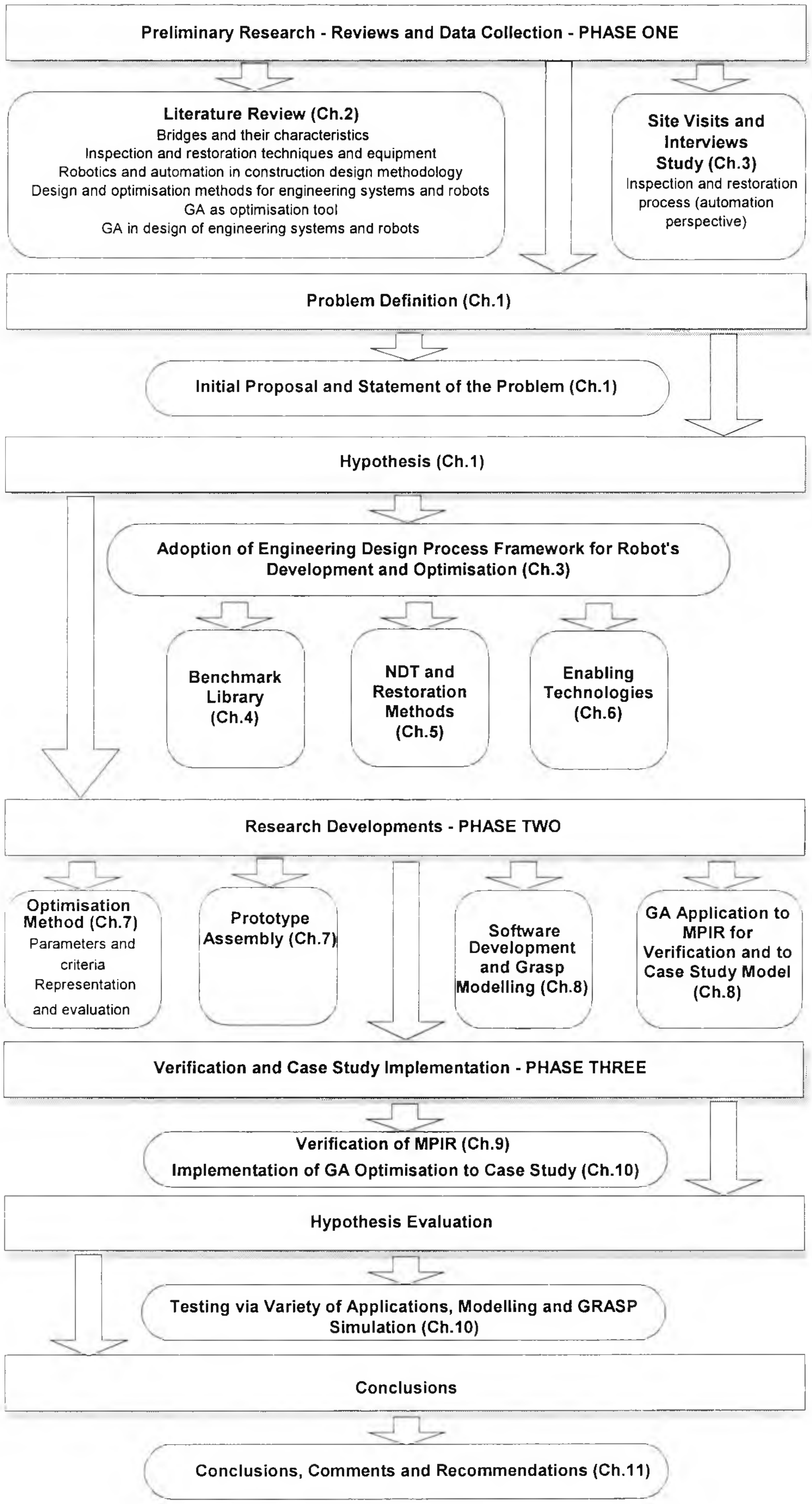

Preliminary Research - Reviews and Data Collection - PHASE ONE
Literature Review (Ch.2)
Bridges and their characteristics
Inspection and restoration techniques and equipment
Robotics and automation in construction design methodology
Design and optimisation methods for engineering systems and robots
GA as optimisation tool
GA in design of engineering systems and robots
Site Visits and Interviews Study (Ch.3)
Inspection and restoration process (automation perspective)
Problem Definition (Ch.1)
Initial Proposal and Statement of the Problem (Ch.1)
Hypothesis (Ch.1)
Adoption of Engineering Design Process Framework for Robot's Development and Optimisation (Ch.3)
Benchmark Library (Ch.4)
NDT and Restoration Methods (Ch.5)
Enabling Technologies (Ch.6)
Research Developments - PHASE TWO
Optimisation Method (Ch.7)
Parameters and criteria
Representation and evaluation
Prototype Assembly (Ch.7)
Software Development and Grasp Modelling (Ch.8)
GA Application to MPIR for Verification and to Case Study Model (Ch.8)
Verification and Case Study Implementation - PHASE THREE
Verification of MPIR (Ch.9)
Implementation of GA Optimisation to Case Study (Ch.10)
Hypothesis Evaluation
Testing via Variety of Applications, Modelling and GRASP Simulation (Ch.10)
Conclusions
Conclusions, Comments and Recommendations (Ch.11)

Figure 1.1 Research Methods

Research methods listed in Section 1.6 and in Figure 1.1 reflect well the development phases within the research. The preliminary research encompasses all the preparatory activities, such as literature review, interviews, information collection, data assembly and critical reviews of the above. Then the research development phase follows with the original input from the author, with the research methods, such as prototype assembly, and evolution of the optimisation approach, application of GA to optimisation model of the case study. Once all the aspects of the research are developed, the third phase commences with testing on the verification model, implementation and experiments. Initially, the method implementation is carried out for the MPIR for verification. Once successful, and therefore yielding credible results, the method can be further implemented to the case study and experimented with, on variety of aspects. Hypothesis evaluation serves as a summarising phase.

1.8 Thesis Outlook and Structure

The original plan of this research was to develop Automated Inspection Facility (AIF) to operate on steel bridges, using computer simulation. The primary approach taken was to determine, by computer simulation and analysis, the optimum form of the AIF and its access system, including the control, manipulation, sensing, and data handling requirements in the performance of benchmark tasks. The main drawback was the verification of the quality of the outcome of the simulation software. Another problem with this approach was the fact that the computer models are non-interactive and therefore, combining the models' features proved to be difficult. The inspiration to further investigate the potential of GA and new direction towards solving the above difficulties came from reading the article by Goldberg D.E. [1994], "Genetic and Evolutionary Algorithms Come of Age".

Trying to obtain the independent solution, using the ideas of genetic computation to intelligently search the robot configuration space solves the first problem. The second problem is solved by realising that robotic features are actually higher level constructs of more basic features. These features are easier to manipulate and can be assembled in a large number of ways that lead to a large space of possible robot configurations. What really helped in combining all the elements, was the realisation that biological mechanisms are developed in the same general manner, and that the approach has been proven to work. The approach taken in formulating this work also mirrors the biological approach in the sense that the first problems to be solved are relatively straightforward, and that once the methodology has successfully solved some relatively straightforward problems, more difficult tasks can be tackled.

The author also felt that the approach to this new methodology requires a certain framework to achieve maximum effectiveness. This is found by application of methods and techniques of the phases of the engineering design process. Particularly the initial stage of specification development and one of its methods - Quality Function Deployment (QFD) proved extremely helpful in fully identifying customers' and manufacturers' requirements, competition benchmarking and finally arriving at a list of engineering requirements, these easily translated into criteria and relevant parameters. Discovering and applying this framework (ED) and further investigating its full potential led to another realisation. The newly developed methodology could be modified and extended to span over conceptual and embodiment stages of ED.

This thesis is divided into three main parts, and the fourth part includes the Appendices. At this point, the author would like to draw the reader's attention to the fact that the theoretical, general information about steel deterioration (corrosion) and Genetic Algorithms is included in Appendices A and B. She felt that these parts are not compelling part of this dissertation and they do not present any selective or analytical original research or contribution. This information, however, may help the reader to get re-acquainted with some basic introductory information.

The first part of the main thesis sets up the particular ED framework with its tasks and their objectives, within which the system's development is carried out and then encompasses all the activities and procedures under the QFD method leading to specification development. It looks into bridges and their deterioration due to corrosion, assembles the benchmark library and lists customer requirements for the product. It also includes the review of the inspection techniques and characteristics of the restoration process, with description of all tools and methods applied commonly to these tasks, as well as all new research into automated systems. The developmental and commercially applied automated and semi-automated systems are also listed and assessed, as well as independent access methods and systems, and tool handling configurations. This part assembles all the preparatory and information gathering and review.

The second part, which is the developmental one, feeds off the findings from the first phase, but includes all the original research and commences with assembling of the model of the prototype, from which the case study is modelled. Then the list of engineering requirements is assessed and transferred into a set of criteria and corresponding parameters. Then the optimisation method is developed. Further, the

Genetic Algorithm (GA) is applied to the method, using the case study. This is carried out in parallel with the software development in FORTRAN and GRASP model development.

Within the third, and final part of the main body of the thesis the author implements the method onto the verification robot and then onto the case study. This section gives the verification of findings (obtained using GA) by applying the method to a previously considered robot (MPIR) [Navon, 1995] and the confirmation by running computer simulation using the GRASP software. Then the author discusses the applications of this work, in particular, an innovative method for appraisal of system's parameters, and a new approach to formulation of the representation and evaluation methods. It develops into the presentation of the application of GA to two different problem domains, namely, the optimisation of the kinematic parameters with respect to the chosen criteria and selection of the superior configuration (for the given task). Finally, the thesis addresses the hypothesis, and a summary and recommendations for further research conclude the thesis.

The fourth part consists of references and all the appendices, which constitute of all the auxiliary reading, copies of the software, acronyms and original MPIR data.

In order to come up with a successful design of a product, the theory and approaches recommended within ED need to be understood, selected and applied. Chapter 3 sets up the particular ED environment and introduces and justifies the methods selected and restrictions imposed. To arrive at the winning design, the first step is to determine the means for clearly expressing the devices being designed. The method chosen must be unambiguous, amenable to computer manipulation, domain independent and able to express a wide variety of concepts. Chapter 4 looks into the working environment, classifying and assessing steel bridges, their deterioration modes and identifying the library of benchmarks. In Chapter 5, non-destructive testing (NDT) tools and methods are discussed with emphasis put on their methods of handling and automation applicability together with reviews of the existing automated systems from the same perspective. Chapter 6 examines the task from the robotisation viewpoint and the enabling technologies such as access systems and tool handling configurations. This concludes the preparatory stages.

Chapter 7 sets out to assemble the prototype options, followed by case study outline and then, the criteria and parameters selection. Then the approach to optimisation on a case study is developed. Chapter 8 applies a tool (GA) and the best GA approach to the case study model, which provides the previously described attributes within the framework of the engineering design process (the basic GA theory can be found in Appendix B). The next step is to introduce the computing tool, by first presenting the basic computing tools for genetic computation, then exploring their applicability and formulating modifications to suit the new application of GA into the design of robots. This process finalises Chapter 8. Finally, FORTRAN software is developed to carry out the computations, with the full version of software used, in Appendix E.

Before the results from the proposed computations can be acknowledged, verification of the proposed method is required. This means applying the approach to a problem to which the solution is either well known or arrived at using independent means. The robot taken for verification is the MPIR developed in Technion, Israel, for series of tasks in interior finishing of a prefabricated shell of a building. MPIR was developed by task and configuration analysis and optimisation. The original approach to MPIR development and verification of the outcome using the author's particular use of GA is dealt with in Chapter 9.

Since the author's primary interest is design of an automated system for inspection and restoration of steel bridges, this is the problem studied. The complete design of the robot involves optimising kinematic and dynamic parameters; however, the author's aim is the optimisation of kinematic parameters only and the computations are applied to the particular application. The results and the variety of experiments are presented in Chapter 10 and the outcome for the new GA approach is analysed. Four sets of procedures are introduced to this model. In the first set, early in the computations, micro-populations are applied in order to reduce the scale of the computations and to observe the development of the solutions and this is combined with the second group and uses the parameter sharing (niching) to improve the GA performance. In the third set, the control parameters are used in various combinations. In the fourth set, another selection method is investigated, in order to further enhance the solution and explore their influence on the quality of the solution.

GRASP modelling is also developed in Chapter 10, so at the end of Chapter 10 the robot's parameters optimised with GA are input into the robotic computer simulation software (GRASP) and the additional conformation of the viability of the results is carried out.

The conclusions are given in Chapter 11. Finally, to assess the potential of the new use of the GA for robot's parameter optimisation, the follow-up questions are also addressed in Chapter 11:

i. To what degree are the criteria of the optimisation fulfilled?

ii. Do the generated designs accomplish the specified task?

iii. Are the generated designs related to the predicted ones?

iv. How practical is the method (time and complexity of setting up and running)?

Further, Chapter 11 answers the above questions, summarises the research, and suggests the future work.

PART ONE

LITERATURE REVIEW AND SUPPORTING STUDIES

CHAPTER 2 - LITERATURE REVIEW

2.1 Introduction

The work presented in this thesis is motivated by the desire to design robots systematically, starting by reviewing the literature on the design of robots. In the process of formulating a procedure for designing robots, it became apparent that a methodology for designing certain classes of systems would not only provide a means for designing robots, but also a means for designing a wide variety of devices. Therefore, it is also necessary to study the general literature on design methodologies. It is also essential to review the literature in the areas of genetic algorithms (GA), to learn of recent applications of these methods to parametric design and optimisation. Section 2.5 looks in the GA application into various aspects of robots' and engineering systems' design. Since, within this dissertation, the author uses the revised form of FORTRAN software and then verifies the results using computer simulation software - GRASP, a brief survey of other uses is included. The whole methodology and approach are developed using the example of steel bridge restoration. This covers inspection followed by surface preparation (paint stripping) and re-coating. A survey of these methods and an insight into deterioration itself commences the examination. Sections 2.2 to 2.5 review the literature in the above areas and Section 2.6 summarises this.

2.2 Design Methodology

The work in the field of mechatronics design methodologies is so vast, that there exists a paper that provides a review of current research in the field - a "tour guide" for the uninitiated [Finger and Dixon, 1989a and 1989b]. Rather than duplicating the review paper, this brief overview highlights several works that place the current work in context and examines other works that present similar approaches.

Attempts to model and formalise the design process have been made since the early sixties. In engineering design these attempts have converged into a phase model comprising four stages. Especially French [1985], Pahl and Beitz [1996] and Hubka [1989] describe this model in several textbooks in slightly different versions. Comparable models are those of Ullman [1997] and Pugh [1991].

Within the context of the four-phase models of design, it is important to determine where the current work fits into this. It is suggested that the current work can be applied to preliminary, conceptual and embodiment design phases.

Although, Pahl and Beitz's work provides a solid framework in which one can carry out engineering design, engineering design has yet to be defined. The definition offered by Dym [1992] is used in this research: “Engineering design is the systematic, intelligent generation and evaluation of specifications for devices whose form and function achieve stated objectives and satisfy specified constraints”. The author feels that this definition comes closest to describing the process of engineering design as implemented by the genetic design methodology. One of the key, underlying assumptions incorporated in this definition is that hierarchies of representation for both form and function exist, and that these representations can be manipulated. In addition, it is necessary that these representations can be translated into other representations.

There exist numerous techniques for representing systems and devices. One of the most general is the function logic method of value analysis [Sturges et al., 1992]. With this, a hierarchy of noun-verb pairs represents technique, objects, and classes of object. Although this technique is capable of expressing a very wide range of concepts, computer implementation is difficult and it lacks some of the formality of other methods. A related methodology by Cagan and Agogino [1987] presents the concept of designing from the basic, underlying principles. Although, this method is shown to be successful for certain, straightforward problems, it does not appear that this can be generalised for more complex systems.

Another approach used is bond-graphs [Finger and Rinderle, 1989]. In this paper, the authors state, "During the design process, a designer transforms an abstract functional description for a device into a physical description that satisfies the functional requirements." (This definition is quite similar to the one given above). To achieve this goal, the authors propose using bond-graphs, a tool for describing generalised lumped-parameter dynamic systems, to achieve the required transformation. The bond-graph technique works well in the domain, for which it is intended, however, it is not clear that this representational domain can be extended. Also, the authors did not propose a methodology for automating the process, a necessary step for producing a range of alternatives from which the designer can choose.

Other means of representation is the use of formal grammars. Stiny [1980], who developed the concept of shape grammars, performed some of the earliest work in this area. Shape grammars, which are used to describe planar shapes, have been shown to be equivalent to other types of formal grammars [Gips and Stiny, 1980]. More recently, Mullins and Rinderle [1990], Rinderle [1990] and Schmidt and Cagan [1994] have used grammars for the design of mechanical systems. In the concluding remarks of the Mullins and Rinderle's paper, the authors state that there are several critical issues in applying grammatical formalism to mechanical designs. One of those issues is the generation of good designs.

Schmidt and Cagan [1994] propose an abstraction model for conceptual design. A conceptual design progresses along several levels of abstraction, from a high level, black-box description "convert electrical energy into mechanical energy" to a low level, component technology description "use an electric motor". The authors develop a grammar that can distinguish between different levels of abstraction while being compatible across the different levels. The strings formed by these grammars are manipulated by a recursive annealing process to produce results that optimise a designer-specified objective function. However, there appear to be difficulties with this approach that may limit its applicability to more complex problems than the sample problem illustrated in the paper.

The author believes that the GA method of search presented in this thesis is superior to simulated annealing for this type of problem for several reasons: GA has been shown to work on a variety of problems with similar representations and the GA approach is less sensitive to the proper selection of control parameters.

The recent works of two sets of authors deserve special attention because they also look at GA to provide methodology for solution into the design problems. Both approaches vary from each other and from the approach assumed by the author in this thesis, but served to broaden the author's spectrum on using GA in methodology development.

In the first, [Maher and Kundu, 1993], the authors present a methodology for performing adaptive design using graph representations and GA representation manipulations. The illustrative problem used, is the topology of floor plans. The use of GA for design relies entirely on the encoding on the design problem, where one encoding results in the solution to an optimisation problem and another encoding provides an alternative search space representation to a problem whose objectives may change as the search progresses. Although, this work does achieve its stated goal of finding topologies that satisfy certain constraints, there is little indication of how this work can be extended into other domains. In addition, there is no indication of how to incorporate numeric values into this methodology.

In the second work [Lohmann, 1993] and [Lohmann, 1994] the author proposes a methodology called "structure evolution" for determining the parameters and structure of an object. This work is based on an adaptation of evolution strategy, a genetic methodology that is related to GA. The author appears to focus much of his attention on issues central to the evolution itself, such as using semi-isolated populations, as opposed to defining a structure into which other problems can be fit. In these papers, the author presents solutions to four problems in four diverse domains. Although, good results are obtained for each of the example problem domains, there appear to be some limitations. Firstly, the author does not explain how the problems are represented, although this may be for pragmatic reasons, i.e., page limitations. Secondly, the degree of structure variation permitted in the various problems appears to be limited. Finally, the author does not present a means for adapting this methodology to other domains. Although, this work does present good results, it seems that this work is inherently limited to problems that require a particular means of representation.

Finally, another author whose work has a direct bearing on this dissertation is Krottmaier [1993]. He compiles the methodologies for use of experiment methods. After the primary design is completed, the optimum values for the system's selected parameters have to be determined. The aim is, during this phase of development, to arrive at a suitable parameter combination to develop a product, which are operable under, specified, varying conditions. Krottmaier presents four experiment design methods: (i) classical, (ii) Taguchi, (iii) Shainin and (iv) GA. He claims that experiment design methods are the optimal tool for product development as well as product planning, since they are based neither on deterministic nor on stochastic rules.

In addition to these works, a number of researchers are also trying to understand the design process itself - that is, how people do design, see [Ullman et al., 1988] for example.

At this point the author feels that the contents of this Section should be augmented by reference to the QFD technique used to generate technical specifications. It is a vital part of the ED and some authors isolate it as an additional phase [Ullman, 1997]. The information structure and quality achieved using QFD, perfectly suited the purpose of this research. Particularly, the fundamental issues: (i) who are the customers, (ii) what are their requirements, (iii) evaluating the competition, (iv) generating the engineering specifications and finally, (v) setting targets, followed within QFD technique made it perfectly fit-to-use for a constructive research framework and an initiation step.

From the perspective of the current study, the key finding from this body of research is that engineering designers are quite adept at optimising a given design, but rather poor at generating alternative design options. This finding directly attempts to incorporate this shortcoming into the approach of the current research.

2.3 Design and Optimisation Methods for Engineering Systems and Robots

Developing a new robot or automated system is usually carried out according to the stages of the engineering design process - see the previous Section. Haas [1996] presents a study in development of an automated crack sealer, briefly relating to relevant stages of the process. The kinematic design and the parameter optimisation are, however, not singled out and justified, but rather buried in the description of the dynamics and economics.

Optimisation of any engineering system is just one stage within the whole of the engineering design process. Optimisation of selected solutions requires combination of modelling, experimentation, simulation, analysis and further acquisition of information concerning the operating environment. For different tasks, certain characteristics are more important than others are, and the type of robot selected should be chosen accordingly. The initial step for any robot optimisation, is to analyse the nature of the problem under design, and as a result, to identify and qualify the set of parameters and generate a geometrical description, known as configuration [Krottmaier, 1993]. The second step focuses on establishing all the parameters significant to the required result and then selecting and evaluating the most important ones. In order to set the significant parameters at an optimum value, several techniques can be applied [Krottmaier, 1993].

One issue that must be addressed in many design problem is that of scaling - both the absolute size of a device and its relative size with respect to their environment. For the robot design problem, the issue of scaling arises both during the “design” phase, specifying the sizes of the robot’s elements to allow it to move effectively, and during the "evaluation" phase, during which the robot must negotiate a feature-rich environment. For example, from the robot's perspective, there are two ways to negotiate a feature in the environment: to simply ignore it (the robot is much larger than the feature) or to move around the feature (the robot is of similar size or smaller than the feature). Several sources report the effects of scaling of biological systems, but only one is found which reports on the effects of scaling in robots, [Kaneko et al., 1987]. Although, this work makes a number of simplifying assumptions, which may limit its practical application for designing robots, it provides a solid analytical basis. In all cases, dimensional analysis is used to determine environmental boundaries. Kline [1986] presents a general discussion of dimensional analysis, but does not apply it to particular systems.

The majority of papers presenting studies on robot development, tend to fall into one of two categories: (i) choosing an appropriate type of robot from a number of candidates or (ii) matching the kinematic design to the task in the form of predicted movement sequence of the tool. Neither provides an analytic explanation for their choice of robot configuration. Navon [1989], who falls into the first group, determines the optimal configuration using computer simulation of the task and the robot (using ROBCAD) and evaluates the suitability of the six configuration alternatives. He does not claim that one configuration is innately superior to the others, but rather claims that for his particular application, one of the configurations may be superior to the others. By presenting a table showing the relative strengths and weaknesses of the different configurations, another engineer might chose one of the other configurations for his particular application. Navon's work clearly illuminates the difficulty of choosing between different configurations when multiple objectives exist. The second group is represented widely throughout the relevant literature. Examples include the masonry robot from the University of Stuttgart, Germany, [Pritschow et al., 1996], concept of a robot for interior building trades from Karlsruhe, Germany [Spath and Andres, 1997] and the report on WASCOR IV by Handa et al. [1996]. The imaginary sequence of tasks is presented then the assumed robot movement and finally a suitable configuration is fitted in. This method is definitely fast, relatively inexpensive, but it does not guarantee the best solution. Among other key issues, the intent of these projects seems to be modelling the activity by changing feedback gain parameters, rather than determining performance measures.

The whole group of publications deals with enhancing the automation of the existing systems, so ‘improvement design’ is taking place. The vast majority of robotics attempts try to incorporate the means of movement, usually by wheels or legs for locomotion. Two papers compare different types of mechanisms for use as robotic legs, [Kilne, 1986] and [Ryan and Hunt, 1985]. Although, purportedly for robotic application, these papers are discussions of the kinematics of mechanisms, in chapter 3 of his book, Todd [1985] gives a list of leg properties and important attributes for leg design. A comparative study is not presented; however, these lists present a number of different leg configurations, configurations that the GA methodology should be capable of generating given an appropriate representation. However, after this initial research it became apparent, that the development in the locomotive system in robots is limited, progresses disproportionally slowly, compared to the task requirements. Therefore, it is established at the early stage that the robot is delivered to the task and further research into locomotion is abandoned.

2.4 GA in Optimisation

The primary source for GA theory is Holland's 1975 book. This sets forth schemata theory and sets the stage for future developments. This is highly mathematical in nature and does not provide any "real-world" examples. Another important source is Goldberg's 1989 book. This book does not dig as deeply into the mathematics as Holland's book, but rather presents more of a user's guide to GA, providing some examples as well as a listing of the then current applications using GA.

Because of the means of representation, GA is not well suited to open-ended problems. A good example of this limitation is presented in Angeline et al. [1994], in which the author uses a GA type approach, but does not use the traditional bit-string chromosomal representation (refer to Appendix B for the explanations of the initial GA concepts and terminology). Since GA work best with fixed length encoding, the GA design research has been limited to problems whose topology remains fixed, but whose parametric values can he changed. Some example application areas include controller parameter estimation filter design, scheduling, truss and strut optimisation, etc. The proceedings of the annual conference on genetic algorithms typically has papers about endless new GA applications [Rawlins, 1991], [Whitley, 1993], [Schaffer, 1993], [Belew and Booker, 1991] and [Forest, 1993].

A classical publication on the GA is Davis's [1991] "Handbook on GA". In the first part, he considers general features of GA, which is written in the form of tutorials, while the second part is mainly editorial work, implementing those ideas. Chapters combine publications on GA application to optimisation in various fields, such as aircraft design, classical travelling salesman and scheduling, and many more, while also comparing and revalidating GA's performance with non-linear dynamics, expert systems and numerical optimisation.

To clarify the exact ways GA work in relation to neural networks and chaos theory, Bauer's [1994] "GA and Investment Strategies" is important. Briefly, but quite proficiently, it introduces many techniques, then concentrates on GA and then, by applying it to the investment strategies, demonstrates the value of GA as tools in the search for effective trading ideas.

Goldberg introduced the notion of "messy GA" [Goldberg et al., 1989]. With this technique, the chromosome size of the members of the population is not constrained to be of the same length. The additional freedom allows the expression of a wider range of objects and permits the creation of new objects.

GA is also used in classifier systems, a type of machine learning algorithm that is formulated from a number of productions, statements typically of the form: IF ⇨ condition ⇨ THEN ⇨ action. By establishing a set of these productions and testing them on known cases, the algorithm increases the weight assigned to a production if it is activated during the learning phase. Once the weights have been assigned, new cases can be input to the classifier system and results obtained. Oliver [1993] presents examples of how this technique can be applied to problems with multiple objectives. However, these examples assume that an evaluation of the candidate devices exists *a priori,* and the program discovers the rules used to generate the ranking. Although, this method does provide a straight forward means for handling multiple objectives, for the general problem of artefact design, *a priori* rankings are not available, so it is not readily apparent how the classifier system approach could be used to solve this problem.

Finally, the most recent publication by Mitchell [1996], under a promising title 'Introduction to Genetic Algorithms', begs a few comments. The main inconsistency of this is the fact that certain aspects of GA mechanisms are just briefly explained (like mutation and crossover) and others investigated to the inconsistent depth (selection, for example). Also, the GA applications mentioned are reduced only to areas of scientific models and general problem solving.

2.5 GA in Engineering Design of Systems and Robots

Many of the current publications related to GA are focused on implementation. This is not unexpected due to the relative newness of these techniques. The work presented in this thesis, while novel in implementation, utilises fundamental techniques and so-called simple GA (SGA), as described by Goldberg [1989], as well as version of hybrid GA (HGA). SGA is domain-independent, general-purpose algorithm providing a perfect balance between the best solution exploitation and the search space exploration and uses non-overlapping populations and optional elitism. It generally performs a blind search using standard operators. There are many situations where SGA does not perform particularly well and various methods of hybridisation were proposed. The variety of hybrid techniques proposed is numerous and it is impossible to predict *a priori* which one is the most suited to the particular case. Only studying other work and looking for parallels provides the reasonable guidance. The methods of hybridisation attempted by the author built from SGA and added local improvement operators, niching and micro-populations. Future work might include implementing some of the theoretical work being done in the GA fields to improve the performance of the method developed here.

Three articles use GA to solve mechanical design problems. The first [Brown and Hwang, 1993] starts with a user specified configuration and applies GA to select appropriate parts from a catalogue. The second [Pham and Yang, 1993] designs a transmission based on user requirements from a set of primitive elements. The third [Bullock et al., 1995] reviews the application of GA to several examples of engineering design carried out in Plymouth Engineering Design Centre. The problems under consideration included design of arch dams, buildings, digital filters, gas turbines and the thermal cycle of nuclear power stations. All show that variety of genetic approaches to design are viable, though neither could be applied to so many varieties of aspects of design, as the work presented in this thesis. Even the recently published book by Gen and Cheng [1997] shares the above-explained shortcomings.

Robotic systems are known to benefit greatly if the individual sub-systems, such as mechanical structure, control circuitry, trajectory specifications, etc., are optimised [Davidor, 1991]. Several aspects of robotic design and programming have been previously addressed using GA. Problems such as mobile manipulator path planning [Zhao, et al., 1993], robot motion planning [Ahuactzin, et al., 1991], collision avoidance [Baba, and Kubota, 1994] and design of redundant manipulators [Davidor and Goldberg, 1990] found their approach justification in using genetic based techniques. The five publications above represent the whole group of works, which use GA for various aspects of control and trajectory optimisation. These areas have been well researched and numerously applied, however, firstly they deal with already pre-determined kinematic configuration and secondly, they are only involved in the separated, small part of overall design process.

Chen and Burdick [1995] show an interesting use for GA in kinematic design. Given pre-determined set of modules, this considers the problem of finding the optimal module assembly configuration for a specific task. This is only worth mentioning as it addresses specifically initial kinematic design, but its limitations lie in restricted choice of components, although the evaluation is based on the task specification. This problem resembles a travelling salesman dilemma (one of the fundamental and most studied combinatorial optimisation problems in GA), more than structured engineering design. Along similar lines develops the approach by Shibata, et al. [1992] in obstacle avoidance by a set of mobile robots. GA is used to optimise the strategy consisting of co-ordinate planning of every robot combined with separate planning of each robot. Path planning and obstacle avoidance is also a subject of Lin's et al. [1994] publication, where the evolutionary algorithm searches the entire, continuous free space. The major limitation here is the fact that the best path is obtained as an improvement of the previously 'best' path and is unable to replace the current global path, by another possibly better global path entirely.

One final item to be reviewed is the Kim and Koshla [1993] paper. In this publication they propose a design methodology called 'Task Based Design' (TBD), a methodology for designing optimal robot manipulators. Although, this methodology is for manipulators, it is intended to solve the design of complex systems. The TBD methodology is used to determine the optimal parameters, base position and poses for a manipulator to carry out a specified task. Since the authors restricted the scope of this work to include only those manipulator arms comprised solely of revolute joints, they exhaustively search the space of all possible manipulator configurations to find the optimal one. For low degree-of-freedom arms (the type most likely to be designed), the space is not very large, and can be explored efficiently. Since the space being searched is well defined, the authors use a GA approach for finding optimal manipulators. Although, the TBD methodology has been used successfully, it does not appear that it can be readily extended to other domains because its success is, in part dependent on the "limited" domain it works in. However, the success of the technique in automatically generating complex systems to optimally achieve a specified task lends support to the notion that the design of complex systems can be automated, at least to some degree. Since TBD is designed specifically for working within the manipulator domain it yields quite accurate results, because of the specific knowledge encoded in TBD.

2.6 Alternative Optimisation Tools and Methods

Several tools and techniques have been developed for optimal design, or selection of a robot for a given task, such as computer simulation [Navon, 1989 and Navon, 1995] or mathematical analysis [Ashiru et al., 1996]. The latter is mainly based on inverse kinematics and inverse Jacobians calculations [Schilling, 1990]. The computer simulation is a good practical tool, but it does not investigate all the possible values of the robot parameters, thus theoretically risking missing an optimal solution.

The mathematical techniques are not versatile enough and they require tedious calculations and they require continuous functions and their derivatives. This does not imply that genetic techniques are the only optimisation routines that do not require continuous functions and their derivatives, as other techniques also do not.

From the heuristic search methods, which focus on regions, which seem to be improving the search, the characteristics of two alternatives were considered, simulated annealing and hill climbing. Both methods require the evaluation function that scores a node in the search tree according to how close to the goal state it seems to be.

In hill climbing [Schwefel, 1995], the basic idea is to head towards a state, which is better than the current one. The algorithm does not attempt to exhaustively try every node and path, and no node or agenda are maintained, just the current state. If there are loops in the search space, then hill climbing cannot deal with them, as going up and back is not possible, additionally, hill climbing terminates, where there are no successors to the current state. This very often results in terminating the search at the local optima in the search space (points which are better than any surrounding state, but not the total solution). Hill climbing is only suitable for limited class of problems, where the evaluation function fairly accurately predicts the actual distance to a solution.

Simulated annealing (SA) is a numerical optimisation technique, where the optimised variables of a system are perturbed between subsequent configurations and the result measured with objective function, with scoring [Aarts et al., 1997]. The primary advantage of SA is the ability to move from local optima and simplicity of implementation. The main disadvantage, however, is the subjective nature of choosing the SA optimisation configuration parameters and that it typically requires more response or objective function evaluations than other optimisation approaches. Also because in SA steps are taken randomly, SA is classified as having weak heuristic search. The fact that SA employs no knowledge of the response history [Vidal (ed.), 1993] was considered by the author as a serious weakness in selection of optimisation tools.

One important fact to note is that GA use random numbers, as do simulated annealing techniques. However, unlike simulated annealing techniques, GA are not directionless, because they make use of past events to guide future events. These techniques yield continually improving performance of the functions being sought, therefore the author has concentrated on the GA application and development.

2.7 Summary of the Literature Survey

The review of the literature has shown several things: firstly, although robots have been designed for many years, there is as yet no straightforward means for designing these systems. Most of the robot designs reported do not discuss the crucial design decisions that led to a particular configuration. Secondly, although there are numerous design methodologies, none seem well suited to the task of designing complex systems. Thirdly, evolutionary computational techniques have the capability of being able to intelligently search large, discontinuous spaces. By expanding these techniques, they become well suited for object design.

Thus, this research attempts to build upon the previous works to develop a system that overcomes some of the previously mentioned inadequacies. The following Chapters 3 to 6 also concentrate on data assembly and critical review, but in relationship to theoretical design framework (ED), existing steel NDT and restoration methods, benchmark library of tasks on bridges and commercially available enabling technologies.

CHAPTER 3 - ENGINEERING DESIGN FRAMEWORK

3.1 Introduction

Before design of any system can commence, the primary factors and a driving forces have to be identified, namely a commercial or evolutionary need for the device.

The contents of this and the following chapter establishes and reinforces the necessity for the automated or semi-automated device for restoration of steel bridges, using the framework of engineering design (ED). The problem of efficient design of the automated device for inspection and surface preparation still is to be solved. According to Ullman [1997], the need can have three sources: the market, the development of new technology, or a need from the higher-level system. To-date there are no commercially available systems, which efficiently, and with minimum human involvement, could inspect, assess and restore a variety of steel bridges. Design cannot commence without exact knowledge of what the market expects from the product. The following chapters look into bridges under deterioration (with typical locations of corrosion), existing inspection and restoration technology, and automation attempts, in order to precisely understand the design problem. The range, relative importance, application, skill requirement and reliability in inspection probe work and restoration process are also investigated here. The methods of handling data collection, safety and economic implications are likewise covered.

All the above is going to be investigated in the formalised manner, within the environment of ED process. During the design process, the function of the system and its decomposition is considered first. After the function has been decomposed into the finest sub-systems possible, (for example, access to task of the probe, access to task of the device, probe or tool holding device) assemblies and components are developed to provide these functions. There are different types of design problems, depending on the path chosen to arrive at the solution and they juxtapose as the process develops. Thus, certain aspects or elements may be selected from a group of similar items (selection design). At specific stage design these may involve assembling or configuring all the components into the completed product (configuration design), or a particular item may have to be designed by finding values for the parameters, which characterise that object (parametric design). These different design procedures lead eventually to an original design.

The scope of this chapter comprises the brief description of the engineering design process, introduction to the techniques to be used on the project and description of their particular application.

The author uses the ED in order to clarify the path to a 'good' design. The originality within this chapter lies in the author's ability to use and analyse a complex format of the ED to select a relevant research path, which provides the overall sequence of activities leading to a successful product.

3.2 Phases of Engineering Design

The brief compilation below recalls the basic phases and techniques within the ED process. Each design problem is different and some of the techniques are not applicable to some problems. The following section will, therefore, concentrate on the specific procedures used in this research. The design process has the following stages, which although under different headings in variety of literature, refer to similar activities in Figure 3.1.

Clarification of the task (problem definition, analysis of the problem, specification development and planning phase)

All information about the problem is collected. This phase includes the following activities: (i) understanding, defining and analysing the problem (task), and as a result (ii) generating engineering requirements, (iii) coming up with design (performance) specification, defining functions, properties and constraints (standards, deadlines, budget) of the product, setting limits to the solution space, providing the evaluation criteria. The specification includes: (i) listing and evaluation of objectives (criteria) and (ii) description of the variables of the objectives (parameters)

Methods:
QFD (Quality Function Deployment), Taguchi, Experimental Design Techniques, Design for X, Design for Manufacturing

Conceptual Design

The following actions comprise this stage: (i) clarifying the task and elaborating the specification, (ii) identifying essential problems, (iii) establishing function structures, (iv) searching for solution principles - literature search, brainstorming, (v) combining solution principles and forming them into concept variants (defining physical and technical characteristics) and evaluating them against technical and economic criteria (data), (vi) generating selected schemes or concepts (costs, weights, dimensions, feasibility assured), (vii) firming up and evaluating concept variants and coming up with the concepts (schemes).

Methods:
Creativity Methods: (i) Association Methods: Brainstorming, Cause-effect Diagrams, Fault Tree Analysis (FTA), (ii) Creative Confrontation Methods: Synthetics, (iii) Analytic Systematic Methods: Function Analysis, Morphological Method, AIDA, (iv) Genetic-based: GA and GP representations, (v) Ishigawa and (vi) FMEA

Embodiment of schemes (embodiment design, evaluating

This phase consists of two sub-parts and it is a continuous refinement of a concept through corrective cycles in which analysis, synthesis, simulation and evaluation alternate and complement each other. The following activities are typical: (i) generating a concept, (ii) developing preliminary layouts and forming designs, (iii) selecting best preliminary layouts, (iv) refining and evaluating against technical and economic criteria, (v) making a final choice between the schemes.

Methods:
(i) Ordinal Methods - The Majority Rule, The Datum Method, (ii) Cardinal Methods - Weighted Objectives Method, Measuring the Effectiveness, Homing-in according to Shainin

Then the subsequent action take place: (vi) generating preliminary design, (vii) optimising and completing for designs, checking for errors and cost effectiveness, preparing the preliminary parts lists and production documents, (viii) coming up with definitive design (decision making) as a result of the optimisation

Methods
(known as Experiment Design): (i) variables search, (ii) full factorial experiments, (iii) Taguchi experiment methods, (iv) graphical evaluation, (v) variance analysis, (vi) GA

Detailing or detailed design (product design phase)

The stages cover: (i) generating definitive design (arrangement of assemblies, components and parts, geometrical shapes, dimensions, materials), (ii) finalising the details, (iii) completing detail drawings and production documents, checking all documents, (iv) assembling final documentation

FIG.3.1 Engineering Design - Overview

3.3 Clarification of Tasks and Methods within Each Phase

After the general introduction of the ED process and stages, the author clarifies the approach she is taking for development of the project [Gen and Cheng, 1997].

3.3.1 Specification Development and Planning

Engineers have become increasingly aware of the importance of ensuring the durability of structures. Any structure requires maintenance and only sufficient inspection tools and procedures can guarantee that the need for restoration is established and delivered in time [Pritchard, 1992].

Steel is a versatile construction material that has been widely used in many forms and structures, however its major disadvantage is a high susceptibility to corrosion. Hence, the need to provide and maintain means of corrosion protection and repair. Steel bridges with their varied geometry, lack of direct access and large scale are among structures requiring special provisions. Careful thought at the design stage directly affects maintainability of a bridge. Therefore, control over the deterioration of the bridge is already decided at this stage and subsequently is dealt with through regular cleaning and re-application of primers and topcoats for long lasting corrosion protection, with the minimum cost in view.

The process of coating removal poses a substantial health hazard and can create high levels of pollution to the environment, which makes it a prime candidate for automation. Removing existing coatings (mainly lead-based primers) is normally done by blast cleaning [Thompson, 1995]. The area being cleaned is eroded away by the mass of the particles, until a firm surface of the required profile is achieved. The residues of the process are particles polluted with lead toxins, which pose a serious health hazard to the workers and inevitably contaminate the environment [Bennett, 1993]. Therefore, an automated device, which is capable of performing this job, is highly desirable, as it would improve the health and safety aspects of bridge maintenance.

3.3.1.1 Understanding the Design Problem

In order to define the problem as soon as possible, the following conditions have to be met (i) identifying the customers and determining their requirements, (ii) determining relative importance of the requirements, (iii) competition benchmarking - to force an awareness to what already exists and point out opportunities to improve on what already exists, (iv) translating customer requirements into engineering requirements, (v) setting engineering targets for the design.

Generating a project plan involves: (i) identifying the tasks, (ii) stating the objective for each task, (iii) estimating resources and time, (iv) developing sequence for the tasks and (v) estimating development costs.

Therefore, the customers are first identified, which included Local Authorities, as owners of most bridges and inspection and restoration companies, which tender for contracts to maintain this type of structures. The main customers' and designers' requirements are recognised and listed according to priorities and are given below in Table 3.1, which is a rough pre-version of the house of quality for the device.

The customers' requirements usually are contained in performance specification and are not formulated in designer's language. The designers' requirements some of which are listed in Table 3.1, can be grouped under the following headings: (i) geometry, (ii) kinematics (direction of motion, velocity, acceleration), (iii) forces (direction, size, frequency, load), (iv) energy (power, efficiency, connection energy), (v) material, (vi) signal (input and output variables, display, operating and monitoring equipment), (vii) safety (protection systems, workplace and environmental safety), (viii) ergonomics (MMI, operating method, lighting), (ix) manufacture (process, tolerances), (x) monitoring (measuring and testing, special regulations), (xi) assembly (installation, foundation), (xii) transportation, (xiii) application (location), (xiv) maintenance, (xv) recycling, (xvi) costs and (xvii) deadlines.

Table 3.1. Matrix of Customers' and Designers' Requirements Relationship

Customers' Requirements L - low priority M - medium priority H - high priority	**Designer's Requirements**	Operator interface - communications	Operator interface - control panel	Sensors - location assessment	Tool interface	Simplicity of assembly	Choice of materials	Surface attachment mechanism	Power supply	Workspace delivery mechanism	Environmental safety	According to common manufacturing	Type of control system	Number of operators	Safety of device	Precision of delivery	Simplicity of operating	Operator safety	Customer cost
Designed according to standards for safety													LOW PRIORITY						
Designed according to engineering standards (British and European)																			
Feasible to manufacture using existing facilities																			
Use of existing components (when possible)																			
Easy to maintain																			
Reliable																LOW PRIORITY			
System safe to use																			
Containment present, easy to attach/detachand lightweight																			
Power provision clearly identified (with/without generator on board)				MEDIUM PRIORITY															
Robustness																			
System easy to attach to and detach from a bridge									MEDIUM PRIORITY										
Remote operation (no need for many highly qualified operators)																			
Versatile on variety of bridges																			
Of manageable size	HIGH PRIORITY																		
Lightweight																			
System easy to operate																			
Speed and simplicity of assembly																			
Components easy to obtain			HIGH PRIORITY																
Low overall cost of the system																			
Low cost of components																			

As this project is a research project, the author recognises the benefits in using the framework of the engineering design process, but certain practical aspects are not applicable, such as time is unspecified. Initial data for the client's minimum capital cost and the manufacturing cost is also not available. The cost estimate may become one of the optimisation criteria and is feasible to assess according to Warszawski [1990]. The clients involvement in the form of predicted amount of items of the final product manufactured per year and number of years are also not stated.

Competition benchmarking encompasses extensive market research into commercially and academically available systems for similar purpose and identification of their inadequacies and superiorities, against list of requirements. At this point a critical review of all existing automated and semi-automated systems commercially and academically available is carried out (see Chapter 5).

Translating customer requirements into measurable engineering requirements aims at producing design specification. It is important to find as many ways as possible to measure each customer demand. If a specific customer requirement cannot be translated into measurable engineering one, then this indicates poor understanding of the customer need. All the competition products need to be measured (not only compared to) against customer requirements.

The last step within this stage is to determine target values for each engineering measure and to set them to a specific value. The technical specification is then completed and the 'solution space' determined. This means the complete set of parameters, which correspond to the basic design requirements.

The procedure of understanding the design problem, described above, follows the steps of the Quality Function Deployment Technique (QFD), as one of the most verified methods for the purpose.

3.3.1.2 Project Planning

A project plan is a document that defines the tasks that need to be completed during the design process. For each task the plan states the objectives, personnel requirements, time requirements, schedule relative to other tasks and a cost estimate. As discussed previously, due to academic nature of the project, personnel and time requirements, as well as detailed cost estimate are not considered. Task titles (in sequence), which at the same time represent decomposing the design problem in into the sub-problems are shown in Figure 3.2 below.

Task	Objectives
Task 1: Investigate process of steel corrosion	Objectives: Understand the process, identify factors influencing speed and type
Task2: Investigate steel bridges	Objectives: Identify key areas particularly subject to corrosion, link various types of corrosion to particular type of bridge
Task3: Study NDT tools and techniques	Objectives: Through analysis of each method, identify their automation applicability, classify and detail it
Task4: Study steel restoration methods	Objectives: Through analysis of each method, identify their automation applicability, classify and detail it
Task5: Review the existing technology in automation and robotics in steel inspection and restoration	Objectives: Carry out competition benchmarks
Task6: Investigate commercially available access systems for bridges	Objectives: Identify the best system according to engineering requirements
Task7: Investigate tool handling configurations	Objectives: Select the most suitable options for the further optimisation
Task8: Complete specification development	Objectives Demonstrate problem understanding by generating engineering requirements (through the criteria and relevant parameters)
Task9: Establish two concepts for product development	Objectives:Set up competitive schemes for optimisation
Task10: Investigate GA as selection and optimisation technique	Objectives: Familiarise oneself with the full potential of the tool
Task11: Clarify the final set of parameters and criteria	Objectives: Develop a framework for the GA software extension
Task12: Development of FORTRAN software to execute the optimisation	Objectives: Select the better out of two concepts
Task13: Enhance the GA technique	Objectives: Make it more universal to be used for extended set of parameters and criteria as well as different stages of engineering design process (conceptual as well as embodiment with directions for further studies into using it for detailed design)
Task14: Application of the method to a robot previously optimised using different methods	Objectives: To verifiy the validity of the new GA application
Task15: Confirm the concepts separately using GRASP	Objectives: Further confirmation of GA outcome

Figure 3.2 Planning for Design

Tasks identified within ED process techniques reflect the objectives from Section 1.3. This further confirms that the earlier choice of research objectives is substantiated by the formalised approach, which serves as reassurance that the path to the successful project outcome is selected correctly.

3.3.2 Conceptual Design - Concept Generation

Some conceptual ideas are naturally generated during the specification development phase, since in order to understand the problem we have to associate them with things we are already familiar with. It has been proven that selecting a single idea at that stage and refining it into a product design leads to poor solution. Additional techniques need to be used to generate a pool of potential solutions, which will concentrate on the function of the device being designed. Two techniques are the most common here: (i) functional decomposition and (ii) generating concepts from functions. The first technique further refines the functional requirements and the second aids in transforming the functions to concepts.

Finding the overall function that needs to be accomplished becomes a first step. Here, the overall function statement is to design an automated or semi-automated system, which can test a variety of steel bridges for corrosion damage and then prepare the surface for repainting, using containment for environmental protection. Then, decomposing the function into sub-functions must follow. The second technique of developing concepts for each function follows this.

At this stage it is more important than ever to realise that most design situations are a mix of various types of problems. Although, the user works on development of a new product, it is recognised that advanced technology exists in other industries, and its transferral, adoption and integration needs further investigation. For situations of this type, constrained sub-sets of original design exist. These are: (i) selection from a limited list of options, (ii) configuration as an arrangement of components or (iii) parameter optimisation as choice of related, characterising values.

For example, expensive, purpose built access systems are commonly employed, rather than low-cost, re-configurable, high performance modular systems appropriate for flexible applications. Also, despite the similarity with tasks in other industries, advantage has not yet been taken of robotic manipulators, which can meet the requirements for tool delivery and handling with hazardous access. Therefore, rather than treating the task as the original design of the whole system, the author sees it as mainly a combination of a selection problem, in the form of several sub-problems. The list of choices serves in generating potential solutions, partially as configuration design in organising the assembly, taking into account geometry, functions and spatial relations and finally, as a parametric design with the criteria functions of the design variables. The Genetic Algorithm approach to optimising kinematic parameters of the proposed system, developed later in this thesis, is primary aiming at evaluating a design concept, which creates a bridge over to the next stage of ED.

2.3.3 Evaluation and Refinement of Concepts

The potential solutions from the selection stages have to be evaluated versus the specific requirements. In parametric design, the solutions are the values of the design parameters, which optimise the criteria functions. All these are effectively part of the evaluation process, which implies both comparison and decision-making.

Evaluation can be based on: (i) feasibility judgement, (ii) technology readiness assessment, (iii) go/no-go screening and (iv) decision matrix method. Feasibility judgement can have three outcomes: (i) total rejection based on lack of compatibility with customer requirements or unavailable technological solutions, (ii) conditional rejection depending on further actions and (iii) worth considering the verdict, further requiring engineering knowledge and experience. Graphical, physical or analytical models need to be developed and evaluated against certain criteria.

Technology readiness assessment applies to evaluation the comparison with state-of-the-art capabilities. Six measures [Ullman, 1997] can be applied to determine a technology's maturity: (i) can the technology be manufactured with known processes, (ii) are all the control parameters which control the function identified, (iii) are the safe operating latitude and sensitivity of the parameters known, (iv) have the failure modes been identified, (v) does hardware exist that demonstrates positive answers to the above four questions, and (vi) is the technology controllable throughout the product's life cycle ?

Evaluation based on go/no-go screening compares each concept with the customer requirement in an absolute fashion. This help in eliminating designs that should not be further considered and help generate new ideas.

Evaluation based on decision matrix or Pugh's method [1991] compares concepts that are not ready enough to be directly compared with engineering requirements. This method provides a means of scoring each concept relative to another in its ability to meet the customer requirements. Scores are generated in relation to the 'favourite concept' taken as a *datum*.

In the course of this research, concentrating on technology readiness assessment and feasibility judgement carries out the evaluation. All components (suspended platform, carriage, arm, wrist, restoration technologies, probes and tools for inspection and cleaning) are investigated and appraised according to the factors listed above. The feasibility judgement is accomplished partially through results from GA optimisation computations (within this thesis only for selected parameters) and partially through GRASP modelling and evaluation, purely for confirmation more than for development.

Early in the robot design refinement, it may be sufficient to find only the order of magnitude of some parameters. This means in practice evaluating some of the parameters (previously assessed as the most important) to simply obtain the outline design of the product. As the product is refined, the accuracy of evaluation modelling must be increased to enable comparison with the target values. It is vital to assess the accuracy needed prior to evaluation. According to this level, the choice and amount of parameters, as well as type of constraints (here criteria) has to be set.

3.3.4 Embodiment or Product Design

The goal here is to refine generated concepts into quality products. This refinement is an iterative process of generating product designs and evaluating them against requirements. Documentation of this stage consists of: (i) layout drawings, (ii) detailed drawings, and (iii) bill of materials. This part of the engineering design process is allocated for further research for utilisation of the GA. Table 3.2, below, lists the guidelines showing principal features of the embodiment design.

Table 3.2 Principal Features of the Embodiment Design

Principal Features	Examples
Function	Is the intended function fulfilled? Which secondary functions are necessary?
Effective Principle	Do the selected effective principles produce desired effect? Which malfunctions are to be expected?
Design	Does the design material guarantee desirable durability, stability, freedom from resonance, expansion, corrosion and wear characteristics?
Safety	Are factors that affect safety taken into consideration?
Ergonomics	Are the MM relationships taken into account? Was a good design practice observed?
Manufacture	Are manufacturing aspects taken into consideration with techno-economic factors?
Monitoring	Are the necessary checks feasible and arranged for?
Assembly	Can all assembly processes in- and ex-works be undertaken simply and unambiguously?
Transportation	Are transport conditions in- and ex-works checked and taken into consideration?
Use	Are events occurring during use, operation or handling taken into consideration?
Maintenance	Are maintenance, inspection and repair procedures feasible and capable of monitoring?
Recycling	Has re-use or recycling been facilitated?
Costs	Are pre-set costs limits to be observed? Do additional operating or secondary costs arise?
Deadlines	Can the deadlines be met? Can a different design improve the deadline situation?

3.4. Summary

The purpose of this chapter is to set up a framework for carrying out the research into the optimisation of the automated system for steel bridge restoration. Many variations of design problems were identified and investigated: selection, configuration, parametric, original, redesign, routine and mature. Allowances were made for the fact that very frequently a design process is a combination of the above. Additionally, the design process is viewed as a continuous constraining of the potential product design, until the final product evolves, which satisfies the initial design requirements. Following the engineering design process (due to its specified stages, characteristics and aims) enables to ensure that the solution is indeed the one most closely fulfilling the design requirements and none of the aspects leading to well designed product have not been missed. Therefore, the process itself is characterised, its stages and techniques assembled and the clarification of all actions assembled, which author is going to undertake within the scope of her research. Figure 3.2 was assembled, which distinguished all tasks and relevant objectives, ensuring the compliance with the ED structure. Not only the logic and sequence of the following chapters is justified, but also the use and application of genetic techniques indicated and allocated to relevant stages. The following three chapters are directly leading to concept development, by specifying the design requirements. The initial stages include: (i) justifying the need for the product, (ii) assessing the working environment of the proposed device, (iii) investigating and appraising the inspection, restoration and enabling technologies.

CHAPTER 4 - BENCHMARK LIBRARY OF TARGET AREAS ON STEEL BRIDGES

4.1 Introduction

The previous chapter set up the design framework for the steel bridge restoration robot, using ED and QFD. Assessing the task environment, which means steel bridges, their types and dimensions, grouping them, and identifying problem areas (particularly prone to deterioration), is the initial point of task clarification and is identified as task two in Figure 3.2.

The types of steel used in bridge structures mainly include steels that do not possess in-built corrosion resistance. Corrosion of carbon / magnesium steels can occur only if both oxygen and water are present and bridges, due to their function and scale, are subjected to continuous supply of both. As the corrosion theory is well understood, a concise compilation is included in Appendix A.

The objective of this chapter is, therefore, to analyse bridges from a structural and geometrical perspective, identify key common locations particularly subjected to corrosion, and link specific susceptibility to particular areas on the bridge.

The author is going to look briefly at the selection, classification and grouping of bridges into classes. Also corrosion-prone areas are going to be identified. Based on this, the selection of fifteen areas is identified. Every 'problem area' is trimmed to a size of approx. 3x3x3 m^3. This is for the purpose of identifying the 'unit size workspace', which classify workpiece, which can be reached from a single location of the robot. Collection of these 'unit workspaces' can create a benchmark library, as many similar detail areas are expected to be found on a variety of bridge geometries. Additionally, each workspace has been allocated four paths that a robot's end effector would be expected to follow in order to proceed with the restoration tasks. These paths are allocated in the extreme locations of the workspace geometry. The bridges' workspaces are drawn using robot simulation software GRASP, towards the prospective robot modelling and simulating the tool movement on the paths for feasibility check.

The paths located within each workspace are the result of the task analysis and this chapter paves the way for close analysis of the activities and relevant tool characteristics, which are involved in variety of restoration tasks. Every point on the path is characterised by a unit ortho-normal vector called a Targ. The orientation of the Targ specifies the orientation of the tool approach, identified from the tool specification. So the chapter is directly linked to the following one, in interrelation between the bridge general and specific characteristics and assessing them from the robot application perspective.

Isolation of the tasks and their characteristics in a robot-applied perspective and assembling the initial benchmark library is a contribution to the objectives of this research.

4.2 Steel Bridge Model

4.2.1 Types of Steel Bridges under Investigation

The classification of bridges covered in this paragraph serves as a framework for establishing the extent of similarities of details or connections across the range of bridge types.

Bridges can be broadly classified under three headings: (i) beam, including simple, continuous, truss and the cantilever types, (ii) arch and (iii) suspension.

Beam bridges have the main structure as a plate girder or, for longer spans, a continuous girder or steel box girder. The transverse stresses and stiffness are provided for by diaphragm cross beams or girders. Beam truss bridges use trusses as the structure for the simply supported as well as continuous spans. Beam cantilever bridges consist of two anchored cantilevers supporting a beam suspended from the ends of the cantilevers. For large spans, cantilever bridges usually comprise steel trusses (trussed girders). Cantilevers can also meet without a middle suspended span. All types in this group are riveted or welded, usually decked and sometimes composite with deck.

With arch bridges, the arch is the main structural member and transmits the loads imposed on it to the abutments. Since steel is capable of taking tension, the arch rings can be very much thinner than in masonry or reinforced concrete construction. The braced spandrel type bridge is usual constructed in steel, as are also bridges where the roadway is supported by hangers from the structural arch. Another type of arched bridge is the stiffened tied-arch, which is called a bowstring girder.

When spans are large (over 600m), suspension bridges are generally the most economical. Usually, there is a central span with two side spans and cables, which passing over the top of the supporting piers, are anchored in tunnels or by other means. The roadway is suspended from the inclined cables by vertical hangers.

Another development in suspension bridges is the cable braced (stay) bridge, where the girder is braced and stiffened by cables radiating from a mast or tower at one or both ends of the span.

4.2.2 Areas Mostly Affected and Standard Remedial Measures

With the diversity in structure types and designs, there is enormous variability in the possible composition of edges, corners, bolts, protrusions, back-to-back angles, joints and flat areas. Considering access difficulties, frequent bad detailing (e.g. causing trapping of water) and a variety of unfavourable environments (e.g. below ground, in contact with water, etc.), these areas are the most susceptible to deterioration and, therefore, of prime interest during inspection. Although, the geometry and spans vary enormously, there are some standard details and similar areas for grouping.

Typical areas covered by different types of deterioration include: (i) main structure steel surfaces, (ii) bearings and expansion joints, (iii) parapets, (iv) waterproof membranes and road surfacing, and (v) steel decks [Gregory, 1992]. These areas need special attention in inspection procedures. The sub-paragraphs, which follow highlight the problems and their origins in more detail.

4.2.2.1 Main Structure and Steel Surfaces

All detailing allowing for sharp edges, indicates potential 'corrosion zones', which cannot be cleaned or coated effectively [Brown, 1995]. Obstructions on the bottom flanges of girders prevent free passage of water along their surface i.e. can lead to pooling. Upward curvature to the bottom flange of plate girders is a common occurrence during fabrication and has a similar effect of collecting water. Where unavoidable obstructions occur, for example at bearing stiffeners or abutment diaphragms, lack of preventive measures for allowing water to drain, such use of notches or 'cope' holes, leads to trapped water. Often, substituting bearing stiffeners with transverse stiffeners also leads to entrapment of moisture and dirt. Poor design provisions for the bridge deck drainage system also result in increased vulnerability of these areas, as it allows discontinuities in deck structure. Another area for difficult corrosion detection is in box girders as the lack of ventilation or drainage impairs the quality of design.

4.2.2.2 Bearings and Expansion Joints

Bearings and expansion joints are regular sources of deterioration problems. Virtually all expansion joints leak, allowing salty water through in the winter, which leaks onto the structure and bearings. Location of expansion joints (due to their relative fragility in comparison with the rest of the structure and their notorious defective construction), lack of detailing facilitating the maintenance and replacement of bearings and expansion joints, and lack of drainage provision for water which leaks through expansion joints, are areas for particular attention during inspection.

4.2.2.3 Parapets

Parapets are susceptible to periodical damage from vehicle impact, vandalism. These can be awkward with exposed fixing details and elaborate shapes that impede periodic repainting and easy replacement.

4.2.2.4 Waterproof Membranes and Road Surfacing

Lack of a sound waterproof membrane, preventing salt-water penetration, through a bridge deck, is one of the most important items preventing durability. On a bridge with a stiffened steel plate deck, incorrect surfacing also contributes to the poor wear out resistance of the deck. Inadequate provision in design for the expansion of the structural steel to the deck reduces integrity and surface resistance to wear and fatigue.

4.2.2.5 Steel Decks

Although, stiffened steel deck plates are dynamically loaded (wheel loads from traffic), a high fatigue category is not always assumed in the design [Troitsky, 1990]. Many premature failures of decks occur because the welds connecting the longitudinal stiffeners (or stringers) to the plate are of poor quality. The weld flexes as wheel loads pass close on each side, and fatigue occurs. If welds are not butt welds and the stiffeners are not pressed into hard contact with the plate with deep penetration fillet welds, a fatigue zone is likely to develop.

Where the stringers are designed solely for static load, fatigue problems have occurred at the splice between the continuous lengths of stringer of inadequate size.

Even stiffened steel decks are still flexible, and this continued flexing is likely to cause fatigue in the surfacing. As the surfacing tends to act compositely with the steel deck plate, fluctuating tensile strains develop in the top surface. This problem is particularly serious for a hard line support, such as web to a main girder. Therefore, when a longitudinal groove about half the depth of the surfacing is absent above such hard-line supports, a potential problem area is created. The above infers that weld inspection is important for bridge decks.

4.2.3 Common Problems Area Geometries for Variety of Bridge Types

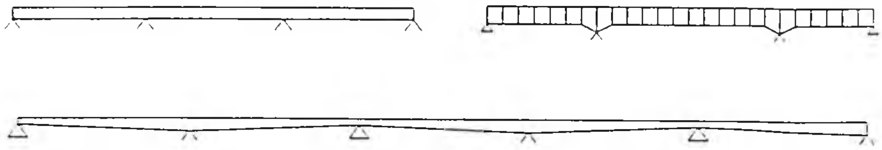
Bridges, which serve as sources of inspection and repair task details, are numerous (British as well as Dutch, French, Australian and American). In order to identify and select the most frequently occurring areas [BCSA, 1968], [BSC-Bridges in Steel Series] and [Garside, 1992] Tables 4.1 and 4.2 are assembled below, showing the most typical bridge elevations and cross-sections with dimensions ranges, to facilitate the identification of further areas for investigation.

Table 4.1 Portfolio of Bridge Types

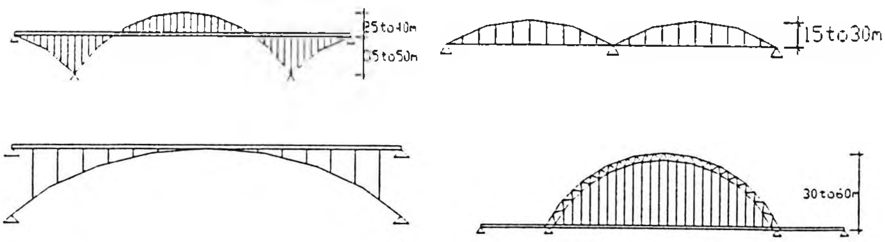
Table 4.2 Portfolio of Bridge Cross-sections

Portfolio of Bridge Types

Bridge Type	Span Range
Beam/Girder (flat, haunched and boxed)	50 to 220m (max. 300m)

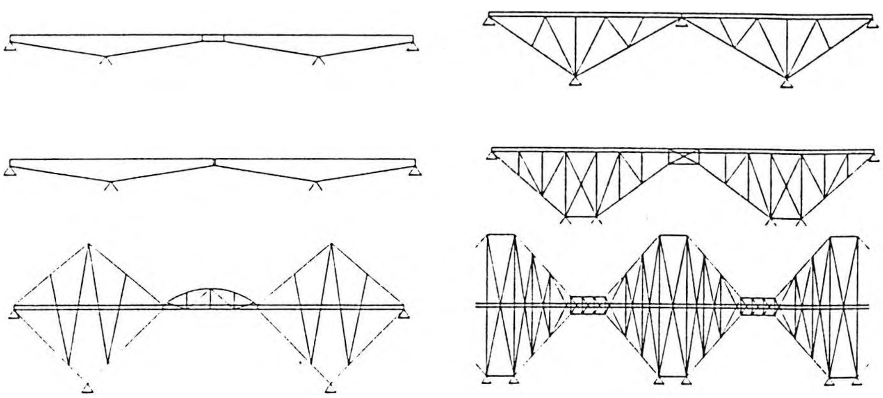


Bridge Type	Span Range
Arch	50 to 300m (max. 504m)

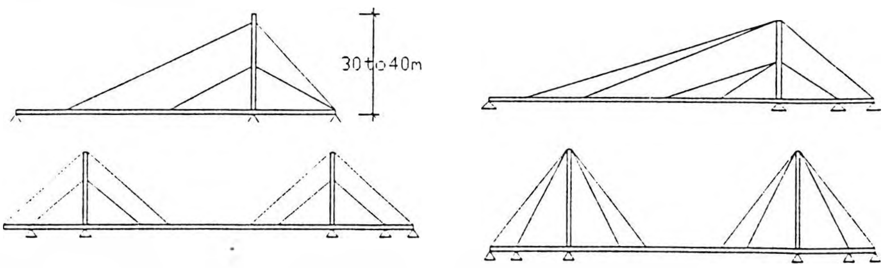


Bridge Type	Span Range
Trussed	100 to 350m

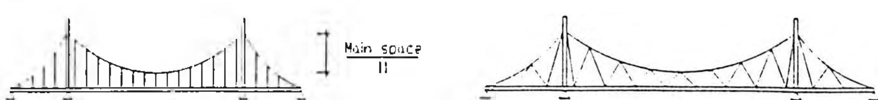
Bridge Type	Span Range
Beam - cantilevered and Truss - cantilevered	100 to 550m



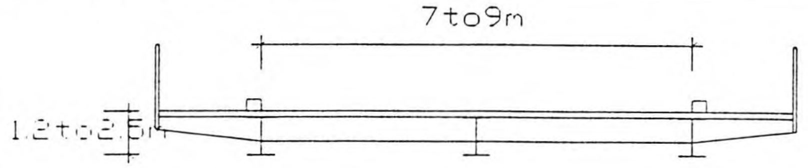
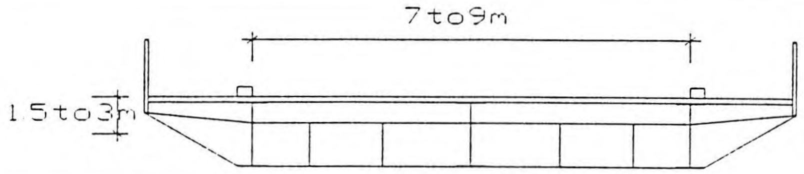
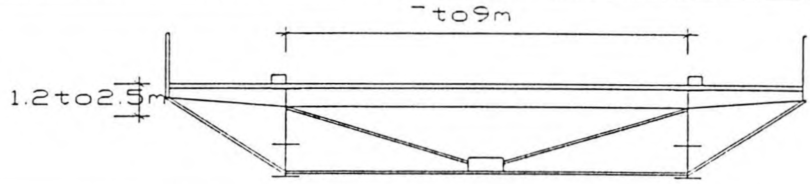
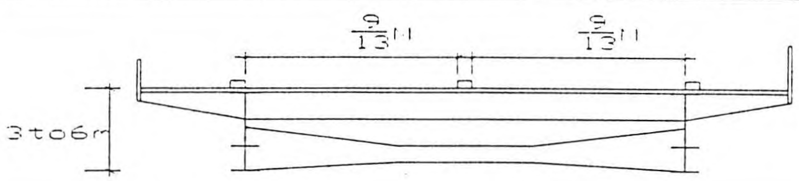
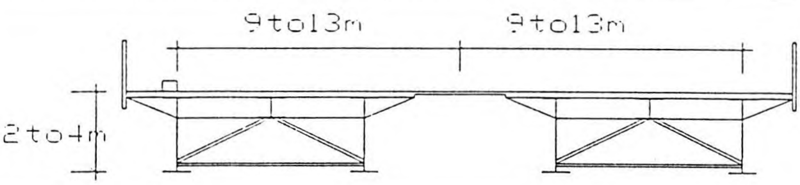
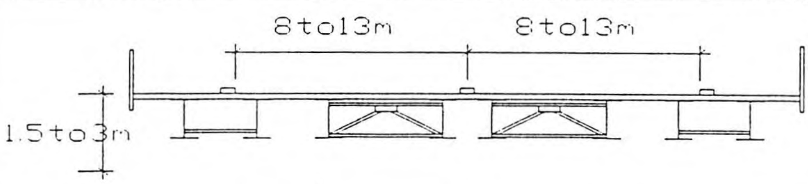
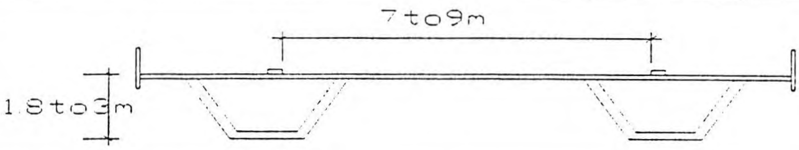
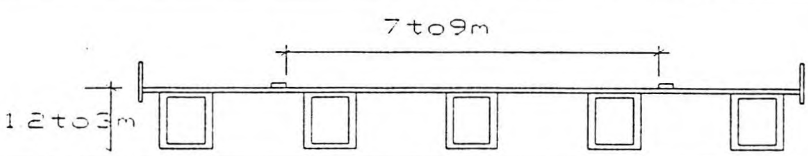
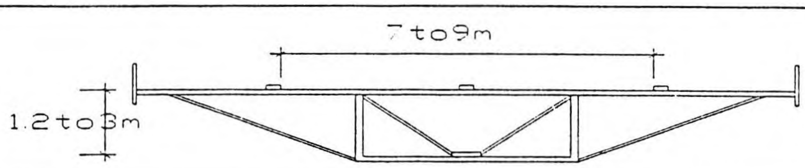
Bridge Type	Span Range
Cable - stayed	100 to 450m (max. 860m)



Bridge Type	Span Range
Suspension	350 to 1375m (max.1410m)



Portfolio of Bridge Cross-sections

Cross-section	Type
	UB or PG open system
	UB or PG enclosed system
	Haunched PG and K-bracing
	Haunched 2x PG and boxed diaphragm
	4x PG and K-bracing
	Multiple PG, K-bracing and straight bracing
	Twin open box girders
	Multiple box girders
	Single open box with bracing inside

Common tasks are represented as user specified set of points e.g. corner area, pier/pylon-u/s of deck, u/s of deck, side face of the pier, etc. and are briefly described in Table 4.3 below and drawn accordingly in Figures 4.1-4.15 in the following Section 4.3.

Whilst Tables 4.1-4.3 are an attempt to generalise bridge zones of all bridge types, several characteristics do become apparent. Although, bridges vary in types, sizes and structure, the basic cells are similar and repetitive. Therefore, the library of 'basic cells' can be assembled on the basis of adding new items from analysing different structures, with Table 4.3 as a starting point.

Table 4.3 Examples of Typical Repetitive Areas on Bridges

Beam/Flat Girder [A] single or multi-span structure acting as a continuous beam	Truss and Truss Cantilever [B]	Cantilever [C] middle spans either cantilevered or suspended	Arch [D] Main structural member is the arch	Suspension [E] central span with two side spans and the cables passing over the top of the supporting piers are anchored in tunnels, the road is suspended from the cables by vertical hangers Cable-stayed
Corner between two secondary beams, main beam (girder) and composite decking over [A1] Figure 4.1	Area between main truss, u/s of deck, main pylon, pylon's bracing and cross stiffeners [B1] Figure 4.4	Area underside decking and along the hinge of double cantilever truss [C1] Figure 4.7	Area around top of steel arches, cross bracing and hangers [D1] Figure 4.10	Area at the top of steel pylons, main cables and hangers [E1] Figure 4.13
Area under cantilevered edge, between secondary girders, main girder with stiffeners and u/s of decking [A2] Figure 4.2	Area between external main trusses, u/s of decking and secondary girders [B2] Figure 4.5	Area between two haunched at the piers main girders, K-bracing, cross beams and u/s of decking [C2] Figure 4.8	Area between steel arches embedded in abutment, u/s of main and secondary structure and hangers [D2] Figure 4.11	Area underside the aerofoil box deck of a suspension bridge and around the pylon [E2] Figure 4.14
Area between main girders, crossed bracing and metal decking above [A3] Figure 4.3	Main trusses, cross bracing and u/s deck [B3] Figure 4.6	Area between two haunched girders, boxed diaphragm, cross beams and u/s of decking [C3] Figure 4.9	Area along top of trussed arch (top of bow-string girder) [D3] Figure 4.12	Area on top of walkway and part of aerofoil box deck of a suspension bridge and around the pylon [E3] Figure 4.15

4.3 Benchmark Library

Here, the objective is to establish a suitable structure in which the geometry of bridge structures can be set in a computer based library of inspection task. Such a library should be built comprising data such as bridge types, geometry, descriptions of inspected surfaces, locality and properties of the encountered materials. These ingredients contribute to the benchmark tasks.

Each benchmark includes a geometrical description of localised structural components, their location on the structure and specifications of the surfaces zones, and motion sequences to be applied in performing inspection and restoration tasks on them. Figures 4.1-4.15 not only illustrate selected details on variety of bridges, as described in Table 4.3, but also indicate set of target points, combined into four paths per detail which suggest the sequential location of the robot's end effector armed in NDT probe or blasting nozzle. The GRASP robot simulation facility is used, in order to verify proposed robot's suitability, at the final stages of the dissertation.

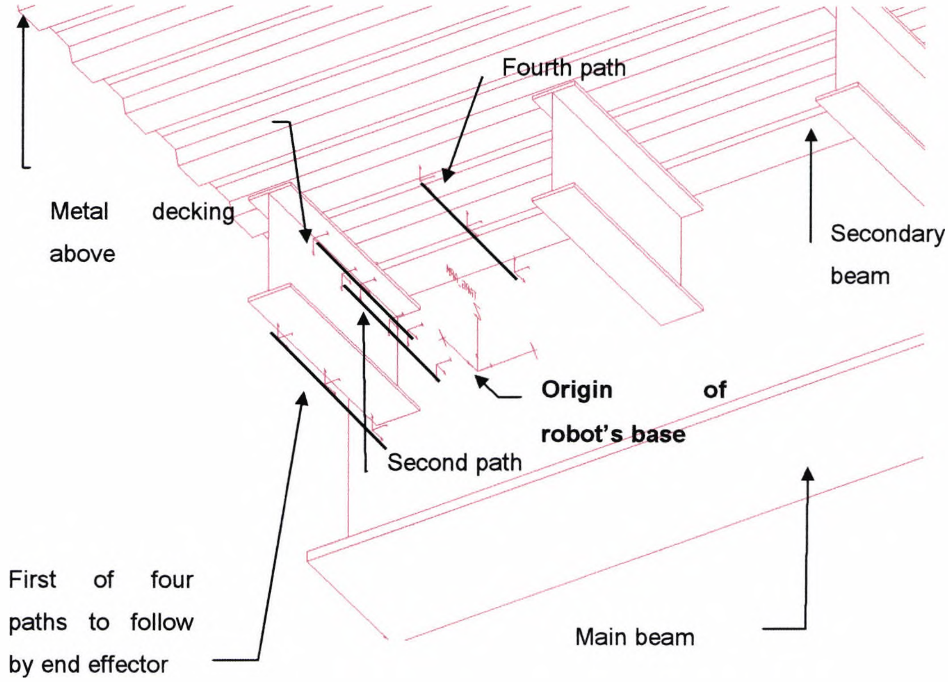


Figure 4.1 Bridge Detail [A1] according to Table 4.3

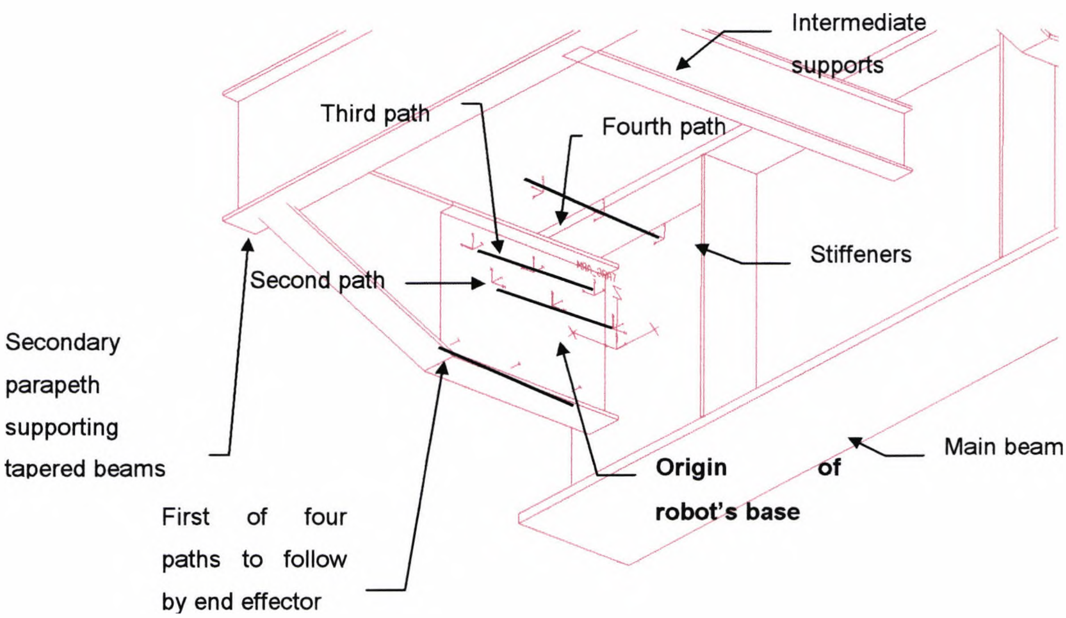


Figure 4.2 Bridge Detail [A2] according to Table 4.3

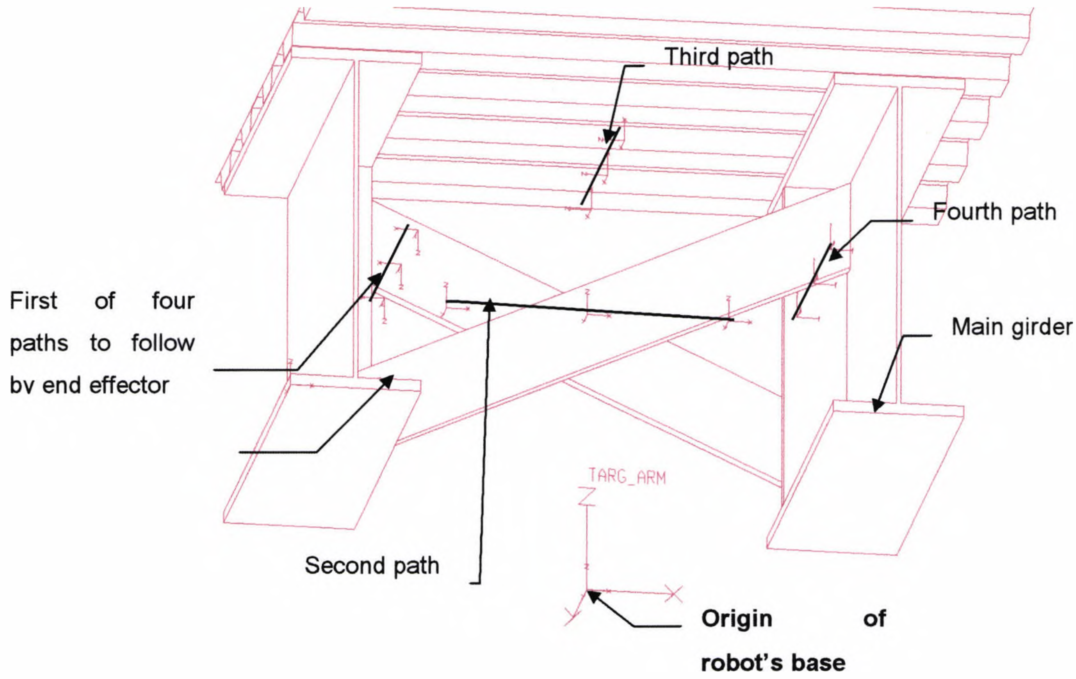


Figure 4.3 Bridge Detail [A3] according to Table 4.3

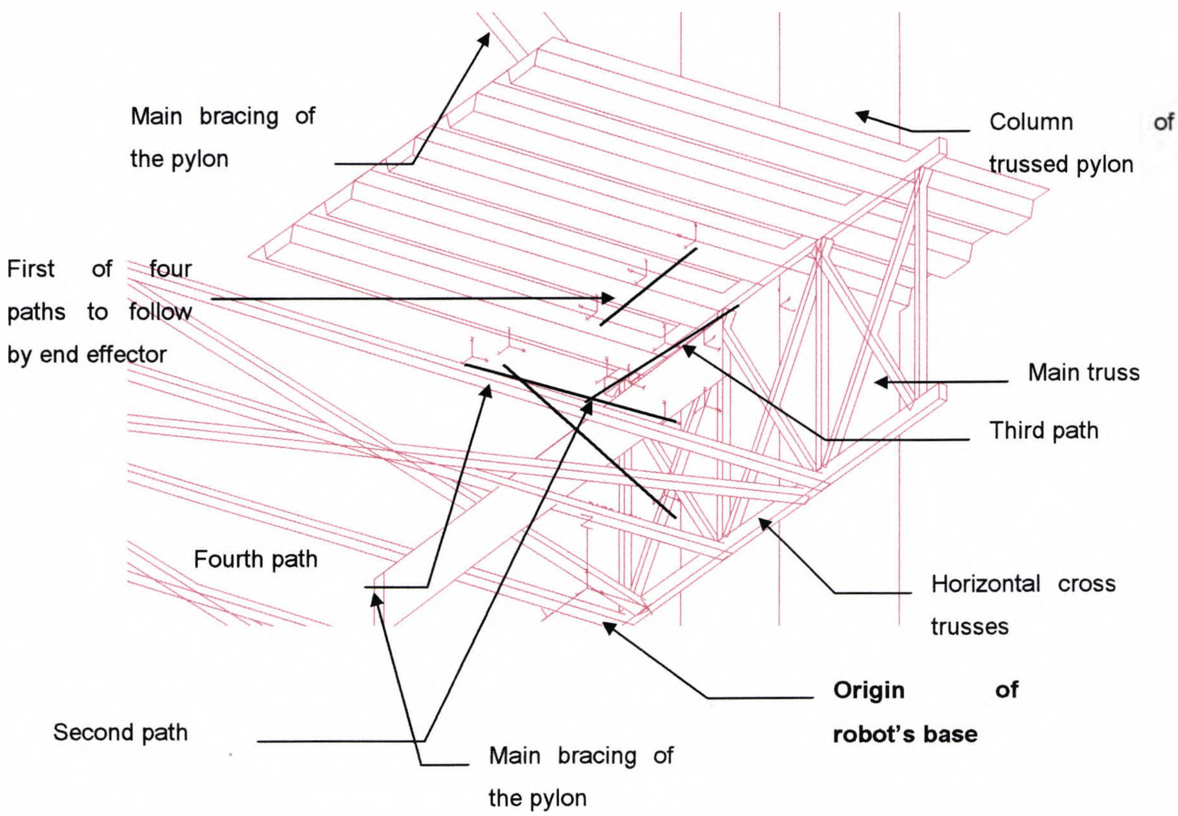


Figure 4.4 Bridge detail [B1] according to Table 4.3

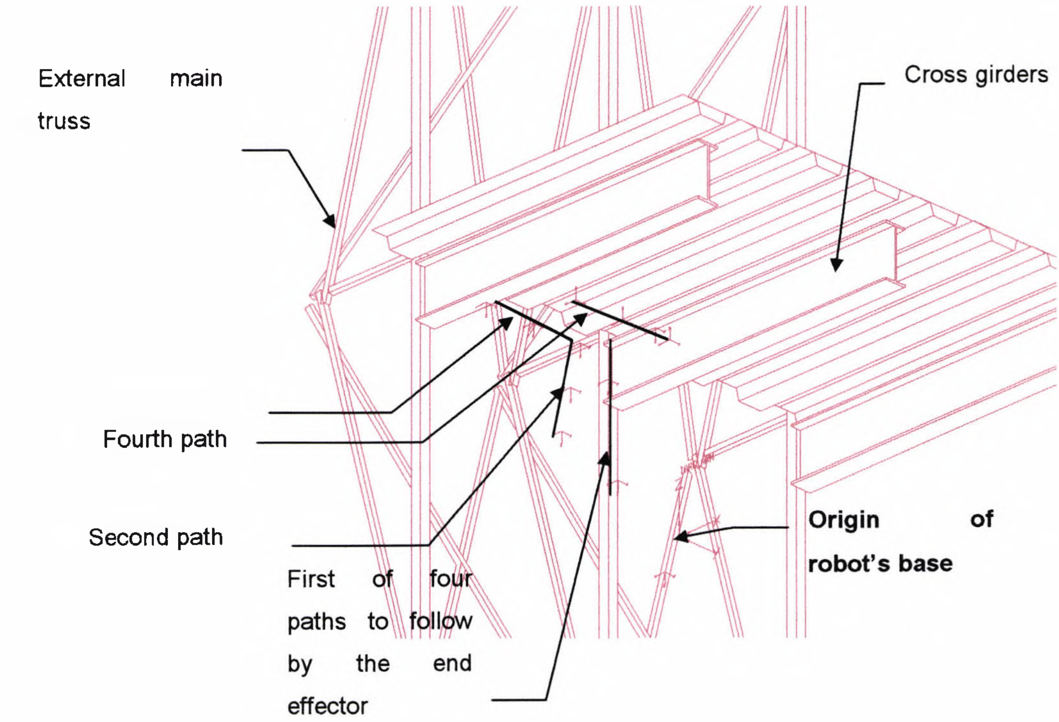


Figure 4.5 Bridge Detail [B2] according to Table 4.3

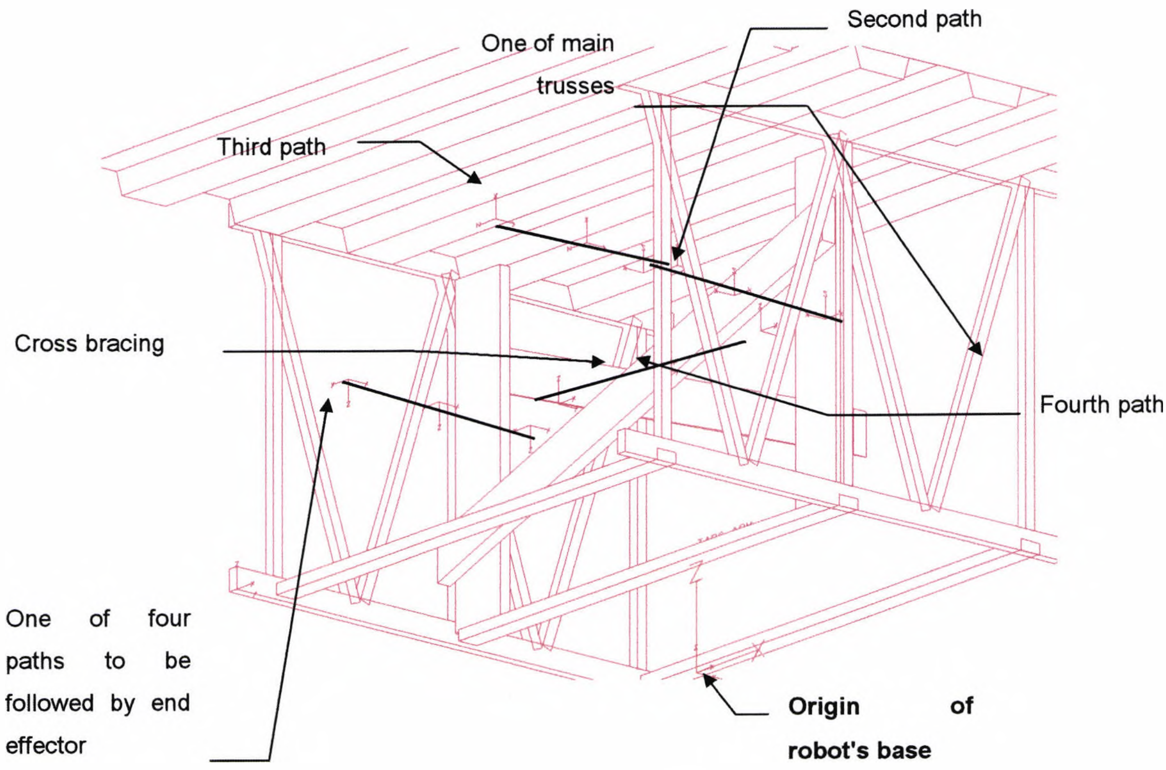


Figure 4.6 Bridge Detail [B3] according to Table 4.3

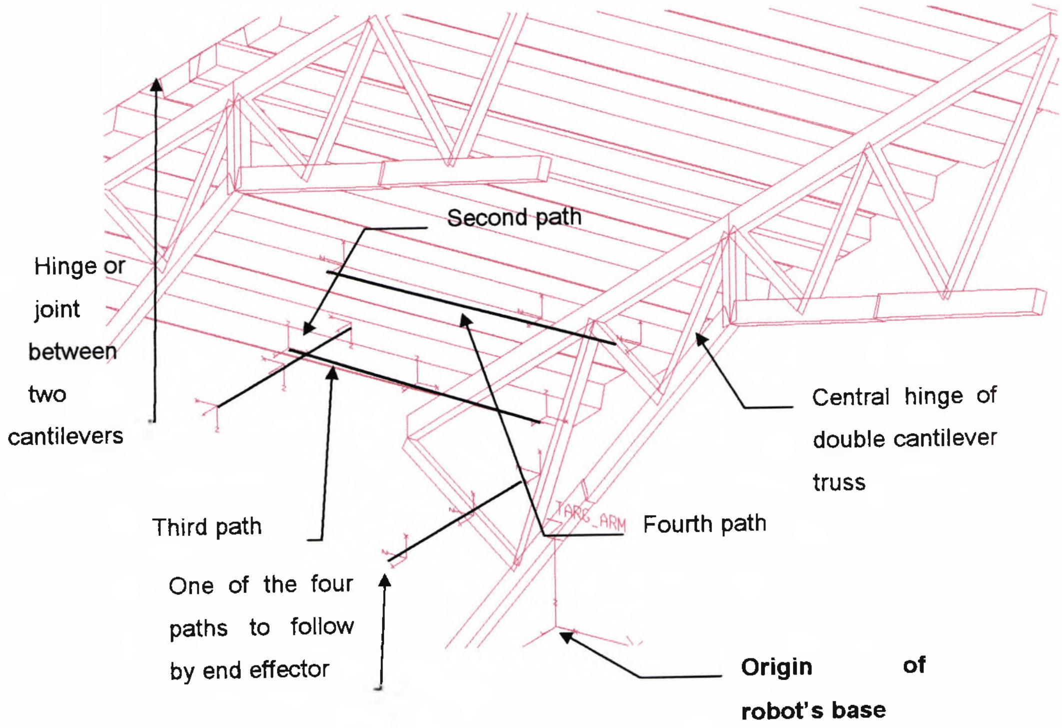


Figure 4.7 Bridge Detail [C1] according to Table 4.3

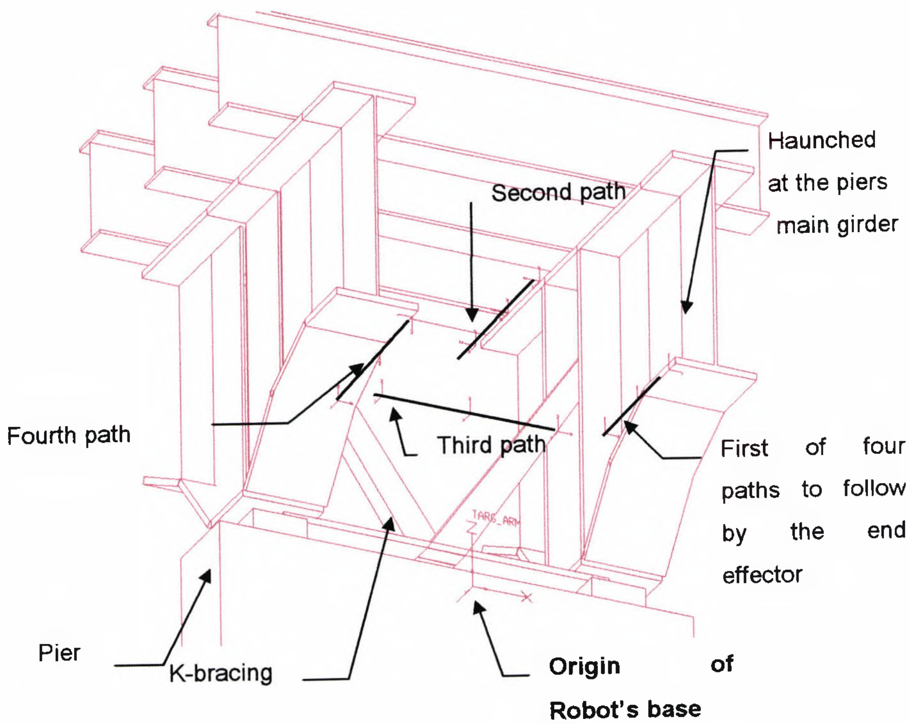


Figure 4.8 Bridge Detail [C2] according to Table 4.3

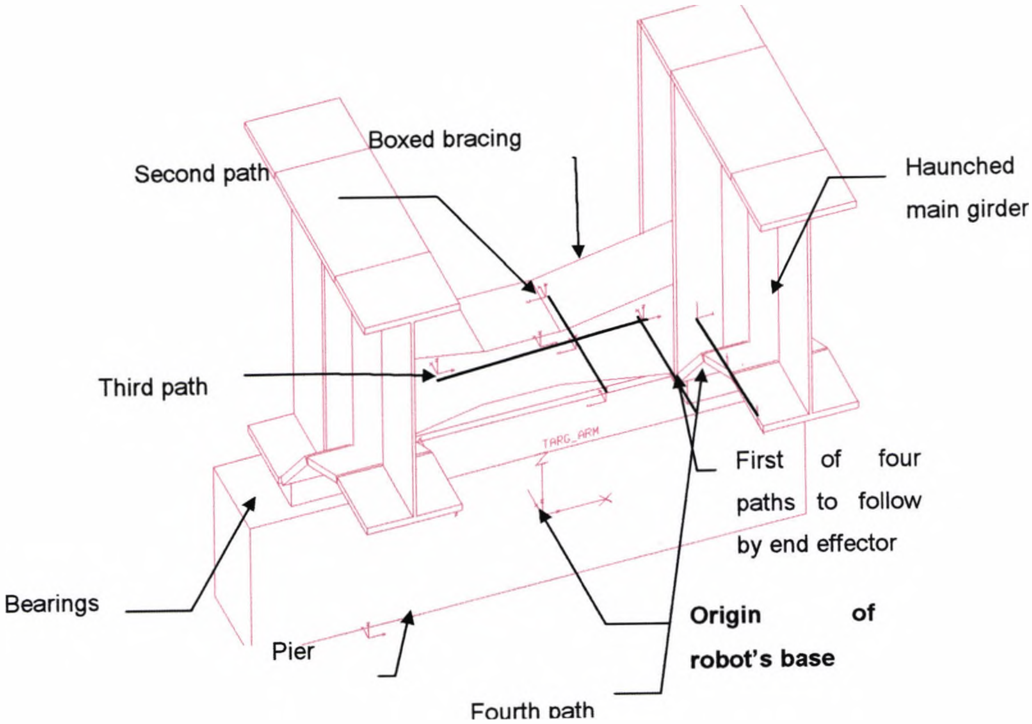


Figure 4.9 Bridge Detail [C3] according to Table 4.3

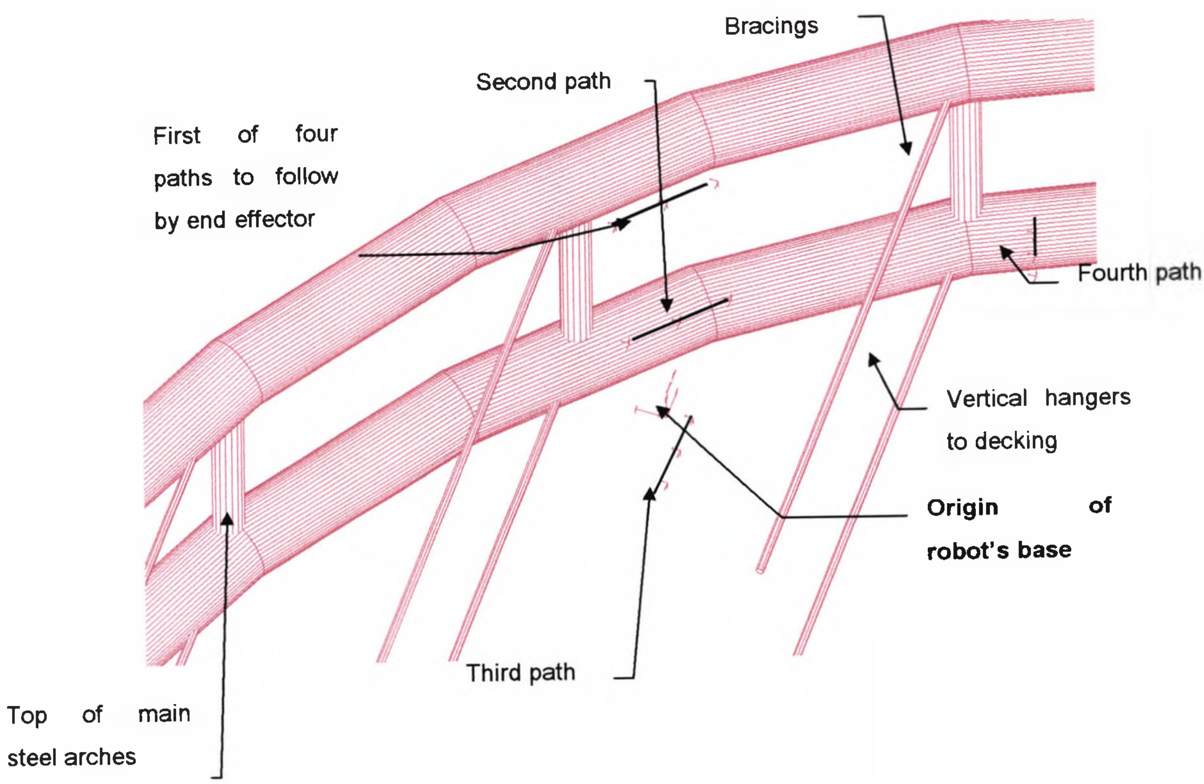


Figure 4.10 Bridge Detail [D1] according to table 4.3

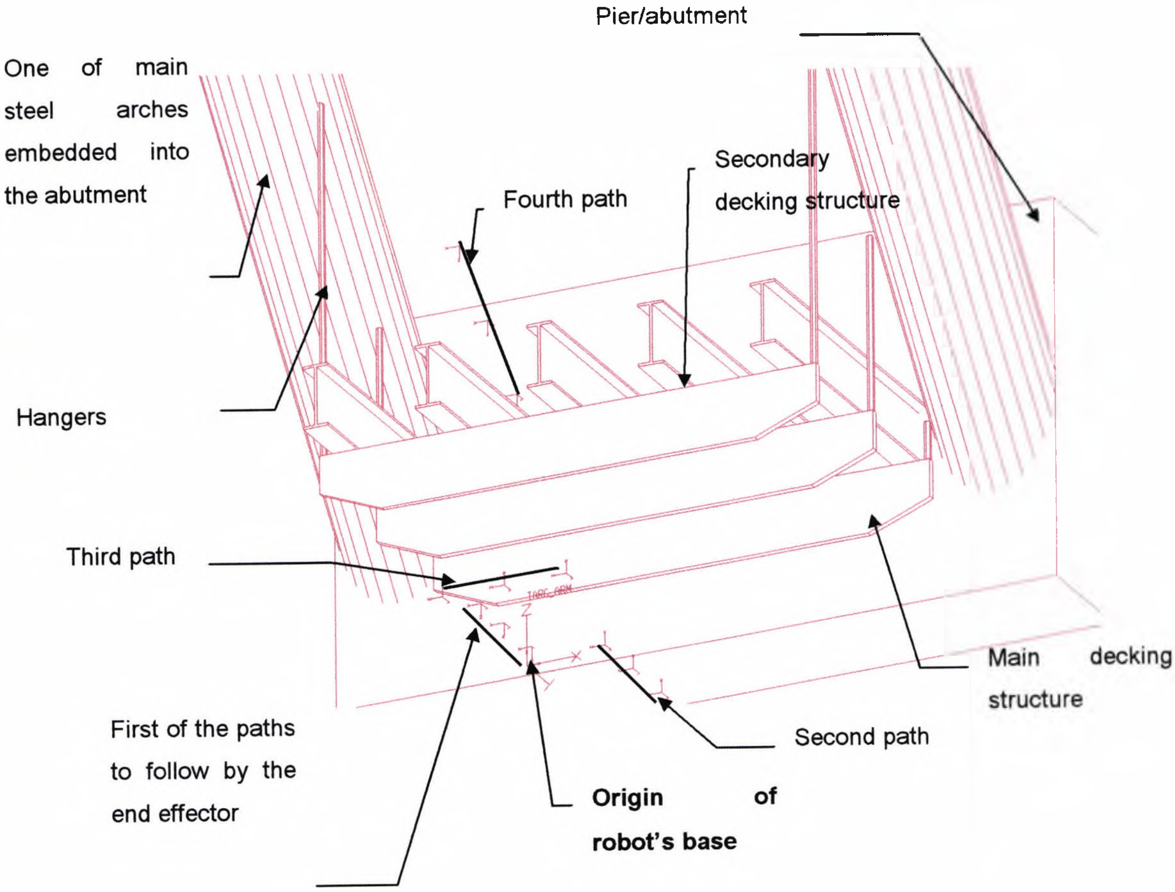


Figure 4.11 Bridge Detail [D2] according to Table 4.3

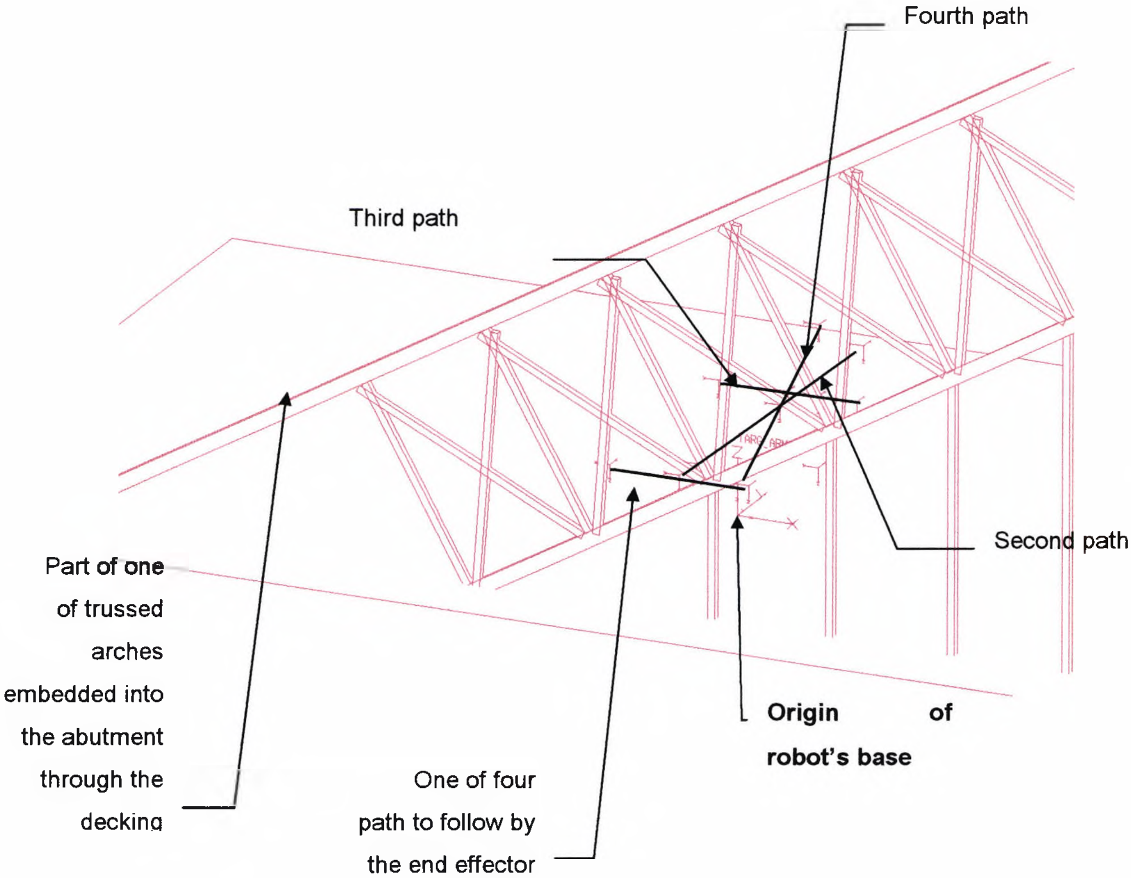


Figure 4.12 Bridge Detail [D3] according to Table 4.3

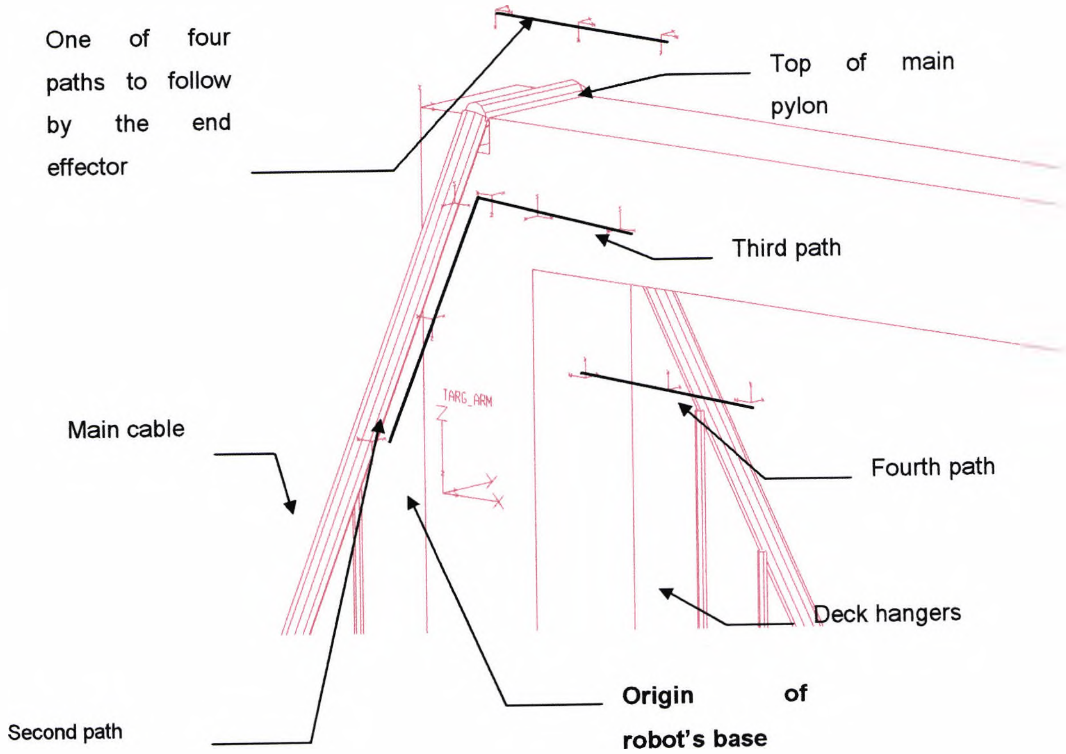


Figure 4.13 Bridge Detail [E1] according to Table 4.3

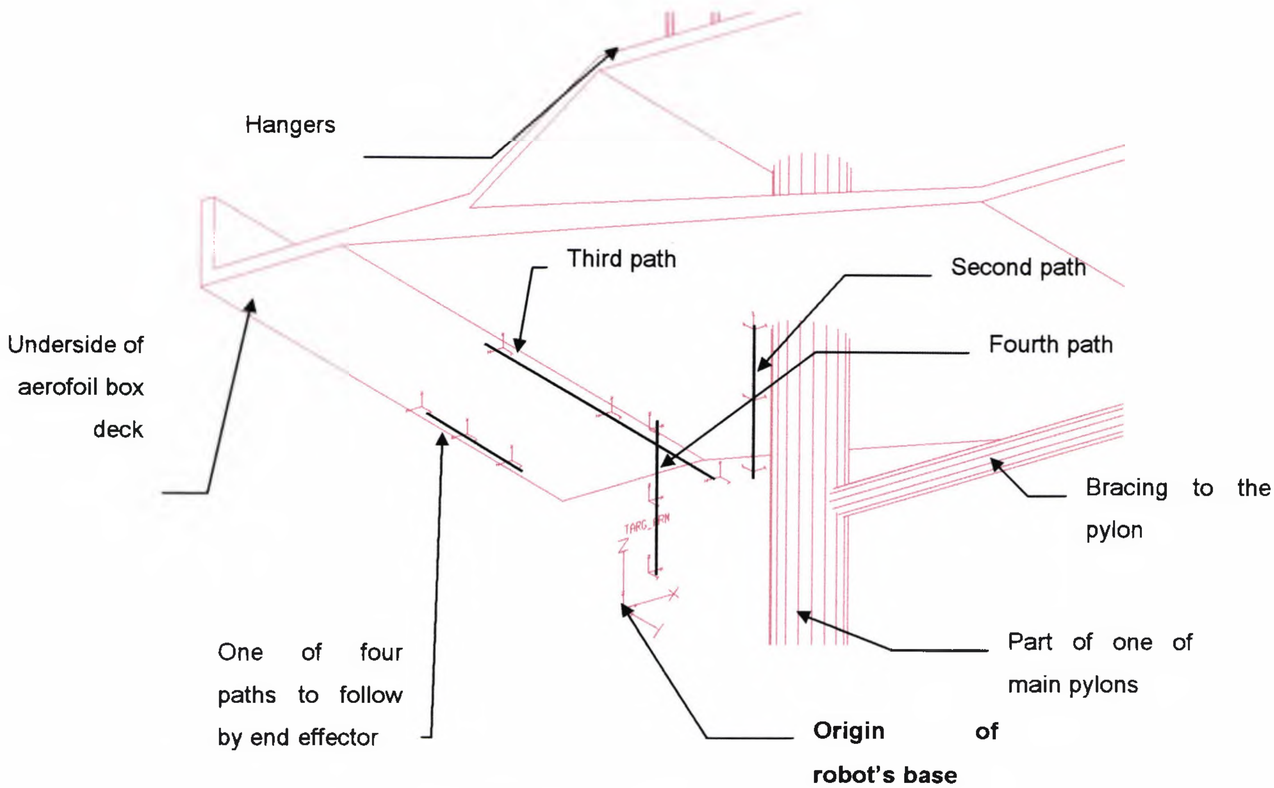


Figure 4.14 Bridge Detail [E2] according to Table 4.3

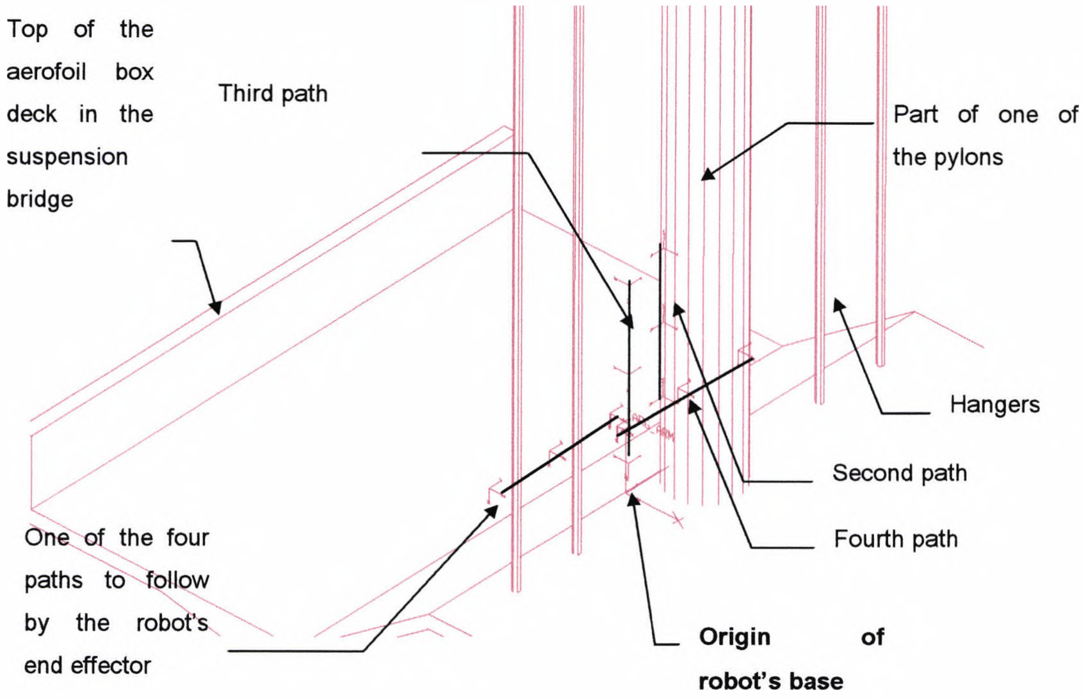


Figure 4.15 Bridge Detail [E3] according to Table 4.3

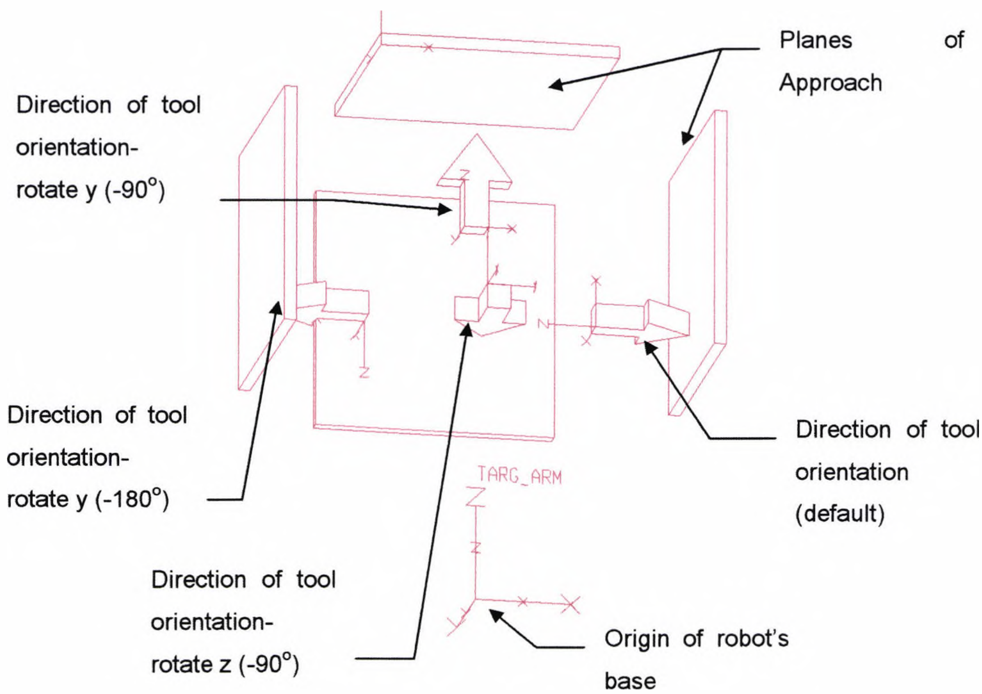


Figure 4.16 Schematic Idealisation of Tool Orientation

4.4 Summary

Steel bridges, although varying in size, geometry and structure, have similar areas affected by corrosion. The main purpose of this chapter was to identify these areas and translate them into computer based graphical representations, in order to create a library of benchmarks. The bridges' details were selected based on the average size of the single workcell, which can be attended to by a robot from a single workstation, without relocation. The geometry of the bridge details could provide further information about the collision envelope and paths to be followed by the tool or the probe, which contribute to achieving versatility of the proposed automated device. Despite distinctions among the details certain arrangements of elements and surfaces, within locations of highest corrosion risk, remain similar in structure and size. This observation helps to arrive at a solution, which has potentially higher precision and wider relevance. Adding the vectors of approach for the tool (in the form of appropriately oriented Targs) gives the basis for choosing the type of the manipulator (arm plus wrist). The contents of this chapter are summarised as the working environment identification for task benchmarking. The following chapter deals with inspection and repair methods, their characteristics, automation suitability and modes of operation.

CHAPTER 5 - INSPECTION AND RESTORATION TECHNOLOGY

5.1 Introduction

In the previous chapter, the types and location of deterioration processes have been identified for the problem areas occurring in the majority of bridges. This chapter investigates the inspection and repair tools and techniques for assessing their automation applicability. Its purpose is to quantify the automation task from the point of view of tool handling.

Many considerations arise when removing, replacing or overcoating surface coatings. They include: (i) feasibility of overcoating, without removing existing coat, (ii) balance between local re-application of coating and overall surface preparation prior to coating, (iii) choice between and necessity of partial or full containment, (iv) level of surface preparation and (iv) workers' exposure and protection. Vetted inspection and assessment is clearly a key issue in this.

Evolution of the inspection tools and techniques and restoration methods is an important requirement for automation study. It is necessary to establish, which tools and methods have good automation factors, those more easily employed in automation due to their simplicity of application, type of power support, storage of information, speed of interpreting results, sensitivity and other factors.

The author's original contribution is in the up-to-date compilation of the automation suitability classification of restoration tools and techniques. This information is essential for the prospective device's design, application and control. The research and assessment carried out in this chapter is an essential pre-requisite to the analysis of the enabling technologies and commercially available partially automated devices for inspection and/or restoration on steel surfaces, which is carried out in chapter five.

5.2 NDT Tools and Techniques

NDT (Non Destructive Testing) is testing that does not involve the destruction of the test piece or component, or impair its designed use [Hull and John, 1988]. An area where NDT is of great importance is the detection of (internal and external) faults. External faults may involve machining marks, external damage, damage of surface finish, cracking, stress corrosion, fatigue and corrosion. Internal faults may involve bolts, rivets, over-stressing and hydrogen embrittlement. Standard terminology and specification for NDT are given in BS 3683:1985 and subsequent five parts, specifying use of penetrant, magnetic particle, radiological, ultrasonic and eddy current flaw methods. The main defects and NDT techniques are given in Table 5.1. Electromagnetic induction and acoustic emission have not been added to the table but they are assessed in Sections 5.2.5 and 5.2.6. Table 5.1 permits a rapid assessment of the strength and weaknesses of each technique and enables selection of the methods most suitable for a particular application. Detailed description of the techniques is given in the remaining sections of this chapter.

Typical surface conditions encountered on bridges older than five years are as follows: (i) tight and intact paint with no rusting, (ii) tight and intact paint with some visible rust, (iii) finish coat worn down to primer, (iv) flaking off mill scale to bare metal, (v) flaking, blistering, alligatoring paint, (vi) bare steel corroding, tight rust, (vii) heavy rusting and pitting, loose flaky scale and (viii) contamination from oil, dirt, debris [Addleston and Rice, 1991]. It is virtually impossible to separate these defects and allocate a particular NDT technique to every single one of them [Golls, 1991]. Many defects are combination of others and, similarly, the testing techniques can then equally be used in a combination style.

Table 5.1 Suitability Comparison of NDT Techniques for Coated Steelwork

General Test Methods	Penetrating Radiation Tests			Ultrasonic and Sonic Tests									Magnetic Particles Tests				Electromagnetic Tests				Liquid Penetrant Tests (discontinuities opened to the surface only)			Other
				Contact Pulse Reflection			Immersion Pulse Reflection																	
Major Variations in Tests	Film Radiography	Fluoroscopy (thin sections only)	Radioisotopes	Normal Beam	Shear Wave	Surface Wave	Normal Beam	Angle Beam	Surface Wave	Through Transmission	Resonanse	Natural Frequency	Alternating Current (wet method)	Alternating Current (dry method)	Direct Current (wet method)	Direct Current (dry method)	Eddy Current	Magnetic Field (Magnetic Flux Leakage)	Leakage Field Pick-up	Direct Current Conduction	Visible Dye Penetrants	Fluorescent Dye Penetrants	Filtered Particles	Electrified Particles
Minute Surface Cracks	0	0	0	0	0	0	0	0	0	0	0	0	3	2	3	2	2	1	2	2	2	3	0	0
Normal Surface Cracks	2*	0	2*	0	1	3	0	2	2	0	0	1	3	3	3	3	2	2	3	2	3	3	0	0
Internal Cracks	2*	1	2*	3	3	0	3	3	0	2	1	0	0	0	1	2	0	1	2	1	0	0	0	0
Internal Voids	3	2	3	3	3	0	3	3	0	3	1	0	0	0	0	1	0	1	0	1	0	0	0	0
Thickness	2	0	2	1	0	0	2	0	0	0	3	0	0	0	0	0	1	2	0	2	0	0	0	0
Metalurgical Variations	2	0	2	1	0	0	2	0	0	2	0	0	0	0	0	0	3	3	0	1	0	0	0	0
Sheet and Plate Thickness	3*	0	3*	1	0	0	2	0	0	0	3	0	0	0	0	0	1	2	0	2	0	0	0	0
Sheet and Plate Laminations	0	0	0	3	2	0	3	2	0	3	3	0	2	2	3	3	0	0	0	1	2	2	0	0
Sheet and Plate Holes	3	3	3	3	2	0	3	2	0	0	1	0	0	0	0	0	0	0	1	1	1	1	0	0
Bars and Tubes Seams	1	0	1	0	2	2	0	2	2	2	1	0	3	3	3	3	2	3	3	2	3	3	0	0
Pipes	3	0	2	3	0	0	3	1	0	3	3	0	1	0	1	0	0	2	0	0	2	2	0	0
Bars and Tubes Cupping	3	0	2	3	0	0	2	0	0	3	1	0	0	0	0	1	0	2	1	1	0	0	0	0
Bars and Tubes Inclusions	2	2	2	2	2	0	3	2	0	2	0	0	2	0	3	0	0	1	1	0	0	0	0	0
Castings - Cold Shuts	3	2	3	2	2	1	2	2	0	2	0	0	3	3	3	3	1	1	1	0	2	2	0	0
Castings - Surface Cracks	2*	0	2*	0	1	0	0	1	1	0	0	1	3	3	3	3	2	2	2	2	3	3	0	0
Castings - Internal Shrinkage	3	2	3	2	2	0	1	1	0	2	0	1	0	0	1	2	1	1	0	0	0	0	0	0
Castings - Holes and Porosity	3	2	3	2	2	0	2	2	0	2	1	0	0	0	0	1	0	1	1	0	3	3	0	0
Castings - Core Shift	3	2	3	2	0	0	1	0	0	0	3	0	0	0	0	0	1	1	0	1	0	0	0	0
Forgings - Laps	2	0	0	2	1	1	1	0	1	0	0	0	0	3	3	3	1	1	2	2	2	2	0	0
Forgings - Inclusions	2	0	0	3	2	0	3	2	0	3	0	0	0	2	3	2	0	1	1	0	0	0	0	0
Forgings - Internal Bursts	2	0	3	3	3	0	3	3	0	3	0	0	0	0	2	2	0	2	1	0	0	0	0	0
Forgings - Internal Flakes	2	0	0	3	2	0	2	2	0	3	0	0	0	0	2	2	0	1	1	0	0	0	0	0
Forgings - Cracks and Tears	2*	0	2*	1	2	2	2	2	0	0	0	1	3	3	3	3	1	1	2	2	3	3	0	0
Welds - Shrinkage Cracks	3*	2*	3*	2	2	2	2	3	1	1	1	0	3	3	3	3	2	1	3	2	3	3	0	0
Welds - Slag Inclusions	3	2	3	2	2	0	2	2	0	2	0	0	0	0	1	3	1	1	2	0	0	0	0	0
Welds - Lack of Fusion	3	2	3	2	2	0	2	2	0	3	0	0	2	2	2	3	2	1	2	1	0	0	0	0
Welds - Porosity	3	2	3	2	1	0	2	2	0	2	0	0	0	0	0	2	0	0	0	0	0	2	0	0
Welds - Lack of Penetration	3	2	3	1	2	0	2	2	0	0	0	0	0	0	0	2	1	1	2	0	0	0	0	0
Heat Treat Cracks	1	0	0	0	2	2	0	2	2	0	0	0	3	3	3	3	3	3	3	2	3	3	0	0
Grinding Cracks	0	0	0	0	0	0	0	3	2	0	0	0	3	3	3	3	2	2	2	2	3	3	0	0
Fatigue Cracks	1	0	1	0	2	2	0	2	2	0	0	0	3	2	3	2	2	2	3	2	3	3	0	0
Stress Corrosion	2	0	1	0	0	2	0	0	1	1	0	0	3	2	3	2	0	0	0	2	3	3	0	0
Blistering	1	0	1	0	0	0	0	0	0	2	3	0	2	0	0	0	0	0	0	0	0	0	0	0
Thinning	2	1	2	0	0	0	0	0	0	0	3	0	0	0	0	0	2	1	0	2	0	0	0	0
Corrosion Pits	3	1	1	1	1	0	0	0	0	0	2	0	0	1	0	1	0	0	0	0	2	2	0	0

* - beam parallel to crack | 0 - unsuitable | 1 - poor | 2 - fair | 3 - good

5.2.1 Visual Methods

Detection of surface faults is important, including cracks, blowholes and surface irregularities. Size and shape comparison is also important, where the eye is quick to detect missing parts or cavities. Optical assistance means, such as lamps and magnifiers often come to aid. In the automated approach, this work can be supported by use of fibre optics or cameras as part of closed circuit TV, with video recording.

Whilst much usual inspection work and assessment is performed manually, there are increasing moves toward the use of supporting technology. This employs video camera systems with a frame grabber and picture store, which is often accompanied by a time lapse VHS cassette storage and PC-based picture library, allowing examinations to take place off-site, automation suitability is assembled in Table 5.2.

Table 5.2 Automation Characteristics and Assessment for the Visual Methods

Main Application	Cracks, visual and geometrical defects.
Description and Specification	3" camera monitor measures and inspects cracks using a movable perspex guide with 0.1mm graticules. Weight of a probe - 2 KGs, probe support (video recorder) - 6 KGs. Cost - approx. £3,000
Sensor Motion Requirements for Automation	Camera should be moved at constant speed and be positioned at right angle and close distance to the surface (offset here depends on the type of equipment). The path of movement should be programmed according to the bridge geometry.
Advantages	(1) low cost; (2) immediate data for viewing and analysis; (3) little skill required from the operator.
Disadvantages	Uncontrolled offset affects field of view and crack width and length.

5.2.2 Liquid Penetrant Tests

Penetrant processes are divided into two basic groups, visible and fluorescent. Visible penetrants are those that contain a very bright dye, usually red, which is after developing, viewed under bright white light. Fluorescent penetrants contain a dye, which fluoresces under filtered ultra-violet ray (black light).

Both categories are sub-divided into three groups, depending on the penetrant removal method:

i. Water washable - penetrants soluble in water are used, these achieved by adding emulsifiers during manufacturing;

ii. Post-emulsified - penetrant is not soluble in water, but made so by the addition of an emulsifier as part of the cleaning process;

iii. Solvent removable - penetrant can only be removed using a suitable solvent.

An essential part of the process is the developer, which is essentially a white powder whose function is to assist development of the penetrant indications, in order to make them more readily visible. There are three basic types of the developers, dry powders, solvent suspended and water suspended. Their correct preparation is imperative to obtain good, even coatings, and not distort results.

It has been found that post-emulsified penetrants are capable of entering the finest cracks and, because they are not water soluble, do not wash easily from wide opened cracks. Water washable penetrants do not enter fine cracks as readily and can be more easily removed from wide, shallow cracks. However, post-emulsified penetrants are very difficult to remove from rough surfaces and they tend to reduce overall sensitivity by having high level of background contamination. Therefore, it is common to use post-emulsified penetrants for the examination of smooth surfaces, where very small tight cracks are sought, and water washable penetrants for the examination of rough surfaces, such as castings.

The process of crack detection using penetrants consists of three stages, which need different equipment and application, if it is to be automated: (i) pre-cleaning and (ii) application of the penetrant and developer and (iii) observation.

The application of the process starts with the pre-cleaning of the parts to receive the penetrant. It is vital that the surface is clean [Lovejoy, 1989], particularly when dealing with in-situ materials, such as bridge stanchions. Then, the penetrant is applied in sufficient quantity, to thoroughly wet the area under inspection and prevent drying. The ideal temperature for testing is between 15°C to 40°C, as higher temperatures lead to evaporation of penetrant's lighter fractions. Lower temperatures slow down the whole process. Allowed penetration time is up to twenty minutes, and after a specified time, the surplus must be removed from the surface. The process must be quick and efficient to ensure a clear background from which inspection can be made. The penetrant can then be dried in warm air (80°C, when possible), to make way for the developer. The developer forms a thin absorbent layer, of uniform appearance, without contamination or discoloration. Then, the inspection can take place under the appropriate lighting conditions. After application of the developer, time must be allowed for indications to develop. Bearing the above information in mind, the automation suitability assembled in Table 5.3 indicates low applicability.

Table 5.3 Automation Characteristics and Suitability of Penetrant Tests

Main Application	The method can only be used to detect defects which are opened to the surface of a homogenous material; the most certain characteristics of a surface crack are indicated - an open discontinuity in a surface [Lovejoy, 1991]. The most common applications: checking for cracks in welds and leak testing thin wall vessels.
Description and Specification	Impossible to identify clearly as dependant on on-site conditions. Requires pre-cleaning to allow access to cracks - depends on type and extend of surface contamination. Penetrant application - spraying under pressure or brushing in. Removal of the excess of the penetrant - wiping or brushing off. Developer application - brushing on or spraying. Eye inspection – needs sufficient lighting level.
Sensor Motion Requirements for Automation	Relies upon experience of the tester. If automated, two stage activity - penetrant spraying nozzle to be moved at constant speed and be positioned perpendicularly and at constant, pre-calculated, or guided by sensors distance to the surface and post-application surface inspection carried out using video camera of motion requirements similar to visual inspection (5.2.1).
Advantages	(1) simplicity of all stages; (2) no sophisticated equipment required; (3) low cost.
Disadvantages	(1) three stage process involving variety of equipment and activities - difficult to automate; (2) inspection process relies wholly upon experience and high observing skills of the tester, combined with the visual acuity and intelligence; (3) requires particular conditions for observation - suitable type of lighting, weather or enclosure dependant; (4) if recorded on tape, special type of film required; (5) surface cleanliness, in order not to restrict the capillary action of the developer.

Additional aspect of surface inspection using penetrants is the post-inspection clean up, which becomes be the additional stage in the automation task. Due to the nature of surface cleaning, the problems for automating these activities are the add-on.

5.2.3. Radiographic Testing (RT)

There are three basic methods available for this process: (i) X-radiography:- electromagnetic radiation of very short wavelength, emitted by electrons, whose velocity is suddenly reduced, (ii) Gamma radiography:- electromagnetic radiation of very short wavelength (shorter than X - rays) emitted by the nuclei of decaying radioactive substances, and (iii) Neutron radiography - neutrons (uncharged atomic particles, similar in mass to protons) when beamed through a component become differently attenuated and may be used to produce radiographs. These types of radiation are capable of acting on photographic plates and ionising gases. Rays can be reflected, refracted, and attenuated, as well as produce fluorescence or secondary beta radiation. Energy or penetrating ability increases with voltage (only for X-rays). Access to opposite sides of the tested element is required.

The principle of this method is the fact that X-rays or Gamma rays passing through an object (up to 25 mm thick) are absorbed differently by flaws or discontinuities. Cracks, voids and inclusions can be viewed as shadows imaged on film. The method is used to detect and locate of subsurface discontinuities within the material (cracks, porosity, voids, separation) and dimensional variations. The principle of radiographic testing is illustrated in Figure 5.1 and automation suitability is assembled in Table 5.4.

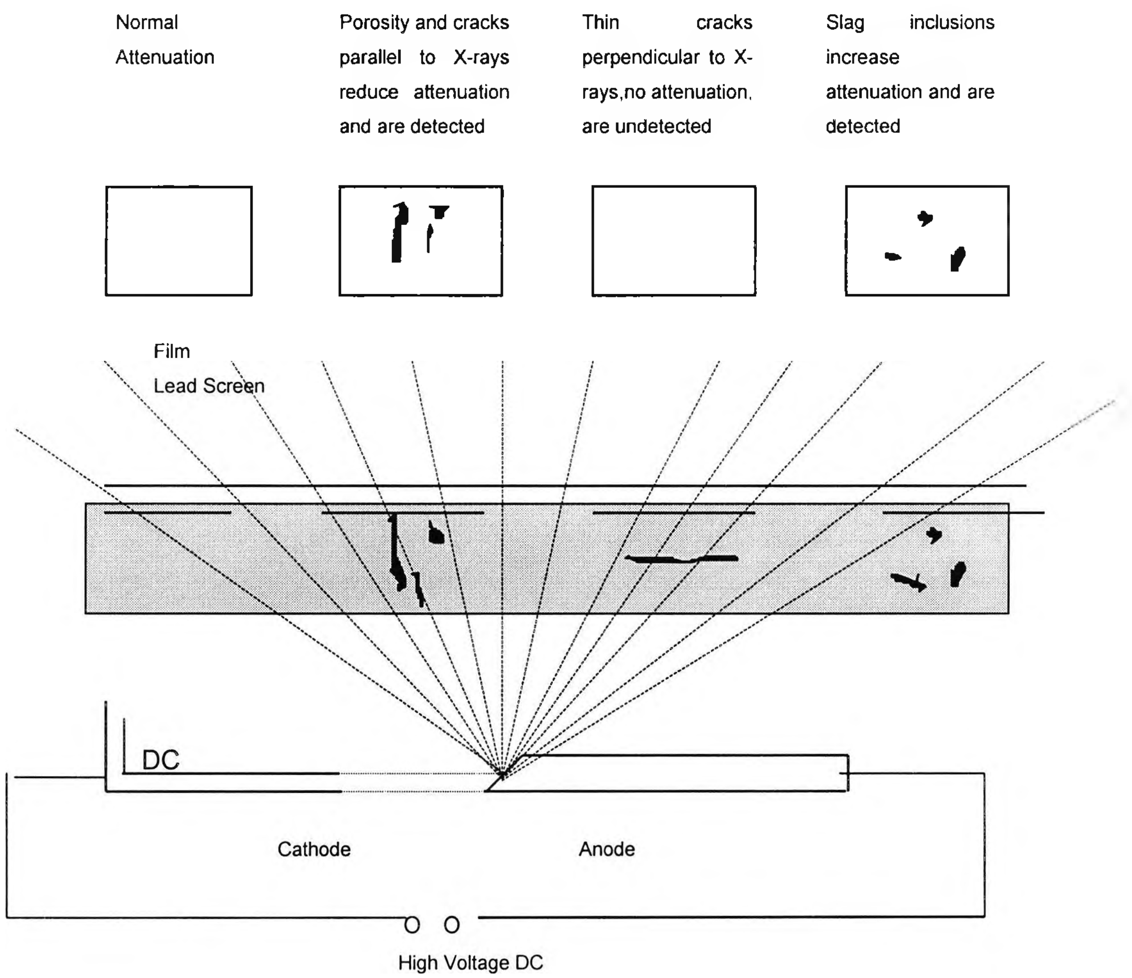


Figure 5.1 X-Radiography Testing (RT)

For a given initial intensity of radiation, the emergent intensity depends upon the thickness and the absorption coefficient of the intervening material, for example at 200 kV, 2.54 mm of lead absorbs as much radiation as 30.48 mm of steel.

Table 5.4 Automation Characteristics and Suitability of Radiographic Testing (RT)

Main Application	Weld inspection, imperfections within the thickness of materials, such as porosity, transverse cracks, voids, cavities, etc. Detectable defect size: min. 1.4 mm deep and approx. 2.5 mm long in 15mm to 25mm thick steel sections.
Description and Specification	Sources of radiation should be as follows: IR (Iridium)-192 Iridium for thickness 25-250 mm, Cobalt 60 for thickness 125-500 mm, Linac-8 MeV X-rays for thickness 500-1600 mm, emitted in 10E-9 to 10E-13 m range. The sensing method is of photo-emulsion, phosphor screen and conversion to video, while through photochemical processing and permanent film imaging the output becomes processed and recorded. The film should generally be of the medium speed or fast direct type X-ray for use with or without lead screens. The film and the intensifying screens should be enclosed in the flat, metal or plastic cassette with sufficient compression ensuring sufficient film-screen contact. Processing and recording method - photochemical processing and permanent film imaging. Interpretation basis - direct interpretation (standard penetro-meters for quality indication), control of contrast, density and resolution critical. <u>Weight</u> – Gamma radiography – 6 KGs, X-ray equipment -100KGs. <u>Cost</u> – Gamma radiography approx. £3,000, X-radiography field equipment above £7,000.
Sensor Motion Requirements for Automation	The X- and Gamma rays source has to be placed close to the monitored surface 200mm - 400 mm (or short projection distance 100mm - 150mm) and set at a known angle to the surface's normal.
Advantages	(1) portability of the equipment and its purposeful design for field use (Gamma radiography only); (2) film radiography yields a permanent record of results and is compatible to computer analysing techniques; (3) large areas can be inspected at one time.

Disadvantages	(1) access to opposite sides is required and monitoring of scattered radiation is necessary; (2) density or thickness variations of 1% to 2% only can be sensed; (3) in X- radiography voltage, exposure time and focal spot size are critical; (4) Gamma radiography requires special mechanisms for storage and extension of source; (5) sensitivity decreases with material thickness (Gamma rays being less sensitive to material thickness, comparing to X-rays and Neutron rays); (6) extensive expertise is needed to implement tests and interpret results; (7) Gamma source is uncontrollable and decays in time, so testing duration has to monitored; (8) cracks must be parallel to beam; (9) source and film geometry and alignment are critical; (10) high risk activity, extensive personnel training needed; (11) need to clear the whole site, while operating the equipment; (12) most items of equipment are produced as stand-alone, with no inter-faces provided to other equipment; (13) equipment mostly very bulky and heavy.

5.2.3.1 Computer Tomography (CT)

Computer tomography, like radiography, uses X-ray and gamma radiation to image the interior defects of an object. The CT image is reconstructed by computer from the penetrating radiation exiting the opposite side as perceived and measured by a detector-array. The image reconstruction process entails combining the data from numerous source/detector positions. Although this method is one of the most advanced NDT techniques, it still has some shortcomings, both positive and negative characteristics are assembled in Table 5.5.

Table 5.5 Advantages and Disadvantages of Computer Tomography in Automation

Advantages	(1) automated analysis possible; (2) great potential for automation in bridge inspection; (3) most sensitive and quantitative of all the advanced NDT methods.
Disadvantages	(1) detector and source must rotate around specimen at numerous scanning angles, (2) direct interpretation depending on the operator's skill; (3) access to all sides of the component required; (4) radiation hazard; (5) equipment custom built (cost approx. £ 300,000+)

5.2.4 Ultrasonic Testing (UT)

This testing method relies on high frequency sound waves being introduced into the material and the fact that ultrasonic pulses are not transmitted through large air voids. A pulse generator is used to generate an electric wave, which is amplified and converted to mechanical vibrations by a piezo-electric crystal probe and transmitted and reflected through the material under test. The reflected signal is then picked up by the probe, converted back to an electric wave and registered as an echo [Dawson, et all. 1990].

If a void lies directly in the pulse path, the instrument indicates the time taken by the pulse to circumvent the void by the quickest route. It is thus possible to detect large voids when a grid of pulse velocity measurements is made over a region in which voids are located, see Table 5.6 for full automation applicability, advantages and disadvantages..

Additionally, it is advantageous to be able to observe the received signal waveform and to measure the attenuation of the leading edge or envelope. Discontinuities interrupt the sound beam and reflect the energy back to a detector. The original and reflected echo signals are digitally compared on a time-lapse basis (Figure 5.3).

Where a velocity-indicating tester is used with attenuation measuring unit (C.R.O), the transit time and attenuation measurements can be made. Detection and location of discontinuities of the order of 1 mm in steel are feasible.

Table 5.6 Automation Characteristics and Suitability of Ultrasonic Testing (UT)

Main Application	Flaw detection in welds, plates, castings, mechanically joined splices and connections, crack sites. Detection and location of discontinuities, mainly cracks, thickness measurement of steel, detection and location of porosity, voids, non-metallic inclusions, corrosion. Thickness measurements with access from one side only, with 2% accuracy and for thickness 1-200 mm. Detectable defect size: min. 1.3 mm deep and approx. 2.5 mm long.
Description and Specification	Standard ultrasonic digital indicating systems comprise: exponential probe transmitting (high frequency pulse generator 2 to 6 MHz) and receiving (amplifier) transducers (ExTx and ExRx), transit time measuring device with processing and a recording unit. Weight - (i) transducers (Rx or Tx) -1.5 KGs, (ii) indicating tester - 3.5 KGs. Additional C.R.O. attenuation unit contains: pulse generator, input attenuators and buffer amplifiers. Weight: 2.8 KGs. Cost (approx.) - £4,000 for simple scan models, £10,000 and up for multi-scan computerised models.
Sensor Motion Requirements for Automation	Transducer perpendicular to the surface, at close proximity, use of light oil as good coupling medium.

Advantages (1) highly portable, lightweight units, tests can be performed quickly; (2) low expertise needed to take measurements; (3) ability to test from one surface only; (4) comparative accuracy in determining defect's size and depth.

Disadvantages (1) surface must be clean, smooth, free of rust and excessive paint; (2) probe alignment and coupling are critical; (3) high expertise is needed for interpretation of signal data; (4) small or thin parts are difficult to examine; (5) requires point by point search, hence expensive on large structure.

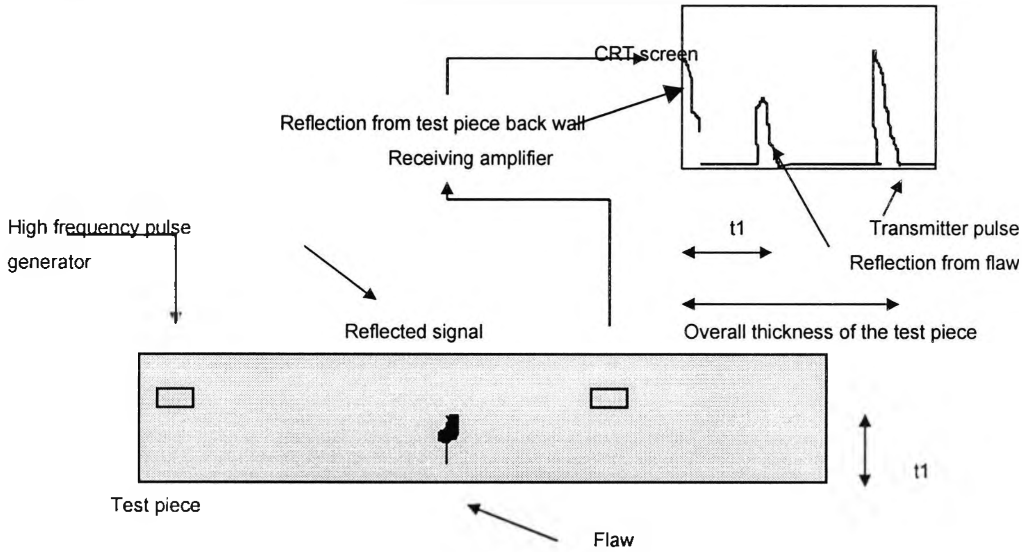


Figure 5.2 Ultrasonic Testing (UT)

5.2.4.1 Calibration

Because the velocity of sound in a material and the response of many of the instruments vary with temperature, frequent calibration is necessary when working under conditions of fluctuating temperature. Before an ultrasonic test is applied, the test sensitivity is determined and the instrument adjusted to allow discrimination between significant and insignificant indications. Therefore, both axes of display need to be calibrated.

The x-axis, i.e. transit time is comparatively easy, as it is usually calibrated in terms of material distance. To calibrate the y-axis effectively is more difficult because the object of the test is to detect defects greater than a specified size. This is literally impossible, as there is no absolute relationship between defect size and signal amplitude. The signal amplitude depends on the ability of the defect to reflect ultrasound back to the transducer. This depends on the defect size, shape, depth, orientation, specific acoustic impedance and surface texture. As it is possible to define the maximum permissible defect in terms of its actual size, it is necessary to use some other criteria, which can be defined and measured. A series of standard reference blocks has been developed for longitudinal wave testing.

5.2.4.2 Recent Developments in Ultrasonic Testing

Ultrasonic techniques can be used to detect both hidden and surface defects although they are less well suited to surface flaws. The minimum size of cracks that can be detected under ideal field conditions is estimated to be 0.05 in. deep and about 0. 1 in. long. Lately there has been the introduction of microprocessors to improve the quality of the ultrasonic technique. Gallagher and Trainor [1991] developed one of the early micro-processing techniques to develop the inherent subjectivity of ultrasonics. Their aims are twofold to make the achievement of accuracy easier, and to provide objective evidence of the test. They identified the key elements of the test where errors reduced accuracy:

i. Calibration - ranges, sensitivity and probe characteristics;
ii. Plotting - measurement of echo range and amplitude.

Then set about the development of a microprocessor-based system which would have the features that are felt desirable, such as:

iii. Portable and suitable for site use;
iv. Interfaces with standard analogue equipment;
v. Menu prompted calibration and plotting routines;
vi. Probe evaluation capability;
vii. Automatic timing and date stamping of test.

The result of their efforts is that of a microprocessor assisted manual system known as 'Mapel Microscan'. The heart of the system is the digitising signal processor (DSR) which is a small portable unit that interfaces with the technician's analogue flaw detector. It contains the A/D circuitry and a 'Husky Hawk' computer for the processing and storage of ultrasonic echo range, amplitude and probe position data. The Husky LCD screen permits menu-prompted calibration and defect plotting routines as well as displaying text input. Probe positional data is gathered by means of a flexible linear potentiometer, which can be used on curved as well as flat scanning surfaces.

The main purpose of all the new development is enhancing the reliability of NDT. The major advantages of ultrasonic testing, as explained earlier in the chapter, are in summary portability, sensitivity and the ability to detect the location of cracks and defects with depths. However, the major disadvantage does not rely upon the machinery itself but the operative, where results are strongly influenced by their skills. Therefore, training, experience and certification are key factors within this area. Other negative features are that the signal amplitude is not directly proportional to the size of the flaw and that the sensitivity can sometimes be too high so that too much information, such as grain boundaries and very minor defects not observable by other methods can cloud the overall picture.

5.2.5 Electromagnetic Methods

These methods are based on the principle that, when electromagnetic waves are transmitted through solids, reflection and refraction occur at interfaces between different materials [Bungey and Millard, 1993]. The interpretation of results can be hampered by uncertainties, such as electrical and material properties [Forde, 1993].

5.2.5.1 Magnetic Flux Leakage (Magnetic Field)

Surface and near surface discontinuities in ferromagnetic materials create a leakage field or perturbation in an induced magnetic field. Discontinuities interrupt the sound beam and reflect energy back to a detector, as seen in Figure 5.4.

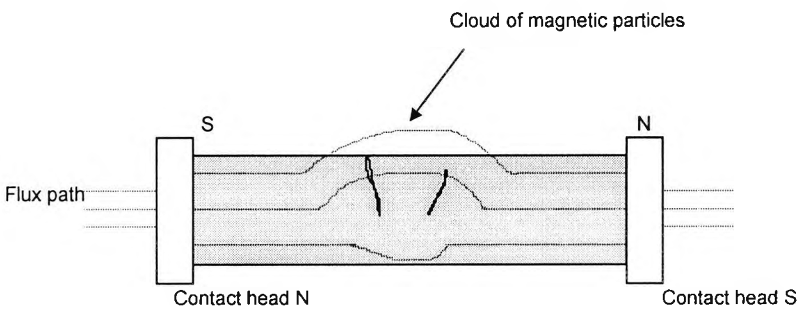


Figure 5.3 Test Piece in Magnetic Flow Detection

The electromagnetic tester consists of a typical probe (search head plus meter) surrounded by three ultrasonic distance measuring transducers, DMTs and supporting circuitry. An electromagnetic field is generated by the search head, which may consist of a single or multiple coil system. The physical principle involved utilises magnetic induction effects. The usual practice consists of magnetising the test object and then applying finely divided particles of magnetic iron oxide (Fe_3O_4) or iron filings, which are attracted to the surface at the points where cracks or other flows cause field leakage. Table 5.7 displays automation characteristics and suitability of this method.

Table 5.7 Automation Characteristics and Suitability of 'Magnetic Field' Method

Main Application	Detection of cracks at or very near the surface, checking pipes for cracks. Components subject to high stresses or fatigue, and those cast, welded and heat treated during fabrication. Defects up to 1.5 mm below surface, in welds up to 12.5 mm below surface.
Description and Specification	With probes using magnetic induction, a multi-coil search head is used with a low operating frequency, typically below 90 Hz. The principle is similar to that of a transformer, in that one or two coils carry the driving current while one or two further coils pick up the voltage transferred via the magnetic circuit formed by the search head and the steel imperfection. Such instruments are less sensitive to non-magnetic materials than those using the eddy current principle (to be explained below). The meter should incorporate scales or digital display ranges. The meter is adjusted so that the needle on the indicator dial (analogue devices) corresponds to the appropriate calibration mark as indicated by the manufacturer ('zeroing' the instrument). The search head is then scanned over the surface. Weight of a probe (approx.) - 2 - 3.7 KGs. Cost - portable units for steel start at £2,500.
Sensor Motion Requirements for Automation	The spot probe should be parallel and in direct contact with the surface or with close proximity (depending on the probe). In case of probes relying on the contact between the probe and the surface over certain area, such contact should be assured over the whole testing area.
Advantages	(1) rapid testing; (2) easily used for automation; (3) little or no surface preparation (only cleaning from grease and oil is needed); (4) cheap and robust probes.
Disadvantages	(1) probe must be near the surface and access to the element is required; (2) probe size affects sensitivity; (3) high sensitivity of the probe may cloud the interpretation; (4) every component must be tested at least twice, to ensure that flux travels in two directions at right angles and so crosses the path of longitudinal and transverse defects; (5) diagonal defects are not always detected.

5.2.5.2 Eddy Currents

Eddy currents are induced in a specimen by a time-varying magnetic field, generated by an alternating current flowing in a coil (probe) and a defect is detected by a perturbation in an electrical field. The conductor's eddy currents, in turn, create impedance in the exciting coil. Alternatively, a separate coil may be used.

The impedance produced depends on the nature of the conductor, the exciting coil, magnitude and the frequency of the current, and the presence of the discontinuities in the conductor, the steel under inspection. Table 5.8 displays automation characteristics and suitability of this method.

Table 5.8 Automation Characteristics and Suitability of Eddy Currents

Main Application	Application is limited to sections of simple geometry, as complex geometry changes the impedance itself and masks the effects of defects. Detect discontinuities such as seams, laps, slivers, scabs, pits, cracks, voids, inclusions and cold shuts. Measure dimensions such as metallic coatings, plating, cladding, wall thickness, outside diameter of tubing, corrosion depth, and wear. Seams and cracks as shallow as 0.03 mm can be detected.
Description and Specification	Alternating currents in the search coil set up eddy currents in steel which in turn cause a change in the measured impedance of the search coil. Instruments working on this principle operate at the frequencies 0.001kHz to 10 kHz (in steel) and are thus sensitive to the presence of any conducting metal in the vicinity of the search head and special calibration may be needed. After initialising, the search head is scanned over the surface. Equipment may include a detector, phase discriminator, filter circuits, modulation circuits, magnetic saturation devices, recorders, and signalling devices. Equipment variations exist, for different applications, such as: (i) equipment using impedance plane analysis, (ii) equipment using a single coil to scan the surface, or (iii) equipment using differential test coils. Weight of a probe (approx.) - 2 - 3.7 KGs. Cost - portable units for steel start at £2,500.
Sensor Motion Requirements for Automation	Spot probe should be parallel and in direct contact with the surface or with close proximity (depending on a probe). In case of probes relying on the contact between the probe and the surface over certain area, such contact should be assured over the whole possible area.
Advantages	(1) rapid testing; (2) easily used for automation for regularly shaped parts; (3) little or no surface preparation; (4) hidden defects and their sizes can be estimated; (5) no contact between coil and material may be required; (6) special coils can easily be made (versatility); (7) no special operator skills required; (8) low cost; (9) permanent record capability.
Disadvantages	(1) probe must be near the surface, at permanent distance and access to the element is required; (2) probe size affects sensitivity; (3) high sensitivity of the probe may cloud the interpretation; (4) applicable to simple geometry sections of the tested structure; (5) sensitivity for detecting irregularities decreases with depth; (6) shallow depth of penetration; (7) reference standards are required and difficult to make; (8) edges, speed, temperature, magnetic history of the part and surface conditions (smoothness) affect the test; (9) test results are comparative and not quantitative.

5.2.6 Acoustic Emission (AE)

When cracks propagate, they emit minute amounts of elastic energy that propagate outward from the source in the form of an acoustic wave. Sensors placed on the surface of the specimen detect and measure these waves and provide the information as to the location rate and of crack growth.

The AE technique involves monitoring the response of a component in the acoustic frequency range to transient elastic waves generated by rapid release of energy from a localised source in the component [Strainstall, 1993]. Consequently, the principle involved is that the component is subjected to a stress of short duration and the acoustic response is recorded. The main application of the technique is used structural steelworks and for detecting corrosion in the cables of suspension and cable-stayed bridges. The principle is illustrated in the Figure 5.5.

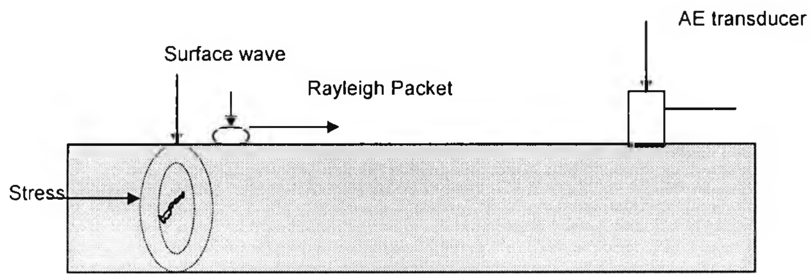


Figure 5.4 Acoustic Emission Testing (AE)

When examining cables for corrosion, a satisfactory way of applying stress to a cable is to use an air hammer striking a metal plate held between the hammer and the cable (to protect the strands). However, there is an alternative approach, which can carry out long-term testing by installing several sensors on a cable and measuring the response under traffic loading over a period of several months. The response is recorded as an acoustic emission count. Numerous applications of acoustic emission to the detection of cracks, but more particularly to the measurement of crack growth have been developed, and have led to significant improvements in the application of acoustic emission technology. This newer method, as used by Strainstall [1993], has more depth than other methods in that it is well suited to detecting defects not readily identified by other methods, such as crack growth, corrosion, cracks in fasteners, hinge pins, or eye bars. Table 5.9 displays automation characteristics and suitability of this method.

Table 5.9 Automation Characteristics and Suitability of Acoustic Emission (AE)

Main Application	Detection and location of incipient and active cracks in stressed structures, crack propagation, locating the tip of known cracks, remote, long-term and underwater monitoring. Example components include welded and riveted connections if proper filtering of fretting noises can be achieved, welded connections, sockets, web and flanges, pin and hanger assemblies.
Description and Specification	With this method energy released at deformation or crack sites is sensed by the piezoelectric transducer. This information is processed and recorded by digital counters, computer filtering, magnetic storage, graphic tape recorder and meter indication. Interpretation of the results is carried out on the basis of comparative or differential analysis of emission count rate, amplitude and frequency spectrum and differences in signal arrival times. Weight - 4.2 KGs. Cost - between £ 23,000 for basic four channel, up to £ 46,000 for multi-channel computerised models.
Sensor Motion Requirements for Automation	Contact between the probe and the surface should be assured throughout.
Advantages	(1) highly portable; inexpensive transducers permanently attached to bridge structures offer the potential for long term and remote monitoring; (2) monitors response to applied loads; (3) capable of locating the source of failure; (4) internal source of elastic waves; (5) cracks less than 0.003 mm in length can be detected.
Disadvantages	(1) acoustic coupling requires clean smooth flat surface and the removal of thick coatings; (2) transducer arrangement is critical to the results; (3) 'noise filtering' waveguides are required in high noise areas; and (4) extensive expertise is required to plan test and interpret results; (5) lack of possibilities to intensify the elastic wave field; (6) measurements cannot be repeated; (7) signals transient and random in time (standard noise reduction methods cannot be used); (8) several simultaneous measurements required for verification and orientation and determination.

5.2.7 Non-established Methods and Techniques

The advanced methods and techniques currently being developed, include devices that can detect decay even before visible signs on its surface appear [Tensiodyne, 1993].

Some NDT methods being developed, include (i) high-energy radiography for internal voids detection and (ii) Gamma-scintillation for detection of voids in structures.

These advanced methods give an idea of the future of NDT in today's ever-increasing sophisticated technology. Other advanced methods are examined in detail below.

5.2.7.1 Electrochemical Potential and Resistance Measuring (EPR)

The electro-potential is a measurement of the degree of equilibrium in the electrochemical process, which comprises the process of corrosion. Preventative surface treatment nearly always ensures that the rate of corrosion is kept at an acceptable level. This method provides a safer indication of hidden damage, so if used to produce a repair work schedule, the amount of damage occurring after renovation is less which means that substantial reductions to future maintenance costs are made. All these characteristics are assembled in Table 5.10 from the perspective of their advantage and disadvantage in automation.

Table 5.10 Assessment of Electrochemical Potential and Resistance Measuring (EPR)

Advantages	(1) potential depends on the rate of corrosion and can be recorded at an early stage; (2) the influences of corrosion can be located before any visible signs occur on the surface.
Disadvantages	(1) sensitive measurement; (2) requires highly qualified staff to interpret; (3) equipment available up-to-date still imperfect if used in site conditions.

5.2.7.2 Infra-red Thermography

Also known as thermal imaging [Titman, 1989], this method shows the surface temperature of an object and, most importantly, difference in temperature between areas. A scanner that consists of an infra-red (IR) detector, which is pre-cooled to an appropriate reference temperature, is used to measure surface temperature. By measuring the rates of cooling, material's structure and composition can be assessed [Robery, 1990].

Liquid nitrogen is used as a coolant but could only provide 'Polaroid' stills of the target. Recent developments include thermoelectric cooling without the use of a coolant, which offers greater flexibility and size reduction of equipment. Video recorders and computer enhancement of the images can also be used with the equipment. All these characteristics are classified and assembled in Table 5.11, below.

Table 5.11 Suitability Assessment of Infra-red Thermography

Advantages	(1) can be used where the delaminated portion heats up or cools down faster than the surrounding areas in response to air temperature changes, solar gain or a controllable heating/cooling source; (2) thermal conductivity differences due to variations in material quality and quantity which result in different heating and cooling rates; (3) scanning system gives real time pictorial display, with resolution of less than 1°C between differences in density; (4) tests can be carried out from a distance to the surface.
Disadvantages	(1) only measures surface temperatures, (2) lack of control over the heat supply versus cooling characteristics, (3) expensive (£ 20,000+).

5.2.8 Automation Potential Assessment of NDT in Inspection

The automation potential of the inspection techniques is assessed according to the following criteria: (i) size, weight and manoeuvrability, (ii) type of application (mode of operation), (iii) type of power supply, (iv) type of data collection, (v) level of operational accuracy required, (vi) overall suitability for automation and (vii) extent of the human supervision and involvement required to perform a test or interpret the results [Chamberlain, 1992]. The choice of assessment criteria results from robot specification, such as payload, type of motion, type of motors, auxiliary equipment requirements, type of data processing, number and type of DOF, size, reach, for example.

The probe specification is the vital element of the correct work definition for the robot. The size and weight of the probe together with operational requirements determine the dextrous envelope as well as the sequence of end effector's positions (tracks), to form a robot program.

The NDT techniques require: (i) a source that projects/ injects electromagnetic or mechanical waves, impulses or agents; (ii) a sensor capable of detecting and measuring the energy after passing through the material; (iii) an electronic means for processing data and (iv) display apparatus by which to observe and/or record data. The sensing requirements take into account normality (perpendicular orientation against tested surface), proximity, requirements for specific motion and contact controlled by force limitation.

The above observations and the information contained in Tables 5.2-5.10 are assembled into a brief comparative Table 5.12, below.

Table 5.12. Automation Potential of Surveyed NDT Methods and Techniques

Factors / Method (1)	Size Weight, Manoeuvrability (2)	Application (operation) (3)	Power support (4)	Type of data collection (5)	Operational accuracy required (6)	Suitability for automation 0-1 (high) (7)	Human involvement required 0-1 (high) (8)
Visual Methods - Camera	0.004-0.005 m^3 (Med); 2-6 KGs;Med	Static or slow motion	40-50 Watts	Video recording	Constant offset	0.6	0.4
Liquid Penetrant Tests	N/A*; N/A*; Low	Multi-stage, complex	40-50 Watts	Video recording	Pre-cleaning important	0.1	0.9
Ultrasonic Testing	Compact; 4-5 KGs +/- 3KGs; Med	Easy to perform, Probe position critical	Microprocessor with A/D board	Analog (direct)	Probe alignment & coupling critical	0.7	0.5
Radiographic Testing	0.05-0.06 m^3 (Small); γ-ray-36KGs; X-ray-100KGs; Low	Simple, health hazard	115/230V AC or 12V DC	Analog (direct)	Constant min. distance & normal to surface	0.7	0.3
Electromagnetic - Impulse Radar	0.07-0.09 m^3 (Small); 25-30 KGs; Low	Simple raw data collection, complex to interpret	110/240 V or Special support unit	Analog to special unit	Contact with surface & speed control	0.6	0.5
Electromagnetic - Magnetic Flux	Med; 2-4 KGs; Med	Simple to perform	Special support unit	Analog (direct)	Contact force & normal to surface	0.5	0.5
Electromagnetic - Eddy Current	Med; 2-4 KGs; Med	Simple to perform	Special support unit	Analog (direct)	Contact force & normal to surface	0.5	0.5
Acoustic Emission	0.001-0.002 m^3 (Small); 2-4 KGs; High	Mainly static Easy to perform difficult to interpret	25V-30V	Microprocessor with A/D board (direct)	Only permanent contact with surface	0.7	0.3
Electrochemical Potential & Resistance Measuring	Large 2-4 KGs Low	Complex	Small	Microprocessor with A/D board (direct)	Contact with surface	0.7	0.4
Infra-red Thermograp	Med Lt/wt	Multi-stage, difficult to	Small	Microprocessor with	Contact with the surface not	0.5	0.5

hy	Low	interpret		A/D board (direct)	necessary		

N/A* – due to the nature of the test the assessment of size and weight is not applicable.

In Table 5.12, the values in column (7) and (8) have to add up to 1.0 and are reached by the author based on the benchmarking and relative comparison of the methods. At one end of the spectrum there is a method which could be fully automated, with no manual operations at all, and at the other end, a method so multi-stage and intricate in application that no automation is possible (liquid penetrant method).

Table 5.1 presents very detailed suitability of all the NDT methods to variety of defects, Table 5.13 (below) provides a shortened version, enhanced by conclusions from the detailed analysis of the methods, carried out in the body of this chapter..

Table 5.13. Applicability and Versatility of NDT Methods [Birchon, 1991]

Method / Defect (1)	Visual Methods – Camera (2)	Liquid Penetrant Tests (3)	Ultrasonic Testing (4)	Radiographic Testing (5)	Electromagnetic - Magnetic Flux (6)	Electromagnetic - Eddy Current (7)	Acoustic Emission (8)	Electrochemical Potential and Resistance Measuring (9)	Infra-red Thermography (10)
Surface cracks and tears	2	2			3	2			
Subsurface - not lower than 3mm - blowholes, inclusions			2	1 x-ray		3		2	
Internal - blowholes, shrinkage cavities, gas holes, inclusions			2 sections over 20mm	3 x-ray 2 γ-ray sections over 20mm					2
Internal - shrinkage, porosity, open grain							2		2

Suitability 0-3 (best)

5.2.8.1 Reliability of NDT

The causes of unreliability and hence greater costs in NDT are many and can be related to a number of possible sources [Gallagher and Trainor, 1991], such as:

(i) The definition of defects - In any inspection a dividing line is set up and features which exceeds this limit are classed as defects while other features are ignored. The placement of this dividing line is based on a consideration of the consequence of the flaw remaining undetected. Setting the limit low will result in a higher apparent defect detection rate but at the same time more of the defects located will not in fact lead to failure. Therefore, the dividing line should be based upon the magnitude of the flaw itself - in the common case of crack detection it would be the crack height. Therefore, it will depend upon the type of flaw being studied as well. Once the level of discrimination becomes comparable to the current capability of NDT there will be a rapid fall in reliability. However, this does not take into consideration the economic aspects of inspection and the rapidly rising costs that would normally be a sufficient brake on the selection of unrealistic inspection standards. Finally, it is important to balance the discrimination level with both the performances of NDT and a realistic failure assessment.

(ii) <u>The physics of the inspection process</u> - The performance of NDT is based on one or more of a number of physical phenomena and the nature of these phenomena determine the fundamental capability of the technique employed. Having discussed the relative merits and limitations of each of the methods, concluding that there are wide variations between the approaches, to ignore the limitations is to risk using an inspection for which the success rate is small, although there may well still be many spurious indications. When concerned with cracks, ultrasonics is likely to perform well and a rapidly rising detection probability is expected. Radiography will also find such cracks, but the need to align the crack and the beam to achieve certain detection means that, for a straightforward radiographic inspection, the detection probability will rise slowly. In this situation eddy currents will not be capable of performing the inspection. Therefore, many considerations have to be taken into account.

(iii) <u>The nature of the technique employed</u> - Having chosen a suitable approach, it is necessary to choose an appropriate inspection technique. Thus having chosen the ultrasonic approach, the most suitable ultrasonic technique must be selected for the task in hand, for example the 6dB drop technique is limited to sizing flaws which are larger than the ultrasonic beam width - typically more than 10 mm. If this technique is selected to discriminate at a size of only 3 mm the inspection is bound to be unreliable. Therefore, it is vital to have a reasonably accurate estimate of the precision of the chosen technique at and around the level of discrimination chosen.

(iv) <u>The inspection environment</u> -This may affect the technique in two ways the geometric factors and the working environment. Geometrically, for example, the complex node joints of oil rigs are clearly more difficult to inspect than a straightforward butt weld and, when techniques are considered the extra inspection complexity this introduces must be taken into account. Finally, there are aspects of safety. Many inspection environments are extremely hazardous, for example, Severn Bridge or some underwater inspection

(v) The type of equipment employed - This concerns the appropriateness of the equipment for the task in hand and the method of recording information. Equipment that is not easy to use will result in variability in calibration, which will feed back to reliability. Recently techniques that tend to provide a read-out which is pictorial and which can be recovered for later analysis away from the inspection environment are more favourable. The continued increase in computer power may eliminate this problem over a period of time provided the very difficult problem of the automatic interpretation of NDT data can be overcome.

(vi) The operator - In general, the operator will work well if his task is straightforward and if they have confidence in the technique. However, the situation is complicated by psychological factors, such as belief in the actual technique used or if he cannot realistically monitor his progress or coverage then the attitude of the operative may diminish and temptation for short cuts may arise. Overall human reliability is a major factor.

These factors and requirements seem obvious, but they are essential considerations in establishing reliability, and limitations in satisfying them can lead to unreliable inspection [Gregory, 1992].

5.3 Cost of NDT

The costs of testing are made up of several factors such as: (i) labour cost; (ii) costs of test materials, such as X-ray film, liquid penetrants and other supplies; (iii) operating costs, such as electricity, water and scaffolding; (iv) fixed costs, such as equipment, depreciation and insurance and (vi) externalities, such as disruption to the public from testing operations. These costs can vary by several hundred percent for any single non-destructive test. The most important cost influencing factors are:

i. Quantity of stages and activities tested: as in the cost of machining, the set-up time is important in non-destructive testing. Some tests require none, as with penetrant testing, however others, such as radiography, may require at least half an hour. A supplemental factor is the variety of sizes or shapes of parts and the total quantity of tests.

ii. Handling of the parts to and from the test unit or field: just as in any production operation, part handling may frequently cost more than the main operation itself. In considering the cost of non-destructive testing, it is important to consider methods leading to the tests, as field preparation, and subsequent methods, such as clean-up, testing equipment removal.

iii. Handling of the parts during the test process.

iv. Automation of the test: many tests demand operator involvement. However, the development of microprocessor-based testing systems has increased the effort to reduce operator reliance in almost all non-destructive testing procedures.

v. Sensitivity required of the test method: this refers to closer control and greater accuracy required for a particular operation, which may result in more training for the operator and more time for test results.

vi. Tolerances permitted in the interpretation of the test results: together with sensitivity this can be the most expensive part of the operation, because it may depend upon time, which will mean greater labour costs.

vii. Percentage of defective parts found by the test: the more defects the more time spent on the operation.

Consideration of each of these factors indicates some ways of reducing cost. However, the costs of such operations will depend partly on the reliability of the NDT system.

5.4 Restoration Methods and Equipment

Based on the inspection results, the condition of the structure is assessed, in terms of the percentage of the surface showing some sort of failure. For structures showing protective coating deterioration greater than 20-25%, it is usually more cost-effective to clean the entire structure.

Surface contaminants such as rust, rust scale, chemicals, salts, dirt, loose paint, dust oil, grease and moisture will cause poor bonding of a coating to the substrate. Good surface adhesion of the primer is essential and this can only be achieved if the substrate is clean and also has an anchor pattern (or profile) on the steel surface, produced by abrasive blast cleaning.

The established surface cleaning techniques can be grouped as follows: (i) sand blasting (olivine sands), (ii) abrasive grit blasting (aluminium oxide (corund), steel grit, coal slag grit, glass bead, staurolite or garnet sponge media), (iii) water blasting, (iv) ice blasting, (v) chemical cleaning and (vi) needle gun cleaning. Concern of the toxicity of the silica sand, used traditionally in dry abrasive blasting, and paint particles has caused the need for alternative surface preparation methods. These include wet sand or grit blasting, water blasting or vacuum blasting and containment of abrasives.

Classifying methods according to tools gives: (i) impact tools (needle scalers, power scabblers, bushing hammers, power chisels and pro-scalers), (ii) rotary tools (wire brushes, sanders and cutters) and (iii) rotary impact tools (differently powered rotary peening machines).

Also, as there is a limited number of equipment supplier companies, the variety of tools and systems are commonly associated with each company (Appendix C.1).

5.4.1 NDT Field Tools

Despite the relatively good defect detection capabilities shown by many NDT techniques, it is often difficult to obtain good performance, under field conditions [Manning, 1985]. For example, sensitivity may be sacrificed to achieve greater coverage, ease of operation or portability. Some of the potential accuracy of the equipment may be lost due to the limited operator skill, operator discomfort or the surface condition. To overcome some of these difficulties, combination systems have been developed, such as ACD system (by an American Research Group - Southwest Research Institute), for detecting fatigue cracks [Tensiodyne, 1993] utilising both, an acoustic crack detector (ACD) and a magnetic crack definer (MCD). The two-instrument system, intended for use of semi-skilled personnel, is designed to first detect (ACD) and then define (MCD) the length of the crack. Each unit consists of a hand-held probe and backpack.

The ACD consists of a 2.25 MHz, 70° wedge, ultrasonic probe calibrated to give a digital display of the distance from the probe to the defect. The instrument also incorporates features to measure the effectiveness of the surface coupling and to indicate a presence of the defect through earphones. The unit operates effectively at 1 to 3 m from the region to inspected, depending on the surface conditions. The expected sensitivity under those conditions is to be able to identify a crack 19 mm long or more. Once the crack is identified, the MCD is used to define its length.

The MCD consists of an iron core electromagnet operating on a 106 Hz alternating current. Two differential coil pickups orientated selectively with respect to the driving magnet detect disturbances in the magnetic field when a crack is present. The light on the probe and sound through the earphones signals the presence of the crack. By following the crack with the probe, the length of the crack can be mapped. The unit is designed to determine crack lengths to within 6 mm and operate on heavily scaled or old painted surfaces.

5.5 Assessment of Automation Potential

The suitability of the repair tools and techniques for automation reviewed above is assessed according to the following criteria: (i) size, weight and manoeuvrability, (ii) versatility to the variety of surfaces, (iii) type of power support, (iv) level of suitability for bridges, (v) level of operational accuracy [Chamberlain, 1992].

Table 5.14. Automation Potential of Commercially Available Surface Restoration Tools

Aspects / Hardware (1)	Size, weight, manoeuvrability (2)	Versatility to variety of structures (3)	Type of power support (4)	Level of suitability for bridges (5)	Operational accuracy required (6)
E10-10 APTDC Blaster [Nelco]	2 m^3 280KGs Limited	Downwards only 250mm wide blast pattern	Electric 10 HP	Limited	Low
AC7-4 Deck Blaster [Nelco]	0.5 m^3 66KGs Limited	Downwards only	Pneumatic 4 HP	Limited	Low constant force normal to surface
JHJ-2000 Hand Held Blaster [Nelco]	50mm wide strip 5KGs High	Perpendicular to any surface	Pneumatic or electric 1.2 HP	High	Low, constant force normal to surface
EV7 - Vertical Blaster [Nelco]	0.07 m^3 40KGs High	Vertical or overhead	Pneumatic-4 HP Electric - 2 HP	Med	Low constant force normal to surface
EV-15-30 Vertical Blaster [Nelco]	0.8m^3 625KGs High	Horizontal or vertical, large surfaces, 400mm wide blast pattern	Electric 30 HP	Med	Low
Vacuum Type 5310 [Trelawny]	0.3 m^3 8-20KGs limited	N/A	Electric	Low	N/A
Vacuum Type 7310 [Trelawny]	0.4 m^3 32-90KGs limited	N/A	Electric	Med	N/A
MCV Vacuum [Trelawny]	0.008 m^3 20-230KGs limited	N/A	Electric	Med	N/A
Needle Scalers [Trelawny]	35mm dia. x 450mm length 1.5-5.5 KGs High	Perpendicular to any surface	Electric	Low Localised repairs only	High, constant force downwards
PPT Peening Prep Tools [Trelawny]	100x80mm contact area + hose + vacuum 3-6KGs	Perpendicular to any surface	Electric	Low Localised repairs only	Med-high, constant force downwards, 25mm clear

	High				from edges
SF 11 Shrouded Floor Scaler [Trelawny]	200x150mm contact area 38KGs Med	Perpendicular and downwards to any surface	Electric	Low-med	Med-high, constant force downwards
SRA Shrouded Grinder [Trelawny]	125mm dia. disc 3.2KGs High	Perpendicular to any surface	Electric 1.2 HP	Low-med	High, constant force downwards
Heavy Duty Roto Peen [3M]	0.008 m^3 2KGs High	Perpendicular to any surface	Electric	Low	High, constant force downwards
Scotch-Brite Removal Discs [3M]	50-200mm dia. discs 1.5KGs High	Versatile	Electric	Med-low	Med, constant force downwards
DKV 2005 Vacuum/filter Container and Separator(hose) [De Kleijn B.V.]	Total: 2.8 m^3 160-460KGs Low	N/A	Air-powered	Med	N/A
DKO 3500 Dust Filtering [De Kleijn B.V.]	2.5m^3 120KGs Low	N/A	Air-powered	Med	N/A
Husky E-150 Pump (water) [Flow Ltd.]	3-3.5 m^3 2040KGs Low	N/A	Electric 150 HP	Med-high	N/A
Husky S-200 Pump (water) [Flow Ltd.]	2-2.5 m^3 2770KGs Low	N/A	Pneumatic 205 HP	Med-high	N/A
A-3000 Hand Tool (to spray high-pressure water), [Flow Ltd.]	400-1300mm long 5KGs High	Versatile	Pneumatic 0.7 HP	Med-high, cleaning path 70mm, localised repairs	Med, normal to suface
Jetlance 5062 Hand Tool (to spray high-pressure water)	0.05 m^3 11KGs High	Versatile	Electric	Med-high, shielded	Med, normal to surface
Jetlance 5060 Hand Tool (to spray high-pressure water) [Flow Ltd.]	1200 mm long x 300mm shield dia. 8.4KGs High	Versatile	Electric	Med-high, shielded	Med, normal to surface

5.6 Critical Review of Commercially Available Robotics Technology in Inspection and Restoration

The author felt that compiling all commercially available systems (in the industry as well as in research laboratories) for inspection and/or restoration to be used for variety of structures and environments will summarise the previous sub-sections (Table 5.15).

Critical review of the automated systems is carried out according to the following criteria: (i) versatility for variety of bridges, (ii) flexibility in accessing the whole of the structure, (iii) type and level of automation, (iv) compatibility between the mobility and process characteristics, (v) level of environmental protection (containment) and (vi) level of speciality (use of enabling technologies).

Table 5.15. Assessment of Commercially Available Automated and Partially-automated Restoration Systems

Aspect / Hardware (1)	Versatility for variety of bridges (2)	Flexibility in accessing the whole structure (3)	Type and level of automation (4)	Compatibility between the mobility and process characteristics (5)	Level of environmental protection (containment) (6)	Level of speciality - use of enabling technologies (7)
ALPHA 100	Low	Low	High	Low	Yes	High
The Aqua Blast 2500 Plus	Low	Low	Low	Med	No	Low
Auto-Blaster	Low	Low	High	Med	No	Med
Laser-Vac	Med	High	Low	High	No	Med
LTC Robotic Unit	Med	Med	High	Low	Yes	High
Nelco System	Low	Med	Low	Low	No	Med
Sandroid Systems Inc.	Med	Med	Med	High	No	High
Ship ARMS Depainting System	Med	Med	High	Med	Yes	Med
DKS 7116 Vacuum Blast Machine	Med	Low	Low	Low	Yes	Med
Blastman B10	Low	Low	Med	Low	Yes	Low
Blastman B20	Low	Low	Med	Low	Yes	Low
Blastman B20C	Med	Low	Med	Med	Yes	Med
Blastman BE20	Med	Med	High	Med	Yes	Med
Blastman BR20 (2 units)	High	Med	High	High	Yes	High
Robotic Bridge	Med	Med	High	Med	Yes	High

Paint Removal System by (RBPR)

5.7 Automation for Bridges

Task-planning procedures combine activities of various scales and at numerous levels. The global levels include identification of 'unit workpieces', meaning the approximate volume of the working envelope of a robot in one position, and as a consequence of this, an optimal sequential positioning of unit workpieces. The focused levels include the task and tool requirement analysis and positioning of the points within the paths, for the end effector to follow, in order to maximise the effectiveness of the workpiece and local geometry juxtaposition.

The Tables 4.1 and 4.2 do not serve generalisation for the complete grouping of various bridge types, as a uniform division into bridge types does not exist, depending significantly on the approach and differentiation criteria and merely provide some indication. However, several characteristics are apparent.

Before these can be further explored, a closer look at a small-scale level is necessary. The 'unit workstations' (workpiece that can be completed without relocating the robot) cannot be optimally positioned on the given bridge structure without knowing the parametric characteristics of the robot, which in turn cannot be fully identify without the geometry of standardised unit workpieces on the bridge. The whole process takes a form of goal seeking through an iterative process. When looking at existing automation technology using 5- or 6-DOF arms, the reach gives an approximate gross volume of the working envelope, which gives a basis for dividing the target bridge into cells of similar size and, therefore, all the details in Table 4.3 correspond to this assumption. A further observation is that, although bridges vary in types, sizes and structure, the basic cells at the small-scale level can remain similar and repetitive for variety of bridges. Therefore, the library of 'basic cells' can be eventually assembled on the basis of adding new items from analysing different structures with Table 4.3 serves as a starting point.

Then the investigation of restoration activity must follow. The task itself (for paint stripping, for example) involves leading the blasting nozzle along paths, oriented perpendicularly to the treated surface and at a certain constant distance from it. These paths, are derived from the distinctive task analysis and bridge geometry, and are mapped as rows of points to be reached consecutively by the end effector. As the paint stripping application requires continuous-path motion control, the spacing between the points on trajectories can be easily altered (increased number and located closer together for continuous-path motion effect). The whole bridge has to be divided into 'unit' workstations (workspace attended without moving the robot) which correspond to the volume enclosed by all the bridge surfaces, which can be stripped, without relocating the mobile trolley.

All the details in Figures 4.1 to 4.15 display such paths consisting of three points in the most effective locations for the task completion, within the cells. Positioning and spacing of the points within paths depends on the characteristics of relevant probes or tools. For NDT, they are covered in Table 5.2. In the case of blasting or spraying activity, a starting point in transition of nozzle movement into task planning is the analysis has been carried out by Rosenfeld et al. [1994]. Locating paths within cells in an optimal way is the function of local geometry and robot characteristics (DOF, kinematic and dynamic parameters). Different strategies are used to control the path taken by the end effector during movement between points on planned trajectory. Controlling the speed of the actuators and relative location of the path points can help 'smoothing' of the path. By using interpolation, a PTP control can also move an end-effector along a controlled path, the required incremental motions calculated from the co-ordinates of the start and. Continuous path (CP) control is required when the path traversed by the end effector is of the importance to the task. The CP control for the spraying task reproduces the path traced by the spraygun, at the teaching stage, when the gun is manually led through the spraying operation.

5.8 Summary

Bridges are classical examples of steel structures, whose design aspects, type of service and exposure make them particularly prone to dilapidation. The factors contributing to deterioration and typical areas affected by types of deterioration are many and varied. Most problems have to be identified by inspection and overcome by post-construction maintenance, therefore, it is vital that both processes are sufficient and cost effective, while safe for workers and the environment. The in-depth analysis of the NDT and restoration methods and techniques has been carried out in this chapter, in order to identify the methods most suitable for automation. Their analysed characteristics included the type of control, methods of application, physical data and type of power support applicable. Collection of this data, and including automation assessment, aimed at successful selection or design of the prospective robot, capable of carrying out the work.

The evidence of several attempts, as assembled in Table 5.15, to come up with the system to be employed in field restoration, supports the need within the industry.

However, as indicated in Table 5.15, and resulting from the detailed descriptions of all the systems in Appendix C, all the systems currently available have major handicaps in terms of versatility and on-site application.

In the following chapter, the existing enabling technologies that have the potential to become part of automated systems, are presented and appraised. The existence of the enabling technologies must not be ignored, as their appraisal and suitability assessment may identify suitable, commercially available components or sub-systems, which can be employed.

CHAPTER 6 - ENABLING TECHNOLOGIES

6.1 Introduction

In the previous chapter, the commercially available testing and inspecting equipment and techniques were overviewed. This chapter investigates the logistics (delivery to the task and the robotic arm carrying the tool). Therefore, the industrial access systems and the available robotic arms (utilised in the nuclear and sub-sea industries) are reviewed for possible application in the proposed device. The contribution to the thesis is the up-to-date assembly and assessment of the access systems and robotic arms with their in-depth characteristics.

It was established in the early stages of research that the proposed bridge automated inspection and restoration facility would have low automobility and rely on being delivered to the working environment. For this reason, research into commercially available access systems was undertaken, such as those offered by BEECHE and TRACTEL [Beeche, 1999] and [Tractel, 1999]. Of these systems the modular space frame platform from Beeche Systems Corporation showed highest versatility. This platform offers fast in-situ assembly for high load capacity accesses equipment, cantilevered access to the underside of bridges and creates large, rigid working platforms of safe expandable form for large civil structures. The frame is assembled fom the octogonal lightweight modules, which offer flexibility, high loading capacity, without overloading the structure due to selfweight.

Also investigated are the existing telerobotic manipulators with five and six degrees of freedom (Tables 6.3 and 6.4), used by heavy industries in hostile environments. For this reason, a second-generation, servo-hydraulic, six DOF arm is appraised for possible use. The application is characterised by the set of key criteria, vital to the high quality performance. The appraisal involves identifying their key kinematic parameters of the arm.

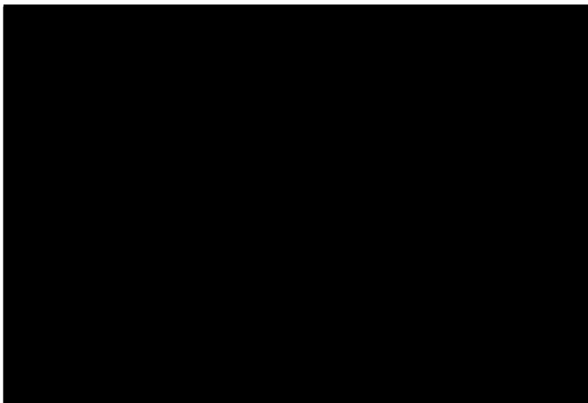
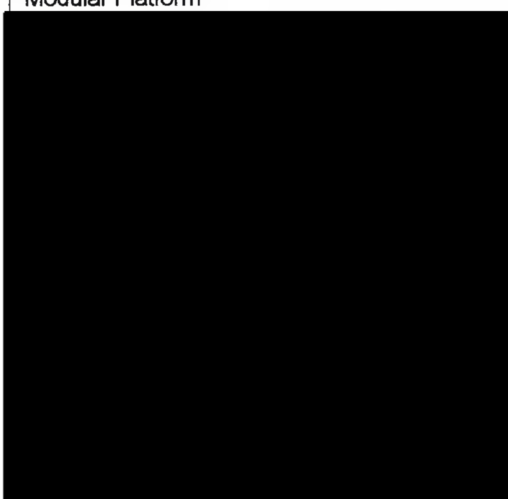
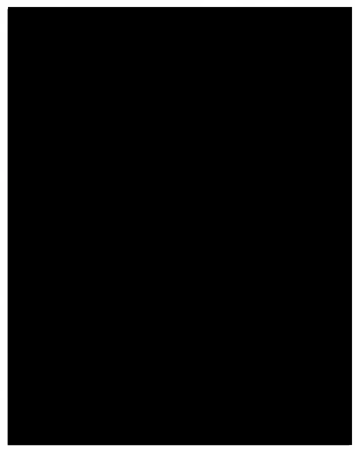
This chapter is the last within the part of the research dealing with information and data collection, assesment and review. The findings from this and previous chapters will serve as a basis for the development of the research model through assemblying the prototype, assessing criteria required for top performance and identifying parameters for optimisation.

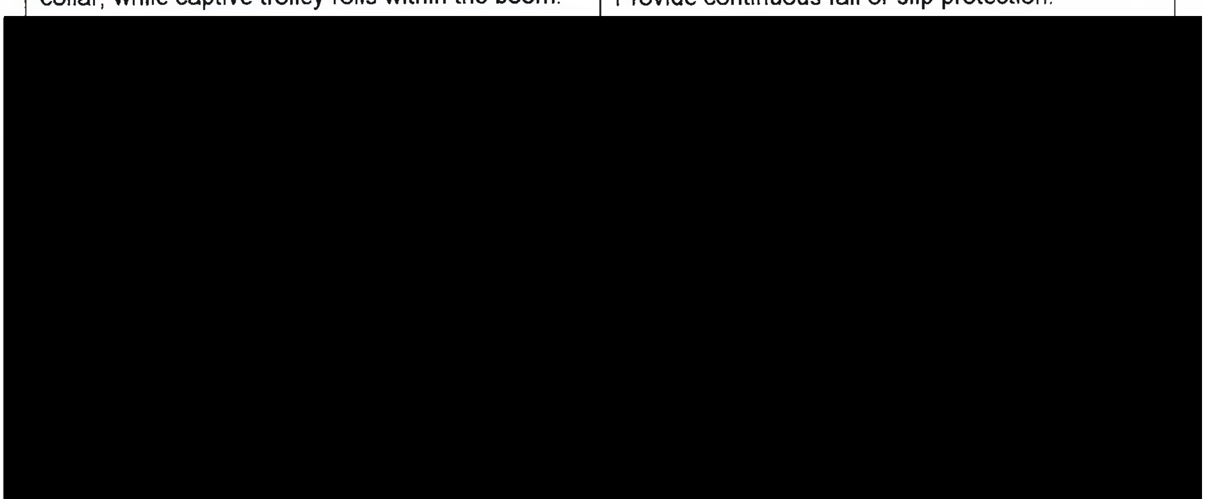
6.2 Primary Access Methods

Identifying automation task benchmarks is closely related to key factors such as access and mobility provisions. Possible solutions to primary access are now reviewed.

The variety of typical access equipment and lifting gear is limited and varies typically by size and payload. The ability to link these elements into systems applicable to variety of geometries and sizes provides the competitive advantage. Table 6.1 provides the graphical information on the basic products. The descriptions of the assembly and application scenarios follow the table.

Table 6.1 Access Equipment

 Modular Platform	 Outrigger
 Hoisting System Type I – Two piece davits can be designed to tilt to horizontal before detachment from the base. The arm rotates on a unique roller collar, while captive trolley rolls within the boom.	 Stop Type Safety Device – Provide continuous fall or slip protection.



Hoisting System Type II - With hoists on the traveling deck rig, the suspended platform has a greater carrying capacity. There is no need to supply power to the platform, eliminating voltage	Travelling Platform - The powered platform is fitted with state-of-the-art hoisting systems and code-required safety features. Folding platforms can run also horizontally or on an incline.

line drops.	
	
Hoisting System III – Winches and self-turning handles allow ease of installation and operation. Rolling davit systems are also available to overcome the labour intensive movement of davits to new locations.	Deck Cantilevered Platform – Deck-car systems are cost-effective access solutions requiring minimum equipment handling so operators can concentrate on maintenance.

Scenario 1:- To inspect pier sides, a hoist with a stop safety device can be mounted on the wall of the pier for lifting a compact telescopic platform. The option of using two guiding ropes stops the platform from rotating.

Scenario 2:- Inspection of underside and sides of the bridge decks can be carried out with a modular platform. For the suspension the outriggers are used on one side, on the other side a special structure is attached to the outriggers, which gave access to the platform and at the same time, allowed work on the side of the bridge. The platform is lifted into place by hydraulically. To pass the piers of the viaduct, one end of the platform is completely lowered until the platform is suspended vertically from one of the suspensions. Then the outriggers are moved and the platform lifted again

Scenario 3:- To inspect large size bridge piers two modular platforms can be used, both operated hydraulically by two independent machines. They are connected together and surround the piers of the bridge, enabling them to withstand the winds and form a temporary gantry. A separate intermediate platform is used as an elevator to bring the personnel and equipment to the working platform. Suspension brackets, can be fixed under the roadway of the bridge straight above the piers. They can be dismantled and moved to their new position whenever the platforms are moved to their new position on another pier. A gantry with a ladder, sited on the roadway and giving access to suspension brackets is also moved every time new pier is inspected.

Scenario 4:- To inspect the underside of multispan railway arch bridges, for example, a modular platform is suspended on the hydraulically operated outrigger system mounted on a platform wagon travelling on rails. When the extra long platform is required the other end of it is suspended by a trolley running on a suspended I-beam to the other side of the bridge.

Scenario 5:- To move a suspended platform under a bridge a system of two trolleys on I-beams is mounted on both sides of the bridge. The machines pull on the two wire rope which are fixed to each end of the bridge, and so move the platform.

Scenario 6:- Modular space frames with various 'unit shapes' which can be linked together in infinite configurations to fit the shape of bridge, tower or building being maintained. The aluminium truss frame comprises modules forming inverted pyramids, which are fastened together to make the multidimensional configurations of platforms. System is able to move vertically and longitudinally and with the frame and deck being designed in modular panels, travel is possible around piers, by disassembling and assembling sections, while suspended under the bridge. The trolley rails allowing the movement are attached to the bridge structure by mechanical means. All the power lifting motors and reelers to hold steel suspension ropes are housed underneath the deck. Accurate levelling of the platform is ensured electronically. It can be powered into position on the bridge towers at a rate of nine meters a minute using man-riding winches – Table 6.2.

As pressure increases for safe and environmental control of hazardous debris, modular space frame system is able to house the grid recovery funnels under the open panel grid decking. Containment helps to improve working environment by reducing the risk of abrasive blast medium, preventing corrosive and coating products from entering vital sustaining plant – Table 6.2.

Table 6.2 Assembled Platforms and Outriggers on Bridges

[redacted] Steel Truss Frame in the work position	[redacted] Steel Truss Frame in the traversing position
[redacted] Steel Truss Platform with Containment	[redacted] Steel Truss Platform with Containment
[redacted] Suspension Outriggers for Steel Truss Frame	[redacted] Steel Truss Frame Assembly

6.2.1 Add-on Concepts for Automation

Analysing these systems proves useful in utilising some of the solutions as supports aiding the automated device. Fixing permanent brackets or rails to sides of the deck and piers would facilitate suspension or support the main vehicle. Therefore, as a result of the products' structure and application, alternative systems' specification are put together as potential delivery assemblies:

Concept 1:- An inspection climbing robot is hung from two trolleys running on rails fixed permanently to the sides of the deck, which enables it to inspect sides and underside of the latter. A separate arrangement of clamps or railings along the piers with horizontal suspension of the robot from similar but vertical trolleys would enable pier inspections to be undertaken. At the same time the robot would be operated by pneumatic power with vacuum grippers, suction pads and pneumatic supply cable suspended from the trolleys. The system would move by means of vacuum sucker feet as grippers synchronised with the movement of the trolleys.

Concept 2:- The automatic platform is suspended by the clamps fixed at each junction between the underside of the deck and the tops of piers. The powered platform would ascend and descend along the piers from wire cables operated by means of two electric motors fixed at either end. If the underside of the deck needs inspection, the same platform would move itself along the wire cables fixed between the clamps at the ends of each inter-pier span using the same motors.

6.3 Existing Manipulators

For a manipulator and end effector to move to any position within its working envelope and adopt any orientation in that position, it must have at lest six independent degrees of freedom (DOF). There are, however, commercially available manipulators with less degrees of freedom, which offer savings in cost, complexity, weight with space restrictions. For many tasks, systems with fewer DOF may be sufficient and therefore, task analysis is of prime importance. Most NDT tasks are likely to require great dexterity and, therefore, 6 DOF manipulators are more likely to be capable of carrying out more complex tasks, than simple, fewer DOF manipulators. Seeing that surface tasks need rotation about three axes at the position for full flexibility, 5 and 6 DOF manipulators are surveyed [Larkum, 1992] in Tables 6.3 and 6.4.

Table 6.3 Manipulators with 5 DOF

International Submarine Engineering Ltd - 'Magnum 5F'

General	Joints	Range [degrees]	Speed [deg/m]	Actuator
Max reach 1.45m	Joint 1 (shoulder yaw)	180	20	rotary
Lift capacity 204 kg	Joint 1 (shoulder yaw)	100	20	linear
Weight 47.6 kg	Joint 2 (shoulder pitch)	90	20	linear
Hydraulic supply 22l/min at 69 bar	Joint 3 (elbow pitch)	100	30	linear
Construction - aluminium	Joint 4 (wrist rotate)	360 cont.	150	rotary

Merpro Ltd. - Mer 1200

General	Joints	Range [degrees]	Speed [deg/m]	Actuator
Max reach 1.24m	Joint 1 (shoulder yaw)	-90 to 90	Not avail	rotary
Lift capacity 120 kg	Joint 2 (shoulder pitch)	-65 to 35	Not avail	linear
Weight 68 kg	Joint 3 (elbow rotate)	360 cont.	Not avail	rotary
Hydraulic supply 10l/min at 100 bar	Joint 4 (elbow pitch)	-90 to 0	Not avail	Not avail
Construction - SS & HE30 alum alloy	Joint 5 (wrist rotate)	360 cont.	variable	rotary

Offshore Systems Engineering Ltd. - HE1

General	Joints	Range [degrees]	Speed [deg/m]	Actuator
Max reach 1.5m	Joint 1 (shoulder pitch)	-90 to 90	Not avail	linear
Lift capacity 27.5 kg	Joint 2 (elbow pitch)	-65 to 35	Not avail	linear
Weight 28.3 kg	Joint 3 (wrist pitch)	360 cont.	Not avail	linear
Hydraulic supply 5.5 bar	Joint 4 (wrist yaw)	-90 to 0	Not avail	linear
Construction - SS	Joint 5 (wrist rotate)	360 cont.	Not avail	rotary

Remote Technology Ltd. - RT6

General	Joints	Range [degrees]	Speed [deg/m]	Actuator
Max reach 1.4m	Joint 1 (shoulder yaw)	Not avail	Not avail	Not avail
Lift capacity 50 kg	Joint 2 (shoulder pitch)	Not avail	Not avail	Not avail
Weight 30 kg	Joint 3 (elbow pitch)	Not avail	Not avail	Not avail
Hydraulic supply 9l/min at 165 bar	Joint 4 (wrist pitch)	Not avail	Not avail	Not avail
Construction - Not avail	Joint 5 (wrist rotate)	Not avail	Not avail	Not avail

Schilling Development Incorporated - HV6F

General	Joints	Range [degrees]	Speed [deg/m]	Actuator
Max reach 0.92m	Joint 1 (shoulder roll)	90	300	linear
Lift capacity 40 kg	Joint 2 (shoulder pitch)	-45 to 45	300	linear

Weight 20 kg	Joint 3 (elbow pitch)	360 cont.	400	lineary
Hydraulic supply 1.5 GPM at 2000 psi	Joint 4 (wrist pitch)	-90 to 0	600	linear
Construction - SS & 6061 alum anodised	Joint 5 (wrist rotate)	360 cont.	0-90 RPM	rotary

Table 6.4 Manipulators with 6 DOF

International Submarine Engineering Ltd. - Magnum 7F

General	Joints	Range [degrees]	Speed [deg/m]	Actuator
Max reach 1.5m	Joint 1 (shoulder yaw)	180/270	20	rotary
Lift capacity 350 kg	Joint 2 (shoulder pitch)	90	20	linear
Weight 45 kg	Joint 3 (elbow pitch)	180	30	rotary
Hydraulic supply 10l/min at 1500 psi	Joint 4 (wrist pitch)	160	30	rotary
Construction - Alum. anodised	Joint 5 (wrist yaw)	180	30	rotary
Power Supply - Not avail	Joint 6 (wrist rotate)	360 cont.	150/25 RPM	rotary

Kraft Tele-Robotics Inc. - Grips

General	Joints	Range [degrees]	Speed [deg/m]	Actuator
Max reach 1.3m	Joint 1 (shoulder yaw)	180	20	linear
Lift capacity 45 kg	Joint 2 (shoulder pitch)	120	20	linear
Weight 59 kg	Joint 3 (elbow pitch)	110	30	linear
Hydraulic supply 11l/min at 105-210 kg/cm^2	Joint 4 (wrist pitch)	100	30	Not avail
Construction - Not avail	Joint 5 (wrist yaw)	105	30	Not avail
Power Supply - electric	Joint 6 (wrist rotate)	360 cont.	200/35 RPM	rotary

Merpro Ltd. - Mer1188

General	Joints	Range [degrees]	Speed [deg/m]	Actuator
Max reach 2.07m	Joint 1 (shoulder extend)	0 to 0.8m	Not avail	linear
Lift capacity 80 kg at 1.25m	Joint 2 (shoulder yaw)	-90 to 90	variable	rotary
Weight 88 kg	Joint 3 (shoulder pitch)	-90 to 20	variable	linear
Hydraulic supply 10-20l/min at 100 bar	Joint 4 (elbow rotate)	360 cont.	variable	rotary
Construction - Not avail	Joint 5 (elbow pitch)	-90 to 0	Not avail	Not avail
Power Supply - 24 VDC	Joint 6 (wrist rotate)	360 cont.	Not avail	Rotary

Norson Power Ltd. - Type2

General	Joints	Range [degrees]	Speed [deg/m]	Actuator
Max reach 1.4m	Joint 1 (shoulder yaw)	180	Not avail	linear
Lift capacity 25 kg	Joint 2 (shoulder pitch)	120	Not avail	linear
Weight 34 kg	Joint 3 (elbow pitch)	100	Not avail	linear
Hydraulic supply 9l/min at 165 bar	Joint 4 (wrist pitch)	180	Not avail	Not avail
Construction -SS	Joint 5 (wrist yaw)	115	Not avail	Not avail
Power Supply - 110/240 VAC, 11A	Joint 6 (wrist rotate)	360 cont.	Not avail	rotary

RSI Research Ltd. - Kodiak 1000

General	Joints	Range [degrees]	Speed [deg/m]	Actuator
Max reach 1.52m	Joint 1 (shoulder yaw)	60/120	40	not avail
Lift capacity 75 kg	Joint 2 (shoulder pitch)	0 - 90	15	linear
Weight 59 kg	Joint 3 (elbow pitch)	132	27	linear
Hydraulic supply 204 bar at 3000 psi	Joint 4 (wrist pitch)	120	32	Not avail
Construction - SS, Alum. & Naval Bronze	Joint 5 (wrist yaw)	110	23	Not avail
Power Supply - Not avail	Joint 6 (wrist rotate)	360 cont.	Not avail	rotary

Schilling Development Inc. - Titan III

General	Joints	Range [degrees]	Speed [deg/m]	Actuator
Max reach 1.98m	Joint 1 (shoulder yaw)	270	64	rotary
Lift capacity 114 kg	Joint 2 (shoulder pitch)	120	64	linear
Weight 102 kg	Joint 3 (elbow pitch)	270	112	rotary
Hydraulic supply 1.5-5 GPM at 3000	Joint 4 (wrist pitch)	180	240	rotary
Construction -Titanium	Joint 5 (wrist yaw)	180	240	rotary
Power Supply - Not avail	Joint 6 (wrist rotate)	360 cont.	0-90 RPM	rotary
	Joint 6 (gripper)	97 mm	N/a	linear

Slingsby Engineering Ltd. - TA37

General	Joints	Range [degrees]	Speed [deg/m]	Actuator
Max reach 2.07m	Joint 1 (shoulder yaw)	-10 to 90	Not avail	linear
Lift capacity 27 kg	Joint 2 (shoulder pitch)	-20 to 90	Not avail	linear
Weight 54 kg	Joint 3 (elbow extend)	0-300mm	Not avail	linear
Hydraulic supply Not avail	Joint 4 (elbow rotate)	340	Not avail	rotary
Construction - Not avail	Joint 5 (wrist pitch)	80	Not avail	linear
Power Supply - not avail	Joint 6 (wrist rotate)	340 cont.	Not avail	rotary

Western Space and Marine Inc. - Arm MK37

General	Joints	Range [degrees]	Speed [deg/m]	Actuator
Max reach 0.94m	Joint 1 (shoulder yaw)	180	180	Noy avail
Lift capacity 23 kg	Joint 2 (shoulder pitch)	120	170	Not avail
Weight 43 kg	Joint 3 (elbow pitch)	180	150	linear
Hydraulic supply 1.2GPM at 3000 psi	Joint 4 (elbow rotate)	220	300	rotary
Construction - Alum. or Titanium	Joint 5 (wrist yaw)	120	300	rotary
Power Supply - 24 VDC, 20A	Joint 6 (wrist pitch)	100	300	rotary

The above brief survey gives the indication of what is available on the market, it is by no means exhausted, apart from brief kinematic specification and some basic data, provides the names of main companies world-wide, manufacturing manipulators.

6.5 Summary

Assemblying the information with critical appraisal regarding existing enabling technologies has established the data platform for the robot's selection. Additionally, this chapter completed the research relating to full comprehension of the problem for which successful robot design is required. The physical description and identification of the remaining auxiliary components, such as device's delivery system to the task, and probe or tool carrying arm, finished the preliminary stages of understanding the design problem, within the overall engineering design. The preparatory stages are concluded and the conceptual level prototype options commences the second part, developemtal part of the research. This concluded the 'engineering' or 'macro' part of research and created the framework for the 'analytical' or 'micro' part.

In the following chapters the prototype is assembled, the case study model identified and the robot's configuration based on the pre-set assumptions and conditions concluded from chapters 4,5 and 6 are set up and investigated.

PART TWO

DEVELOPMENT OF OPTIMISATION APPROACH AND GA APPLICATION WITHIN ED FRAMEWORK

CHAPTER 7 - ROBOT'S MODEL AND FRAMEWORK FOR OPTIMISATION

7.1 Introduction

With the previous chapter, the part of the research dealing with the information collection and the descriptive data assembly and assessment is completed and the set of prototype models is anticipated. This chapter initiates the analytical and conceptual part of the research. All the information and data analysis carried out in Chapters 3 to 6 enable the prototype assembly for further optimisation. The assessment and the positioning of the design within ED process further reassures that the design is carried out in a structured manner towards the optimal result.

Further, this chapter devises the conditions and the approach to the optimisation and the analysis of criteria for robot's performance and robot's parameter identification reflects the objective (iv) from Section 1.3. Selection of the most effective robot for a designated range of activities is a complex and labour intensive process. Therefore, a methodology needs to be devised to speed up the selection process and improve the accuracy of the task suitability.

As one of the aims of this research is to develop a methodology for selecting a bridge restoration robot with optimal values of the chosen (kinematic) parameters, based on the named criteria, the framework has to be set up before any tool can be employed. The contribution to the thesis is outlining the approach to the optimisation and setting up the model.

Therefore, in this chapter the author not only models the problem for optimisation, but selects and justifies the parameters and the criteria.

7.2 Design Positioning within ED Process

According to the stages of ED process, in Figure 3.1, the clarification of the task phase has been carried out, except for identification of criteria and relevant parameters. This is carried out in the following Section. Once the criteria are identified and in depth review of available NDT and restoration methods and enabling technologies is completed, a set of alternative solutions for further evaluation can be assembled at this point. A set of prototype model should emerge for further optimisation. As this is a research project, certain limitations are assumed. One prototype model is constructed and it is furthered modelled into a mathematically applicable system for optimisation operations. The overall optimisation procedure and approach to criteria evaluation and parameters optimisation has its origin within the conceptual design phase but the approach and results actually produce level of outcome characteristic for embodiment design phase. Therefore, the devised approach spans over both phases.

7.3 Parameters and Criteria Classification Outline

The generic approach to the problem of optimal robot design or selection, which would produce the most versatile option, is to specify all the required tasks in a given environment, extract all the robotic configurations, analyse them and determine the best configuration for a particular mission. In such a way, even for a limited number of configurations, the number of combinations of robot and environment interaction points would grow combinatorially. Equally valid is the fact that there is no single correct solution, as there are multiple, often competing, objectives.

The following are the most important kinematic parameters in a construction robot selection: (i) the total number of Degrees Of Freedom (DOF), (ii) the robot's major configuration (the first 3 DOF counted from the carriage to the tool), (iii) its minor configuration (the wrist); (iv) the length of the links, (v) the joints working ranges (angles) and (vi) the joints' motion characteristics (velocity and acceleration).

Establishing the criteria for the parametric design is a direct consequence of the findings from the QFD application. The customer requirements (assembled in Table 3.1) must be translated into measurable design targets for identified parameters. As this research is only an academic one and the QFD was only indicatively employed, only some 'design targets' or criteria were identified.

The relevant criteria in robot selection are: (i) singularity avoidance, (ii) collision avoidance (making sure that the robot will not collide with its environment), (iii) cost, (iv) productivity, (v) dexterity (the tool ability to achieve required orientation), (vi) optimal task planning (finding an optimal sequence of motions, given a series of task specifications), (vii) versatility, here the number of different tasks that the robot is capable of performing, in the number of different workspaces, (viii) manoeuvrability (how free the robot is to perform its task in the confined bridge environment) and (ix) percentage of coverage (to what extent the robot can perform the entire task).

Out of all the criteria listed overhead, the following are selected for the parameters' optimisation in this research: (i) collision and singularity avoidance, (ii) dexterity, (iii) productivity and (iv) percentage of coverage. The above selection followed the analysis of the functional and physical characteristics of the customers' requirements and the related performance of the kinematic parameters. The human factor requirements, safety, time, cost, standards, testing and codes of practice fell outside the scope of the research.

The above criteria, although grouped together, represent different levels of importance. Collision and singularity avoidance has to be always ensured, as this determines the ability to deliver the task safely. Collision avoidance, as referred-to in this approach, means that any part of the robot does not intersect at any point with the bridge, which in this example depends on the location of the tool, and not its orientation and is a criterion of primary importance. Dexterity determines the aptitude to reach required locations with specified orientations in the predetermined trajectory. In other words, it indicates the regions of the working envelope not reached due to insufficient orientation flexibility of the tool. Full assessment of productivity cannot be fully carried out without the full dynamic appraisal of the robot and here only time of the task completion is minimised. As the robot in this research is optimised kinematically, percentage of coverage specifies the range of the assigned task, which the robot can do, thus indirectly it indicates the amount of manual involvement in the task completion and, again, is a function of the location rather than the orientation.

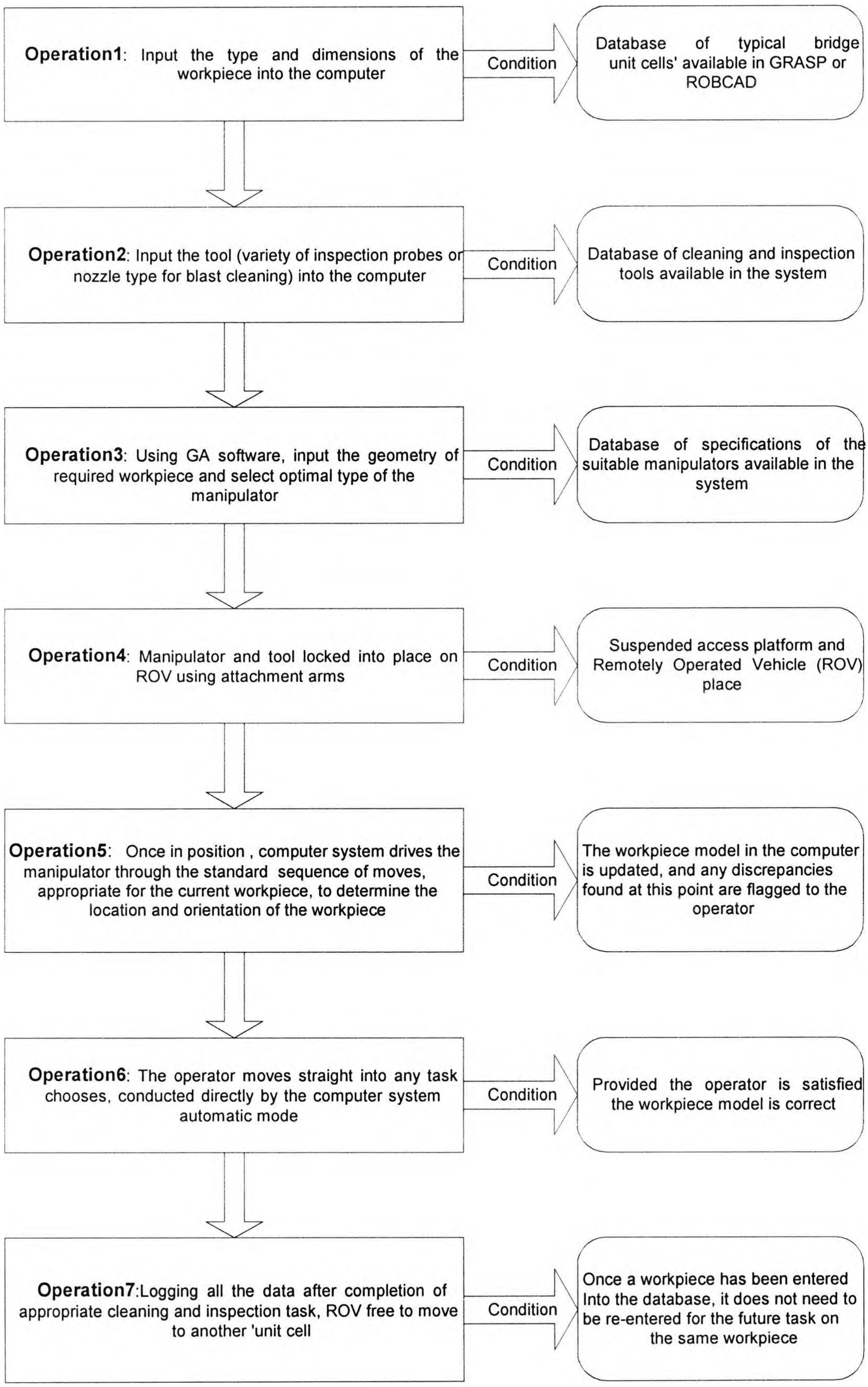
The parameters listed in the beginning of this paragraph also require further expansion. The first two parameters deal with the type of the robot joints, of which two basic types are commonly used - they are either revolute (R - exhibit a rotary motion about an axis) or prismatic (P - exhibit a linear motion along an axis. The axes of the first three joints cater for the delivery and location of the tool, therefore these are also referred to as the major axes. The axes of the remaining joints, referred to as the minor axes or the wrist, establish the orientation of the tool. The geometry of the work envelope is mainly determined by the sequence of joints used for the first three axes. The dexterity of the envelope is governed by the wrist configuration.

Simultaneous assessment and optimisation of several parameters can only be justified if they show interdependency. In this context, interdependency is displayed as the mutual influence (altering one of the parameters, automatically results in changes in other parameters). For example, changing the type of configuration may imply different types of motors (rotary vs. linear) with different working ranges and motion characteristics. Altering the link lengths within the same configuration results in changes in the working ranges of joints in order to maintain the size of the working envelope. A need for increased joint motion (velocity and acceleration) results in different kinds of motors, and therefore influences the working ranges of joints, or even their type and weight. Changes in the wrist configuration, being mainly responsible for tool orientation, lead to the requirement for altering the joint working ranges and link lengths in order to maintain the ability to deliver a specifically oriented tool in the workspace.

7.4 Towards a Prototype

At this stage data collection and analysis is completed and together with the results of QFD analysis in Chapter 3, this can be applied to outline the form of an automated facility (AF). As pointed out earlier, self-mobility is not assumed and the device relies on being delivered to the task location. Access systems to support this were reviewed in Section 6.2. The automated facility itself is assumed to be a 5 or 6 DOF tele-robotic arm. Subsequently, the kinematic parameters of the arm are subject of GA optimisation were discussed in Sections 7.3. Once the access and the general outline for the facility is resolved, the stages described in Table 7.1 give the framework of activities leading to completing the bridge restoration task.

Table 7.1 Stages of a Typical Operation Using the Prototype


Operation1: Input the type and dimensions of the workpiece into the computer
Condition
Database of typical bridge unit cells' available in GRASP or ROBCAD
Operation2: Input the tool (variety of inspection probes or nozzle type for blast cleaning) into the computer
Condition
Database of cleaning and inspection tools available in the system
Operation3: Using GA software, input the geometry of required workpiece and select optimal type of the manipulator
Condition
Database of specifications of the suitable manipulators available in the system
Operation4: Manipulator and tool locked into place on ROV using attachment arms
Condition
Suspended access platform and Remotely Operated Vehicle (ROV) place
Operation5: Once in position , computer system drives the manipulator through the standard sequence of moves, appropriate for the current workpiece, to determine the location and orientation of the workpiece
Condition
The workpiece model in the computer is updated, and any discrepancies found at this point are flagged to the operator
Operation6: The operator moves straight into any task chooses, conducted directly by the computer system automatic mode
Condition
Provided the operator is satisfied the workpiece model is correct
Operation7:Logging all the data after completion of appropriate cleaning and inspection task, ROV free to move to another 'unit cell
Condition
Once a workpiece has been entered Into the database, it does not need to be re-entered for the future task on the same workpiece

The picture in Figure 7.1 shows the general image of the platform, the bridge, as selected from Table 4.1 and the articulated arm performing work on underside of the metal deck.

Figure 7.1 View of the System while Executing Blasting on U/S of the Decking

Figure 7.2 View of the Prototype at Work on the Secondary Beam

In order for automated systems to continue their growth and become more versatile, frequently applied and inexpensive, it is necessary for a series of applications to spread across industries (diversification), as well as to use already existing components, through assembly (simplification). The situation in the nineties verifies the above claim, when building purpose serving, complex, expensive and inflexible in application hardware had to be abandoned in order to improve reliability, efficiency, practicality and cost.

7.5 Case Study Model for Optimisation

The task required of the arm is to position the wrist. The type and the sequence of the arm's DOF (starting from the base joint) defines the workspace, the manipulator's end effector is able to access. Therefore, after analyses of the workspaces of the arm configurations (Cartesian (PPP), cylindrical (RPP), spherical (RRP), SCARA (RRP) and articulated (anthropomorphic - RRR)) and bridge geometries (Figures 4.1-4.15), it became apparent that the optimisation of the major configuration is between the spherical and the articulated configuration, as shown in Figure 7.3.

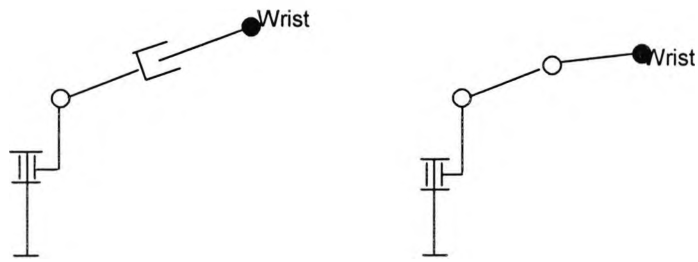


Figure 7.3 Schematic Representations of RRP and RRR Configurations

Consequently, the choice of the two configurations in this research is governed by: (i) the operational requirements of the restoration task, (ii) the geometry of the workspace and (iii) the form and context of the transformation matrices of the direct kinematics calculations.

The choice of the criteria and serving parameters allows separating the whole optimisation approach into two stages.

In the first stage, the robot representation (to be optimised) (see Figure 7.1) is separately developed for both configuration types (RRR and RRP) and is using the concept of the direct kinematics. The procedure for the computation of direct kinematics is derived from the open kinematic chain of the manipulator structure and the Denavit-Hartenberg convention was adopted [Denavit and Hartenberg, 1995].

$$T_0^6 = T_0^3 \times T_3^6 = \begin{bmatrix} R_0^3 & d_3^6 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} R_3^6 & d_3^6 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_0^6 & d_0^6 \\ 0 & 1 \end{bmatrix} \qquad 7.1$$

A robot's gross work envelope is defined as the loci of points in three-dimensional space that can be reached by the end-effector. The tensor of joint variables, using direct kinematic calculations, determines the position ('d' part of the transformation matrix in Equation 7.1) and orientation ('R' part) of the end-effector. Therefore, the set of the kinematic parameters specifies indirectly (before direct kinematics calculations) the location of the end-effector.

Hence, a string of such sets of parameters geometrically describes set of loci in space and several series of such points relate to the trajectories, 'strategically' placed on the boundaries of the workspace. Minimising the distance between the kinematically calculated locations and the predetermined 'desired' end-effector positions (paths) is the essence of the approach. The minimised distances 'produce' the optimal values of the robot parameters and, based on those, further investigation can quantify the percentage of coverage.

Precise analysis of the robot's activity is necessary at each task location of the robot. The robot end effector can operate two types of movement depending task characteristics, and consequently, on the control provisions. The first type is point-to-point (PTP) motion, where the tool moves to a sequence of discrete points in the workspace. The user does not explicitly control the path between the points and therefore, typical applications include: (i) spot welding, (ii) pick and place and (iii) loading and unloading. PTP motion can be risky for surface work, where collision is likely. The other type of motion is continuous-path, or controlled-path motion (CPM), where the end effector must follow a prescribed path in 3D space. Typical activities include: (i) arc welding, (ii) spraying and (iii) gluing.

It is important to identify as early as possible the type of motion required for the investigated restoration activity and assess every probe and tool from this perspective. Not only the distance from the surface and the orientation have to be addressed, but also the size of the mapped grid. For spraying and blasting the close analysis of the orientation to the surface, distance required and the size of the spraying cone determine the width of the treated strip. The satisfactory uniform coverage of the surface is secured by controlling three variables: (i) distance from the surface, (ii) required orientation (45° for blasting, 90° for spraying) and (iii) required flow rate of the consumable.

The number of parameters varies for both configurations. The spherical configuration (RRP) has two rotary motions and the third variable prismatic, axial motion, therefore the third joint provides the reach (hence the parameters are θ_1, θ_2, $\mathbf{d}_2$). RRR has three variables assumed as the angle movement ranges of three revolute joints and the fourth marking the split in total link length so the reach is additionally addressed as a unity divided between two links which becomes an extra parameter (θ_1, θ_2, θ_3, $\mathbf{L}_1$ where $\mathbf{L}_1+\mathbf{L}_2=1$).

Due to the different number of parameters, separate computations have to be carried out for each configuration and a preference established, based on the qualitative analysis of representations with the highest evaluation scores. This evaluation requires assessing the collision avoidance against given boundaries of the working environment, represented by an externally input tensor **C** (containing the corner points of the collision envelope) (Figure 7.4) and the percentage of coverage. To address the latter, the origin, which is the robot's base, is placed centrally within the workspace and provisionally 500 mm below the underside of the secondary steel beams, to correspond to the robot's position on the mobile trolley. In order to perform a task (for example paint stripping) within a typical bay, the nozzle has to run underside the beams, along the sides of the beams and underside the deck, giving twelve points in total. Figure 7.5. illustrates a typical working space, a perimeter of collision envelope and the points along trajectories to be followed by the tool.

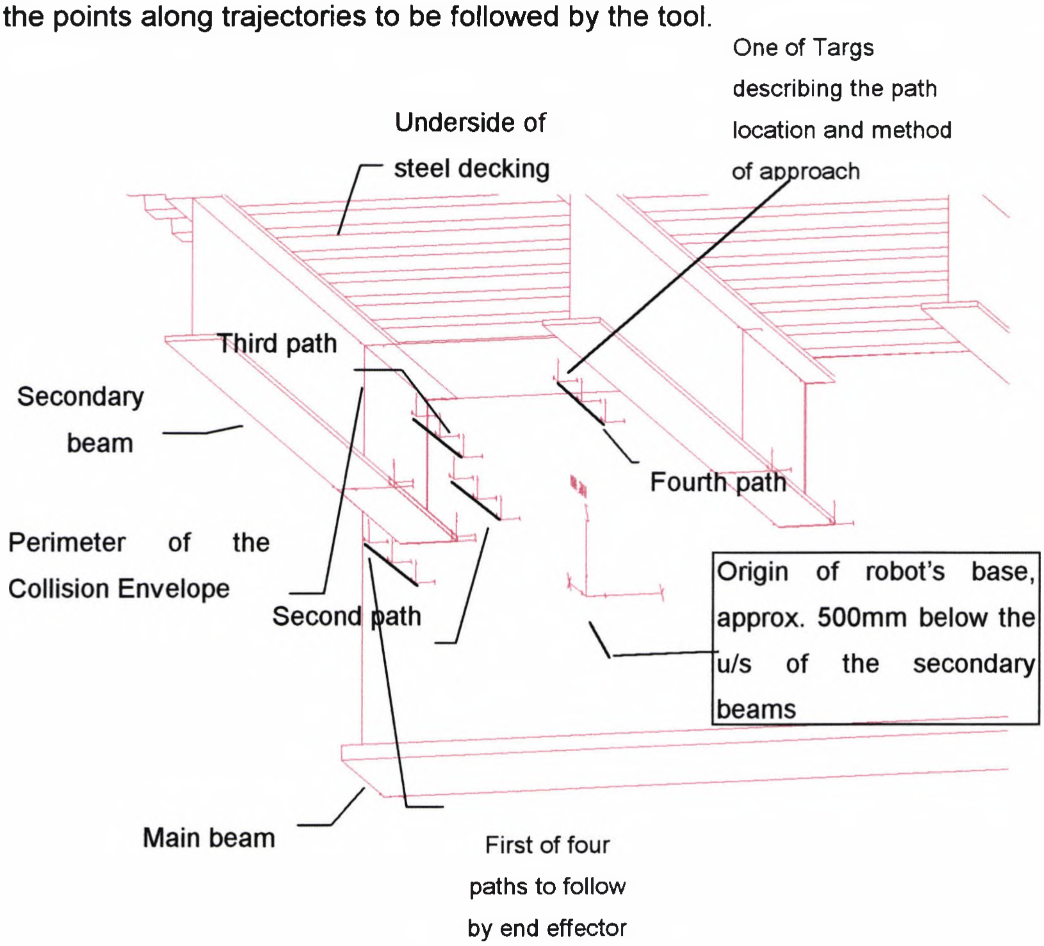


Figure 7.4. Graphical Task Description

Although, the number of independent parameters is only four for RRR and three for RRP, the number of computed parameters reflects the number of points to be reached, here twelve (four paths of three points). Each point implicitly includes all configurations' parameters, as they (after direct kinematic calculations) indicate location of the end effector in space and relate to a point along the path. If the distance between all locations of the end effector and path points can be minimised without violating the optimisation criteria, then the values of the parameters indicate the optimal ranges within every parameter's movement sector. The results of the first stage computations yield the 'fitter' configuration for the given task (outlined by the location of the path points and collision envelope).

In the second stage, the parameters of the preferred main configuration from the first stage are augmented by the choice of the minor configuration (wrist) - the last 3 DOF, responsible for the orientation of the tool and velocities and accelerations of all six joints. This allows evaluating the candidate solution with respect to dexterity and productivity.

Compatibility of the dexterity between the wrist within the representation and the required 'ideal' one, due to the nature of the task and constraints of the environment, is achieved by comparing the relevant parts of the orientation matrices resulting from direct kinematic calculations.

Task completion time depends on the type of algorithm controlling the robot's motion, with the optimal control incurring a severe real-time computational effort. In addition to this computational load, mechanical constraints such as torque, speed and acceleration of gears and harmonic drives performance place physical limitations on the robot's speed. This is why the optimisation of the total time control has rarely been applied to, in robot design, and an alternative approach to addressing productivity has to be assumed. Co-ordinated motion (i.e. the simultaneous movement of several manipulator axes, in order to move the end effector to the desired location) of all axes is assumed. The data, which include the angles of rotational (or the distances of translational) movement of each joint and the corresponding speeds and velocities, allows the calculation of the time required to rotate (or move) each joint. The trapezoidal velocity profile is assumed to simplify the model. However, it should be understood that state-of-the-art motors use the parabolic velocity profiles with varied acceleration. The longest axis time is an indication of the approximate overall time required moving the manipulator i.e. the least time move. At this point it is vital to stress that the type of motion (point-to-point or continuous), and therefore, the control are not addressed here. The above approach to productivity is therefore only an indication of the comparative duration of similar movement.

7.5.1 Collision and Arm Singularity Avoidance

Locating the corner points in space, which enclose certain shape, geometrically creates the assumed collision envelope. On the bridge geometry, the line joining all the 3-D points represents the contour of the obstacle (**C**) within the workspace. These points belong also to the intersecting planes, which make up the 'walls' enclosing the collision contour (see Figure 8.1 for assembly of planes, as isolated from Figure 7.4). Direct kinematic calculations result in matrices specifying the co-ordinates (x,y,z) of the tool and elbow for each arm position, relating to every point along the required path.

The physical fact of the impact between the robot's arm and any plane of the collision envelope can be addressed using geometry and the equations for intersecting straight lines and planes. The intersection point between the plane and the line passing through the tool and elbow limits, lies simultaneously within the boundaries of the plane and boundaries of the arm, (between tool and elbow co-ordinates). Each location (co-ordinates) of the tool's and the elbow's position within the representation is checked separately, against all planes of the collision envelope.

Singularity takes its origin in multiplicity of the solutions of inverse kinematics. If it occurs while reaching two adjacent points, the trajectory cannot be followed continuously. Indication of possible singularity occurs when the joint's rotational movement increment, while approaching neighbouring points, displays significant change. Hence, if the condition is imposed that the angles' changes, while reaching closely located points, are kept small, the singularity is potentially avoided. In practice, if the differences between the values of the angles of rotational movement of the same joint, reaching adjacent locations exceed certain, small, pre-programmed value, the candidate solution is dismissed from further evolution.

For the RRP configuration, additionally to major configuration's singularity considerations, the manipulator is in singular configuration, when the wrist centre intersects the centre of the base, as rotation about the base leaves this point fixed [Spong and Vidyasagar, 1989]. This can be avoided if the rotation angle of the second revolute joint is not allowed to reach the following values: $\theta_2 = 0$, π or 2π. Singularity in the RRR configuration occurs when: (i) the wrist centre intersects the axis of the base, hence when $(1-L_1)\cos(\theta_2+\theta_3)=-L_1\cos\theta_2$ or (ii) the elbow is fully extended or fully retracted (this occurs when the rotational angle of the third joint (θ_3) equals to 0 or π).

If any of the arm positions collide with any of the planes, or singularity is detected when reaching consecutive path points, the whole representation is assigned high penalty as a result and the overall suitability of the candidate solution diminishes. So, it remains only to secure no intersection between any point along the arm and the contour of the obstacle to address collision avoidance.

7.5.2 Percentage of Coverage

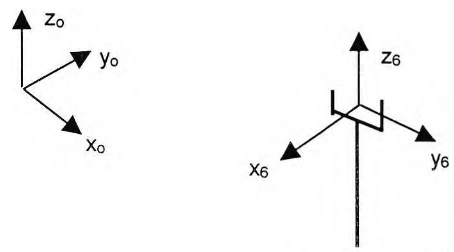
The criterion of percentage of coverage includes information about the portion of the work volume done by the robot and hence it identifies which part will need further manual completion. Spraying paint and the interior finishing painting task is well analysed in Rosenfeld et al. [1994], and although similarity with the performance of the blasting nozzle and paint spraying nozzle are similar, Rosenfeld's proposal of task analysis should not be followed blindly. Instead, it is an excellent approach to follow, while analysing data based on manual sandblasting, in order to assess distance from the surface, speed of movement and flow of the material. Trajectories to be followed by the tool are located in such positions, as to enable the complete paint stripping of a typical bay and are meant to be reached consecutively. At every iteration, an algorithm developed (in the present study for this purpose), generates twelve locations of the tool, calculated using direct kinematics calculations on the encoded parameters. The desired twelve locations of the robot's end effector are separately mapped within the working environment (external tensor **P**). The most suitable candidate robot is selected on the basis of the minimum value of the sum of distances between the two sets of points described above, which meets the criterion of the maximum percentage of coverage.

By analysing the values of the parameters in the best candidate for the optimal solution, in each generation, and by observing the standard deviation of the set of distances, it is feasible to assess and authenticate the distance between the required path and the achieved tool positions. Two scenarios are feasible. In one - certain path points are never reached and good overall quality of the solution comes from high precision in reaching only some points along the path. This can provide the higher value of the standard deviation. The second scenario provides a different optimal solution, where all points are approached with nearly equal proximity, which on average also results in a solution of good quality but is also represented by the lower value of the standard deviation. The sought after solution will have both, high fitness and low standard deviation. Further, a separate analysis of the quality of the best solution can be performed for example graphically (using GRASP or ROBCAD), just to allocate the percentage of manual engagement.

7.5.3 Dexterity Assessment and Manipulator Singularity Avoidance

There are several methods of addressing wrist dexterity [Klein and Blaho, 1987] depending on the chosen definition of dexterity. The definition used in this thesis identifies dexterity as high level of proximity among the relevant components, between the rotational part of two overall transformation matrices. In this approach, it has been recognised that delivering the tool to the task, accompanied by its most suitable orientation, has to comply with the tool characteristics and task specification. Numerically, this means close proximity between the relevant components within the 'ideal', rotational part of the entire transformation matrix. See Figure 7.5 for the approach to dexterity part of the transformation matrix compatibility.

The choice of wrist configurations is governed by their applicability and versatility, and is concentrated on RPY (roll, pitch, yaw) and RPR (an Euler or spherical wrist). The evaluation of the wrist type arises from the compatibility of the orientations of the standard ortho-normal bases of all points along the trajectory (oriented in relation to the base co-ordinate frame) and the orientation of the ortho-normal basis of the wrist (tool attachment point). This orientation is displayed (in relation to the same co-ordinate frame) in the rotational part ('R' part of the 7.1 equation) of the final transformation matrix calculated for all 6 DOF using direct kinematic calculations. The first column of that matrix represents the projection of the X-axis of the wrist onto all three axes of the robot's base frame (x_o, y_o, z_o), the second column represents the projection of the wrist's Y-axis and the third, the wrist's Z-axis. All points along the trajectory must also be oriented (in relation to the ortho-normal basis of the robot's base frame) in the required tool approach position (compatible with the nature of the task and the geometry of the workspace). Therefore, they are allocated the unit vector's ortho-normal bases (referred to as 'Targs') suitably rotated. Level of compatibility between the equivalent elements of the ideal and given matrices assessed for each trajectory point, results in an orientation data.



Tool up

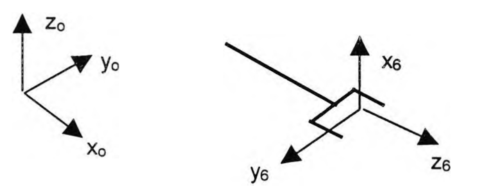
No tilting

Irrelevant as rotation around Z permissible

X6	Y6	Z6	
++	1	0	X0
-1	++	0	Y0
0	0	1	Z0

No tilting

z_6 along z_o



Tool horizontal (1)

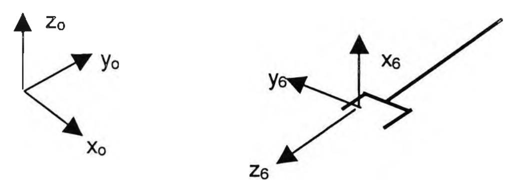
Irrelevant as rotation around Z permissible

z_6 along x_o

X6	Y6	Z6	
0	++	1	X0
0	-1	++	Y0
1	0	0	Z0

No tilting

No tilting



Tool horizontal (2)

No tilting

Irrelevant as rotation around Z permissible

X6	Y6	Z6	
++	-1	0	X0
0	0	-1	Y0
1	++	0	Z0

No tilting

z_6 along y_o and opposite

Tool horizontal (3)

No tilting

Irrelevant as rotation around Z permissible

X6	Y6	Z6	
1	++	0	X0
0	0	-1	Y0
++	-1	0	Z0

No tilting

z_6 along y_o and opposite

Figure 7.5 Dexterity Matrix Compatibility

The first matrix is the required one, set up after evaluating tool's position when performing the task in the workspace. The second matrix is calculated for every member of the population through generations and specifies the orientation of the tool frame relative to the base frame for the current set of the parameter values.

After analysing the blasting operations (position and orientation of the tool in order to carry out the task), it becomes apparent that the tool has to approach the work surface orthogonally, without tilting. Rotation of the nozzle around the longitudinal axis is irrelevant for the direction of the stream of the blasting particles. If the tool is to be oriented vertically, pointing upwards, in order to reach three points under the secondary beam (first path) and three points underside the deck (fourth path) on Figure 7.4. In matrix terms it means that values of the first two components in the first two rows are insignificant, as rotation around roll axis is permitted, the first two components of the third row and column are expected to be zero, not allowing for tilting and the last member on the diagonal should equal to one. For the tool to be oriented horizontally, means following two sets of three points on the side of the lower (second path) and higher (third path) part of the secondary beam, and being on the negative side of the y_0-axis of the base's origin, as indicated on Figure 7.4. Matrix-wise this means that the values of the first two components in the first and third row are insignificant, as rotation around roll axis is permitted. Additionally, the first two components of the second row and the first and the third component of the third column to be zero, with no tilting allowed and the middle component of the third row to equal minus one as the working surface is on the negative side of y_0 axis.

Singularity of the arm is addressed again at this stage by simply limiting the range of travel of its joint variables, resulting from the first stage. The additional considerations have to be taken into account when both arm and wrist are analysed. The boundary singularities, as considered not particularly serious [Schilling, 1990], are omitted. The interior singularity is the one, which is potentially more common and it occurs when two or more of the revolute joint axes become co-linear. This situation occurs for an Euler (spherical) wrist when $\theta_5 = 0$ or π and in case of RPY wrist, when $\theta_5 = \pi/2$ or 1.5π [Ranky and Ho, 1985]. Therefore, these limitations are added to the second stage computations in similar manner as for the first stage. If singularity threat is detected in a candidate solution, a penalty value is added, which discredits the candidate.

7.5.4 Productivity

The criterion of productivity is addressed as a guideline only, as the dynamics of the robot are not included in the analysis. The speed of each joint of the robot is controlled independently, therefore, the robotic manipulator is capable of continuous path motion control with the speed of the tool regulated. The trapezoidal speed profile means that the robot accelerates to a running speed and then decelerates to zero, which effectively limits the maximum acceleration required to reach the running speed, if the two adjacent locations to be reached, are too close [Navon and Warszawski, 1992].

The movement of all six motors is assumed as simultaneous and the time required to move them through the movement sector of each joint is calculated separately for each motor, using the trapezoidal velocity profile. Under this, the longest time is accepted as the overall time for reaching a certain location along the trajectory. For the path of all twelve points, the twelve longest times are added together and are acknowledged as a total time for reaching all locations on the path. The representation with the shortest overall time is favoured in the evaluation.

The productivity calculations follow the dexterity, therefore the penalty from dexterity assessment is also added to a candidate for this phase of optimisation.

7.6 Summary

This chapter has initiated the analytical part of the research, aiming at the optimisation of the kinematic parameters of the bridge restoration robot. The prototype model has been assembled, based on the outcome of the study in Chapters 4 to 6, for further optimisation. The prototype has been further mathematically modelled and set for optimising the arm configuration with respect to the task (environment). The parameters and criteria have been identified, analysed and justified. Task application has also been considered in detail. The optimisation process aims at the initial identification of the superior configuration for the given task (the spherical (RRP) or the revolute (RRR)). The procedure to optimise the limits on the ranges of the movement sectors of the major configuration's joint variables has also been set up. This information has not only led to the evaluation of the percentage of necessary manual involvement, but has assisted in the kinematic design and the choice of actuators. The second stage of optimisation has refined previously optimised ranges of the parameters, for the second set of criteria (productivity and dexterity). Additional parameters have included the preferred choice of the wrist configuration (the choice here is between the Euler and the RPY configuration), the movement sectors of the joints within the wrist, which in turn, aid the choice of the actuators and the geometry of the end effector. The optimal values for the velocities and accelerations have reinforced the preference for the joints' actuators. Setting up of the whole procedure and assuming the approach laid out in this chapter (in order to optimise robot's kinematic parameters) is part of the unique and original contribution to solving the problem of selecting the optimal robot for the pre-determined task.

The following chapter introduces the optimisation tool (Genetic Algorithm) and the method of application of GA to the problem.

CHAPTER 8- APPLICATION OF GENETIC ALGORITHMS TO CASE STUDY MODEL

8.1 Introduction

The model of the robot and the whole approach to optimisation of its kinematic parameters was introduced in the previous chapter. Further developments before the model can be optimised include the justification and the choice of the optimisation tool. The general choice of the optimisation method and its suitability to the problem is addressed in the early parts of the chapter. Then, the computing tool is selected and the software developed. The modelling of the problem in GA-applicable format follows and is explained in detail, in order to fully utilise GA capabilities and also achieve overall objectives of the approach to yield reliable, optimal outcome, under the pre-determined criteria. This allows straightforward development of the computing tools (software) to carry out the computations, which is covered in Sections 8.6.

The selection of the GA application and liasing it with the optimisation approach in order to achieve feasible results is one of the major contributions to the overall research into the bridge restoration robot.

The author feels that the general overview of GA method does not contribute to the original input into the research, and therefore, is taken outside this thesis and is included in Appendix B.

8.2 Justification of GA Choice as Optimisation Tool

Genetic algorithms (GA) are search algorithms that use the notions of natural selection and genetics. Each technical problem can be translated into a genetic one that can be optimised by means of biological rules. The basic processes used are survival of the fittest, with information exchange among the survivors. Like biological systems, there is some randomness to this process, but instead of causing detrimental affects, this randomness gives GA robustness and the ability to generate better solutions. Even though they are based on the laws of coincidence, they do not represent an aimless search for an optimum. Rather, they take advantage of pre-information in order to derive improvements from it.

Optimisation techniques based on genetic rules of selection and improvement of individuals' performance and characteristics through generations, offer a possible tool, as they allow optimisation and analysis of the mutual interrelations of several parameters simultaneously and are capable of intelligently searching large, discontinuous sets of possible solutions. It requires representing the parameters in special, coded strings. These strings combining all the parameters, become the candidate solutions to a problem. In general terms, the representation of the design is a direct consequence of the choice of parameters for optimisation, while the criteria serve to evaluate the quality of the solution.

It is relatively simple to automate the design as well as the evaluation of experiments. This method ensures to a very high degree of probability that the absolute optimum will be found, thus the complete potential of the product will be recognised and utilised. Although this method does not require any pre-requisites, a parameter reduction is recommended, since the number of experiments increases proportionally to the number of parameters to be optimised. A drawback of the method is that a great deal of experience is required in order to be able to estimate the number (and thus the cost) of the experiments. Furthermore, it is not possible to distinguish significant from non-significant parameters. Therefore, it is necessary to make this assessment outside the GA optimisation process and allocate the parameters to different stages of optimisation, externally.

For more comprehensive introduction to GA, the reader is referred to Appendix B.

8.3 Approach to GA Analysis

Designing a bridge restoration robot requires optimisation of its parameters related to the previously selected criteria (Chapter 3 and 7). Because of the large number of parameters and the wide range of the potential values they can assume, the number of alternatives to be evaluated is immense. Genetic Algorithms are proven to be an excellent technique for exploring large search spaces for optimal or near optimal solutions [Goldberg, 1989], as explained in the previous section. In robot selection and design the technique also showed promising results [Davidor, 1990].

The criteria upon which the series of parameters is assessed, may be selected from the following, the task specification, expected design performance, workspace characteristics (total volume swept out by the end effector, as the manipulator executes all possible motions), clients requirements, various aspects of cost, available enabling technologies, etc. Some of the criteria are of primary nature, as their importance to the robot's performance is crucial. Secondary criteria only improve the robot's performance and their choice depends on the nature of the task and client's brief.

The representation of the robot is based on the gross working envelope of a robot as defined by the locus of points in the three dimensional space which can be reached by its end effector. The vector of joint variables and robot geometry, using direct kinematic calculations, determines the position and orientation of the end-effector. If the vector of parameters specified above is compiled in a GA applicable format, every chromosome represent a point in the 3-D space, which the robot is capable of reaching. Therefore, a string of chromosomes can geometrically describe several series of such points, or otherwise paths or trajectories, within the envelope. Assessment of how close these points are to the required predetermined end-effector locations yields preferable values of the robotic parameters, based on the criteria of percentage of coverage.

The required trajectory's geometry, orientation and location is a direct product of the geometrical analysis of the workspace (here, the geometry of a selected part of the bridge) and assessment of the automation potential and operating conditions of the task tools (here, the manoeuvring, position and orientation of the blasting nozzle). The blasting nozzle is selected as a model tool for allocating a relevant movement and location of path points. If NDT activities are selected, a different type of movement is required. One of the typical NDT activities is guiding the probe over the grid of points, with the brief contact with the surface at the grid corners where the probe is required to take measurement. The relevant sequence of the tool positions would be a grid of points on the surface and the identical grid lifted from the surface by a distance required to lift and move the probe to the next grid point. So although the paths are different, the principle is similar and emerges directly from the task analysis and can be applied to variety of the required tool behaviour.

Theoretically, it is possible to represent all the optimisation parameters in a single chromosome, and all the points of interest within several trajectories in a single string of chromosomes. However, performing genetic operations on such a long representation and then a quality or 'fitness' assessment of each string in the population, for each generation, increases the computing time beyond the "reasonable" time. Consequently the optimisation process is divided into two stages. At the first stage the robot configuration, the lengths of the links and the joints' motion ranges are optimised. At the second stage, the wrist configuration and the joints' speed and acceleration are added to the optimal configuration from the first stage. It is possible to divide the process into two stages without violating the simultaneity principle, as explained in following section.

8.3.1 Stage One

Initially, the dimensions of the arm links and joints' working sectors of the major axes are operated on. The working sectors of the joints refer to the reachable ranges in joint movement despite the physical limitations in the joint construction. The orientation of the tool (minor axes) is not taken into account at this stage, as for the task of the coating-stripping the tool operational requirements are not too sensitive to minimal changes in orientation.

The following six parameters represent kinematics of the robot and are optimised here: (i) choice between two major configurations: (a) spherical RRP - first two joints of revolute/rotary (R) type, with angles of movement θ_1,θ_2 respectively, the third is prismatic/linear (P) $\mathbf{L}_3$, and (b) revolute RRR - the first three joints of revolute type ($\theta_1,\theta_2,\theta_3$), (ii) optimal division of the unit length between both links in RRR option ($\mathbf{L}_1$ and $\mathbf{L}_2$, where $\mathbf{L}_1+\mathbf{L}_2=1$), and (iii) determination of the joints' movement ranges. In the revolute configuration the amount of variables is raised to four, to include the split in total link length for both forearm and elbow, therefore, the assessment of the outcome needs special consideration. The choice of the two configurations for this example is governed by the nature of the NDT and restoration tasks and the form and context of the transformation matrices, so that the crossover would not violate the logical structure of the chromosome. The number of variables in the transformation matrices depends on the type of configuration, thus the length of the representation (chromosome) and the number of genes would differ for these configurations. As the crossover operation involves exchange of genes referring to the same parameters, with a number of genes varying, the genes relating to different parameters would be swapped and the integrity of the process violated.

The parameters at this stage are optimised, based on the criteria of singularity and collision avoidance, and percentage coverage. These three criteria are mainly influenced by the values of the parameters selected at this stage, so splitting the process into two stages does not violate the merit of simultaneity. Considering collision avoidance and percentage coverage, therefore, it is possible to combine these two criteria using the same representation. Additionally, the two-stage approach does not violate the simultaneity of the process, as the values of the parameters from the first stage are refined in the second, and 'full' optimisation of additional parameters is performed at the second stage.

Each chromosome contains set of genes. Each gene contains a parameter value. Chromosome, in turn, represents indirectly (direct kinematic calculations are required) a point in 3-D through a set of joint variables (configuration). A string of chromosomes describes the sequence of the tool locations (using direct kinematics), which is the followed trajectory. The closer the representation (string of chromosomes) approaches the planned path (trajectory) on a bridge surface, the higher its evaluation value (fitness). The optimisation here is based on minimising the distance between the trajectory followed by the model (represented within the strings of chromosomes) and the pre-set path on the bridge, and the ability to closely follow a number of trajectories located in various parts of the workspace.

In this work the following notation is adopted: $\mathbf{T}_{3i}$ (x,y,z point co-ordinates) (i=1,...,N) is a matrix specifying a trajectory of N points, represented by a series of locations of the tool with respect to a global Cartesian reference frame. $\mathbf{E}_{3i}$ (i=1,...,N) is a similar matrix, but specifying the locations of the elbow while following the path, also with respect to a global frame. If the number of paths (M) is investigated simultaneously, to check the ability to perform various tasks without relocation of the mobile platform, the representation consists of $\mathbf{T}_{3i}$ and $\mathbf{E}_{3i}$ (i=1,...,NxM). $\mathbf{P}_{3i}$ (i=1,...,NxM) specifies N number of "ideal" points to be reached in each of the M paths in the workspace, marked in relation to the location of the base. A fitness value $\mathbf{f}_1$ is represented as the sum of the squared distances between $\mathbf{T}_{3i}$ and $\mathbf{P}_{3i}$ (to avoid offsetting by negative values). The minimum value of $\mathbf{f}_1$ indicates the best ability to reach all the points along the trajectories and awards the candidate the highest fitness.

At the same time, evaluation of collision avoidance is in-built into the fitness function. The bridge geometry under analysis, is represented as an obstacle with its contour marked as a series of points $\mathbf{C}_{3k}$ (k=1,....,K), with K being the number of points along the geometry contour, see Figure 8.1. The intersection between any point along the elbow ($\mathbf{T}_{3i}$ - $\mathbf{E}_{3i}$) and the surface of the obstacle adds a high penalty factor to $\mathbf{f}_1$ for a candidate solution.

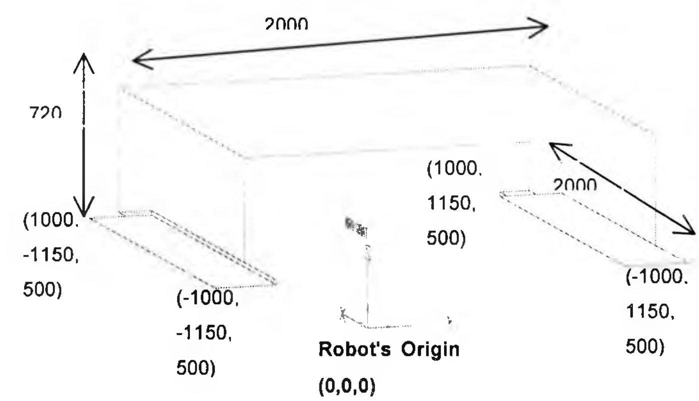


Figure 8.1. Collision Envelope Isolated from the Unit Workpiece

8.3.2 Stage Two

The optimisation at this stage uses all the parameters from the previous one, except that it includes only the 'fitter' configuration. The formerly optimised parameters from the first stage, namely link lengths and the first three joints' working ranges are admitted to this stage. Their optimal values are used in evaluation calculations and further refined in this stage. The allowance for additional optimisation of the first stage parameters is provided in the form of the oscillation bandwidth. It is assumed that the optimal values from the first stage are allowed to oscillate by 1deg up, twice, from the original value and 1deg down. The bandwidth of oscillation as well as amount of steps depends on the motors' characteristics. The additional parameters are: (i) two wrist configurations, (ii) minor axes joint working ranges, (iii) all joint velocities and accelerations. The criteria by which they are optimised are: (i) dexterity, (ii) productivity, and (iii) wrist singularity avoidance.

The choice of wrist configurations is governed by their applicability and versatility, and concentrated on RPY (roll, pitch, yaw) and RPR (indicating an Euler wrist performing roll, pitch, roll motion). The evaluation of the wrist type arises from compatibility of the orientation of the standard ortho-normal bases of all points along the trajectory, measured in relation to the base co-ordinate frame and the orientation of the orthonormal basis of the wrist TAP (tool attachment point) calculated in relation to the same co-ordinate frame. Wrist orientation is assessed analysing the rotational part of the transformation matrices within the kinematic calculations. The unit vectors of the ortho-normal bases (already oriented in the tool/task compatible position) of the points along the required trajectory $\mathbf{P}_{3i}$ are projected onto the orthonormal basis of the robot's base frame. The 3x3 matrix $\mathbf{R}_{ij}$ represents the transformation matrix from the co-ordinates of $\mathbf{P}_{3i}$ with respect to the global frame to the co-ordinates with respect to the robot's base frame. The rotational part of the final transformation matrix of the robot's full 6 DOF configuration $\mathbf{A}_{ij}$ already projects the unit vectors of the wrist onto the same robot's base frame. Compatibility of the equivalent column vectors between the matrices results in matching orientation.

8. Model Representation

The optimisation of the robot aims at determining the major configuration, which is most suited to the task. The number of parameters varies for the two configurations, because in RRP the third joint, being prismatic, provides the reach, while in RRR all joints are revolute and the reach becomes the additional parameter and is embedded in the length of both links. Therefore, if RRR configuration is selected, the problem of allocating the optimum reach lies in the division of the total length of both links (here taken together as a unit) and is addressed as a unity divided between two links, which becomes an extra parameter. The number of independent parameters is three or four, the number of the computed parameters reflects also the number of points to be reached (here twelve). The representation constitutes thirty-six parameters for RRP (3 parameters x 12 locations) and thirty-seven for RRR configuration (3 parameters x 12 locations + link length), in order to cater for access to all twelve points along all four paths.

8.3.1 Coding of the Representation in Stage One

The representation is separately developed for both configuration types and, comprises twelve sets of encoded robot representations - four paths of three points each, of the first three joints' movement sectors and additionally for RRR configuration, a share in the unit length of the first link's length, as $\mathbf{L}_1 + \mathbf{L}_2 = 1$. Due to the different number of parameters, two separate computations have to be carried out for both configurations, and a preference of the major configuration established, based on the qualitative analysis of the representations with the highest fitness values calculated during their evaluation, within the GA procedure.

The first stage parameters are encoded as a single chromosome (Figure 8.2) for each location, in a binary representation. The chromosome is built as M number of the sequences of N genes and represents M required trajectories, following N points each. The number of paths (M) and their location and orientation describe the sought level of versatility, or otherwise their ability to get close to the boundaries of the workstation within the bridge geometry.

The continuous ranges of parameters are divided into sets of discrete values, as continuous variables are difficult for GA to handle. To solve the problem the range is divided into 64 (including 0) discrete cut-off values, (numbers 0 to 63 serve as a type of shorthand for each specific numerical cut-off value), which provides a step of 5.625^{deg}. For the joint ranges, the search space is discreticised and represented by 3 integers (one for each joint), each belonging to (0,....,63) in a binary representation. Boundaries of working ranges are assessed, based on the working environment dimensions and practical motor considerations (here, initially 360°).

Arriving at the optimal solution means, in practice, analysis of GA performance and level of convergence and terminating the optimisation process at the 'satisfactory' level of the greatest fitness. This generates the boundaries of working ranges of the first three joints.

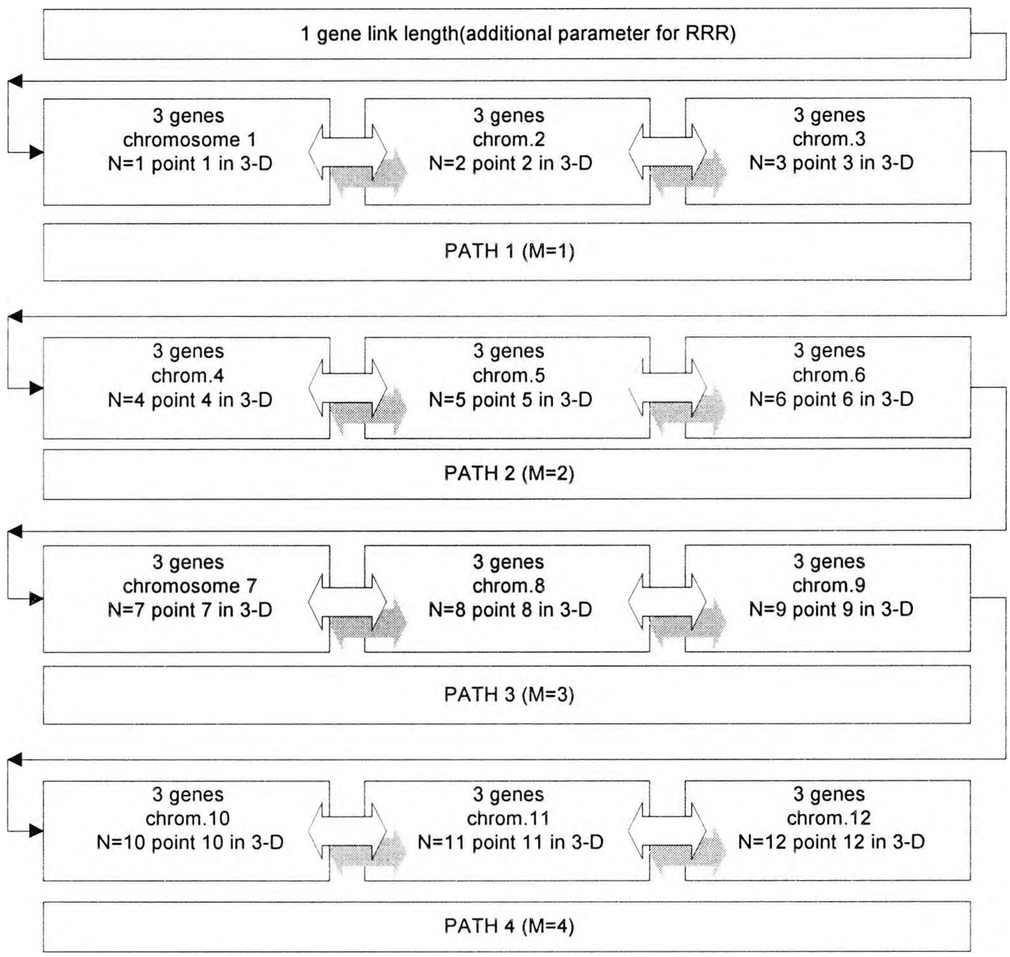


Figure 8.2. Stage One String = Representation

8.4.1.1 Stage One Chromosome Contents

The search space is sized as follows: 6 digits represent the link length (in case of computations for RRR only), and the joint angle range is represented by 3 times 6 digits. The length of the second link is set automatically due to the requirement for the arm to be of a constant length. Each chromosome is encoded by 18 bits. The first digit specifying the configuration is located at the beginning of the string, as the type of configuration has to be consistent throughout the whole of each trial, similarly to the link lengths which have to be constant (the same robot geometry). M=4 trajectories are analysed, each consisting of N=3 points, therefore the overall length of the string is 6+12x18 = 222 bits for RRR or 216 bits for RRP configurations.

8.4.2. Coding of the Representation in Stage Two

The second stage deals with the more suitable major configuration, emerging from the first stage and is devised to (i) determine the superior wrist type, (ii) specify working ranges for the wrist's joints, and (iii) determine the joint's velocities and accelerations, which in turn specify the type of motors, needed to evaluate the productivity criterion.

The optimal values of parameters for the criteria of maximum coverage percentage and collision and singularity elimination (from Stage One) may not necessarily comply with the values of the same parameter optimised for productivity, or dexterity. A solution to this problem is introduced in Stage Two. The cut-off values of all the parameters oscillate round the optimal values from the first stage.

An essentially similar coding scheme to represent the parameters into genetic strings is assumed, but a different approach to the criterion assessment appears in the second stage. However, coding of the previously optimised parameters from stage one alters. Integer number of possibilities per parameter, for optimal code efficiency is 2**, i.e. 2, 4, 8, 32, 64, etc. So the optimal values of the parameters from Stage One are augmented by 1^{deg} up twice, from the optimal value and down once to provide the oscillation boundaries and provide 4 options of the parameter value, according to the code capability. The refinement steps (here 1deg) as well, as amount of options is purely experimental and should be decided after characteristics of motors are investigated.

Two minor configurations chosen for the detailed analyses are: RPY and RPR, as described in Section 8.2.2. Also all joints' accelerations and velocities are incorporated into the representation. These constitute two hundred and eighteen or two hundred and seventeen parameters (the additional parameter is due to partition of the unit length, if the better configuration emerges as RRR from the first stage). The basic value is calculated by adding the choice of the wrist's configuration, twelve points times six joints' movement variables and twelve points times six joints' velocities and accelerations.

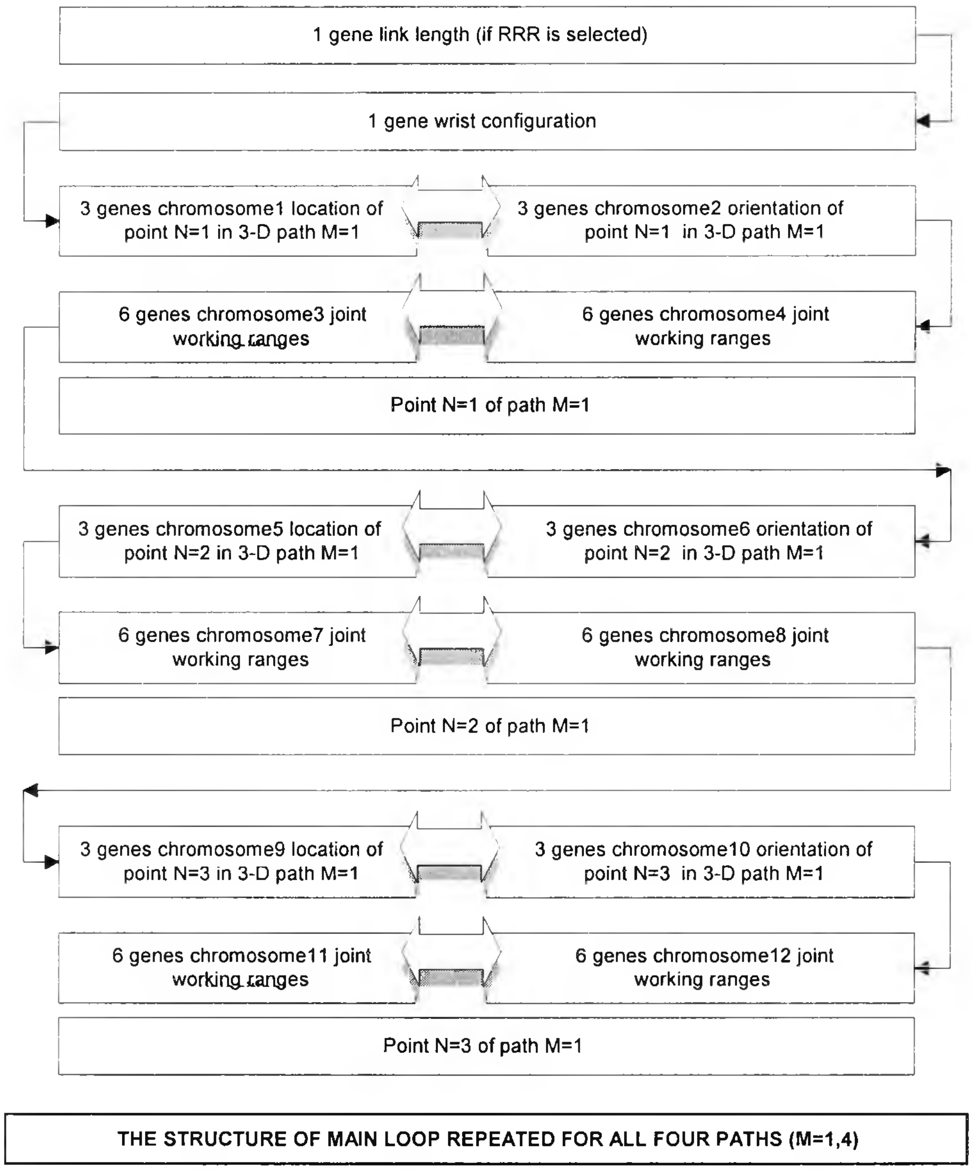


Figure 8.3 Stage Two String = Representation

8.4.2.1 Stage Two Chromosome Contents

The type of encoded representation at this stage consists of N times the following: (i) 6 digit string for the link length (if RRR configuration is the best option from stage one), (ii) one digit for the manipulator type, (iii) 3 times 4 digits for the first three joints of the 'fitter' configuration (determined at stage one), (iv) 3 times 6 digits for the manipulator joints' ranges, (v) 6 times 6 digits for the joint velocities, and (vi) 6 times 6 digits for the joint accelerations. N represents the number of points in a trajectory (here N=12). The string's length has therefore 6+1+12x{3x4+3x6+6x6+6x6} bits, which amounts to 1231 bits. The second criterion is evaluated for the same number and size of the paths as at the first stage.

8.5 Evaluation and Fitness

It is important to make the evaluation process execute as efficiently as possible because of the large number of times it is performed. GA use the fitness value of each string of the current generation to decide if and how many copies of the string should be passed to the next generation. The fitness value can never become negative.

8.5.1 Objective Functions

When optimising a function using GA, the objective function is typically the function being optimised. With the approach proposed in this thesis, the objective function is typically how well the generated form compares to some norm. But what is the objective function for a robot? The answer depends entirely on the problem for which a robot is being sought. If the problem is to design a mechanism that traces out a specified curve, the objective function might be how closely the generated mechanism follows the specified trajectory. For a vehicle, for example, the objective function might be to minimise the power consumed, normalised for vehicle mass and velocity. For many real systems, there is no single metric that is useful to defining the fitness of an individual.

Once the objective function (or functions) is identified, the means for evaluating the device must be determined. Engineering systems (with robots belonging to this category) can be grossly categorised into one of two classes: deterministic and stochastic. The difference between the two classes is not the robot itself, but rather how the robots are evaluated. Those devices classified as deterministic do not change their behaviour due to external disturbances while those devices classified as stochastic can modify their behaviour. Examples of deterministic systems include trusses and certain types of kinematic mechanisms (for example, single DOF mechanisms). Although, in both cases the systems might be subjected to time-varying inputs, the system cannot change, in response to these inputs. Examples of stochastic systems include biological and robotic systems. These systems typically utilise information about the environment in order to modify their behaviour in response to that environment.

There are two ways to evaluate a stochastic system: (i) create *a priori* list of all conceivable responses to a varying environment or (ii) incorporate the necessary means to permit the generation of adaptive responses. Since each system requires the designer to have developed a list of responses, the method may not produce novel concepts because only certain behaviours can be utilised. The incorporation of adaptive responses, called controllers, into the design of a system is the distinguishing feature of a stochastic (or a complex) system [Gen and Cheng, 1997].

The purpose of a controller is to permit the automatic evaluation of a system in which the input/output relationship is not fixed. An example of this, are the sequence of tool placement and size of the step between the tool moves for a restoration robot - it must respond to changes in the geometry. To evaluate such a robot, therefore, a generic tool-move planner is required. However, since the robot's configuration is not known a *priori*, its move planner cannot itself be designed *a priori*. Thus, the system's controller must be designed simultaneously with the system itself. Since the controller can typically be represented as a computer program, it can be generated in the same means as the system itself, genetically. For a system to perform well, it must therefore have an appropriate configuration as well as an appropriate controller.

The incorporation of an environment-adaptive controller means that the system needs to model the environment in some manner. This modelling requires the use of sensors. To achieve optimal performance from the system, it must have access to a range of potential sensors as well as the ability to integrate them into the model in an optimal manner. In the approach developed by the author, the role of the controller is taken by the precise geometrical modelling.

When using optimisation methods to maximise mathematical functions, it is imperative that the form of controller used, converges to the correct result. Holland [1975], De Jong [1992] and Goldberg [1989a] all show that GA do indeed converge to the optimal, or near-optimal, result for a wide variety of classes of functions, provided that the evaluation function contains some form of controlling mechanism.

One trap the designer must be aware of is that most systems are used over some range of operating conditions. When evaluating the system, if only a single operating condition is used, the system developed will certainly work well for that particular condition, but may fail in other conditions. To generate systems that are robust across a range of operating conditions, the designer should evaluate the system across the full range. This can be done by either testing each system for several different operating conditions or by changing the operating conditions for each generation of systems (one of the possibilities is to present noisy data to the system's sensors). Obviously, if the operating conditions are known precisely, then these known operating conditions should be used for the evaluation.

In the approach assumed in this thesis, the objective function includes the problem how closely the generated kinematic configuration is able to reach the pre-determined trajectory on the bridge. In order for the system to succeed in versatility, the objective function should test number of the bridge geometries for variety sizes and shapes of the collision envelopes, as well as, varied location of the trajectories. The selection of bridge details assembled in Figures 4.1-4.15 is to serve as 'a starting pack' to be tested in the objective function to achieve basic level of versatility.

8.5.2 Multiple Objectives

All systems with multiple objectives can be divided into two sets; those in which the various objectives can be combined in some manner to yield a single, representative evaluation and those in which the objectives cannot be combined.

Combining multiple objectives into a single value is typically done by assigning a weighting factor to each of the objectives and summing the result:

$$F = \sum_{i=1}^{N} c_i f_i \tag{8.1}$$

where, F is the overall fitness, f_i the fitness of objective i and c_i the weight assigned to this objective. This type of overall fitness is sometimes referred to as a single figure of merit (SFM) [Goldberg, 1989a].

There are several problems with using a SFM, including: (i) widely different characteristics, from easily quantifiable parameters like link lengths, for example, to subjective assessments of risk, are combined into a single weighted number, (ii) assumes the same degree of confidence for all characteristics, which is typically not true and (iii) choice of scores and weighting factors can be subjective.

Despite these difficulties, this method is frequently used. However, this approach does not work well for an automated design tool for two reasons [Buckley, 1988] and [Voogd, 1988]. Firstly, the designer must decide *a priori* how to assign the weights. Secondly, the systems under design may be quite different from each other therefore the issue of the validity of the weighting becomes important.

There is a second method which uses a rank based fitness assessment for multiple objectives, known as Pareto optimality [Goldberg, 1989], [Eschanauer, et al. 1990] and [Fonseca and Fleming, 1993]. This involves the comparison of the separate objective function vectors associated to each criterion with a view to identifying a non-dominated solution. The parameters associated with the non-dominated solution define the Pareto optimal design. Thus in the multi-criteria problems, not a single solution, but a set of answers is obtained which is non-dominated by others. For a two-objective problem, an example for Pareto optimal set is shown in Figure 8.4. By creating such a set, the designer is stating that it is impossible to combine the multiple objective functions into a single function and that all points in the Pareto optimal set are equally acceptable. In the research done on Pareto approach by Fonseca and Fleming [1993], in conjunction with GA, the solutions are ranked, based on their relative superiority to other solutions. For the limited dimension objective space, this method works well.

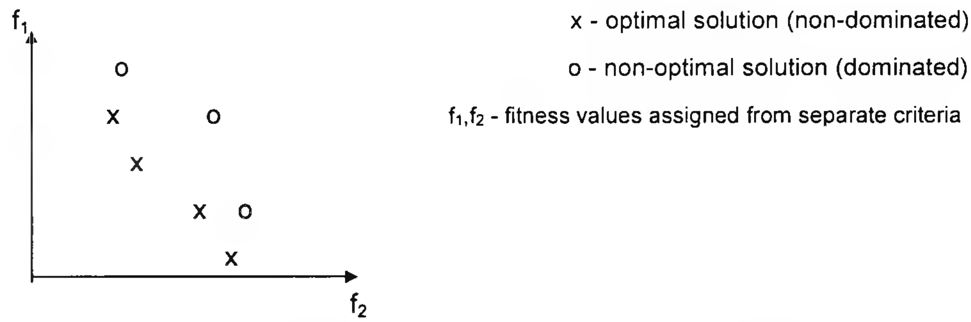


Figure 8.4 Two-objective Function Showing Pareto Optimal Set

If there are large numbers of objectives to be met, and the space of all possible solutions and systems is sufficiently rich, the large percentage of the population falls on the portion of the convex-hull formed by Pareto optimal set, which has a limited quality for the designer. The designer must ultimately make a value judgement among the alternatives to arrive at particular decision. Another possibility mentioned in the literature is the use of a distance in Pareto space, as a single valued functional valuation [Eschanauer, et al. 1990]. Subjectivity is introduced with this method because the designer must specify the point from which the distances are measured. On many occasions the origin can be specified with little prejudice. After reporting on two approaches to multi-criteria optimisation, which occurs in the second stage (time and dexterity), the author preliminarily selects the coefficient method, however the final approach depends on the particular optimisation environment. The Pareto approach to the multi-objective functions is presented here, although not utilised in the software, because the SFM approach instinctively feels as producing superficial results. The author included the SFM option in her software, however commented it out, as the coefficients as well as ranking depends on the specific design circumstances.

A separate problem emerges when assessing the values of the fitness. At both stages the preferable solutions will receive numerically smaller fitness values, therefore a scheme needs to be employed which changes these small numerical values into higher fitness and employing the sum of reciprocals seems to produce satisfactory results.

8.5.3 First Stage Evaluation

The evaluation at the first stage is based on direct kinematic calculations, providing series of positions of the end effector relative to the base frame. Fitness is assigned to each member of the population (each string), based on a combination of the ability to follow closely the pre-set number and type of paths, as well as collision avoidance. The aspect of reaching points in pre-determined sequence is not addressed here. The distance between path points and tool position is calculated between the nearest points irrespective of their sequence, therefore a short routine of distance assessment has to be introduced, prior to the distance calculations. The closer the set of configurations brings the tool to the desired goal without collision, the better its rank will be. At the first stage of the optimisation, a fitness function is defined as follows:

$$f_1 = \sum_{i=1}^{NxM} \frac{1}{(T_{3i} + P_{3i})^2 + I} \tag{8.2}$$

where,

$$I \begin{cases} = 0 & \Rightarrow \ (T_{3i} - E_{3i}) \cap C_{3k} = 0 \quad \text{or} \quad (T_{3i} - E_{3i}) < C_{3k} \\ = \infty & \Rightarrow \ \text{otherwise} \end{cases}$$

(The variables are explained in Section 8.2 and defined in 'Notation and Variable Definition' section at the beginning of the thesis).

8.5.4 Second Stage Evaluation

At the second stage, to evaluate the best individuals in the population, two aspects relating to the criteria of dexterity and productivity have to be addressed. The time of every attempt to reach every location out of **NxM** number of points is attributed to the longest time $\mathbf{t_i}$ required moving each out of all the six axes. The representation with the smallest value of the previously calculated longest times receives the highest fitness. The distances to the points are not an objective, as they are selected from $\mathbf{P_{3i}}$ at the first stage. Trapezoidal velocity profile is specified for the motion. Part of the fitness function relating to the productivity (time) at this stage is assumed as follows:

$$f_{2a} = \sum_{i=1}^{NxM} \frac{1}{t_i} \tag{8.3}$$

A separate, third component of the fitness function addresses the dexterity. The rotational parts of the transformation matrices are assessed for the final orientation of the wrist. This orientation is displayed through the final transformation matrix $\mathbf{A_{ij}}$, where the column vectors (n, s, a) in Eqn. (8.4) represent the projection of the co-ordinates of the wrist's unit vectors onto the robot's base frame. All points along the path $\mathbf{P}_{3i}$ must also have allocated unit orthonormal vectors representing the required tool orientation. For example, a single point within the trajectory although having only three values (x,y,z) of co-ordinates indicating the location, can have an infinite number of unit orthonormal bases, specifying the required ways of approach.

$$T_0^6 = \begin{bmatrix} R_0^6 & d_0^6 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} n_0^6 & s_0^6 & a_0^6 & d_0^6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8.4}$$

The columns of the matrix $\mathbf{R}_{ij}$ indicate the projection of the co-ordinates of the path point's unit vector onto the robot's base frame. The representation of the greatest fitness will have the maximum value of the squared reciprocal of the differences between the relevant column vectors of both matrices:

$$f_{2b} = \sum_{\substack{i=1 \\ j=1}}^{NxM} \frac{1}{(A_{ij} - R_{ij})^2} \tag{8.5}$$

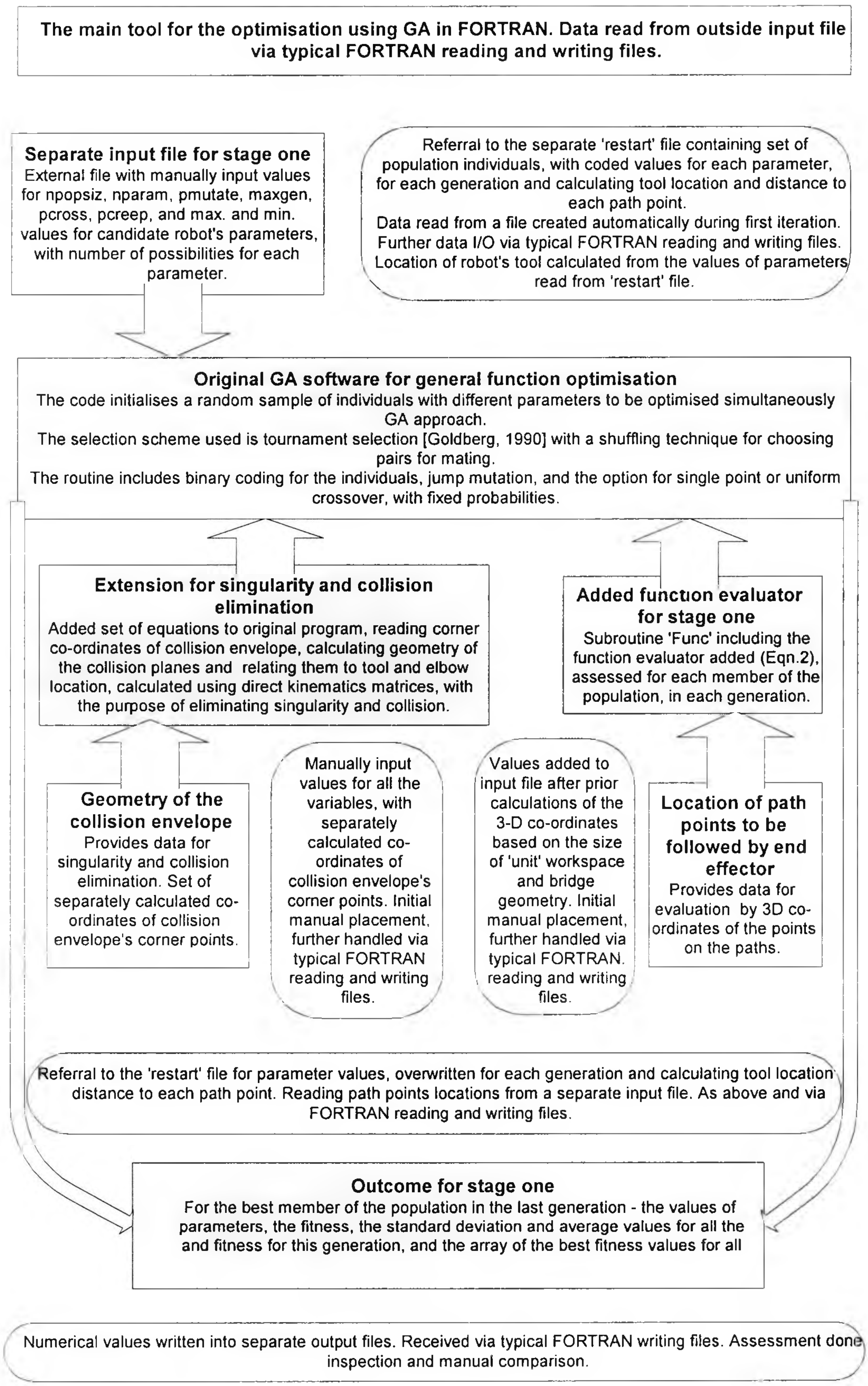
The total evaluation function will consist of two components:

$$f_2 = A^*f_{2a} + B^*f_{2b} \tag{8.6}$$

where, A and B are 'weighting' factors assigned, depending on the relative importance of productivity against dexterity. Due to the fact that assuming certain 'weighting' factors, without any logical or practical justification, would only cloud the results, the author does not combine the results into f_2, but separates and comments on both f_{2a} and f_{2b} individually.

8.6 Computing Tool Development

The program for this specific optimisation is a FORTRAN version of a genetic algorithm driver [Caroll, 1996]. This code initialises a random sample of individuals with different parameters to be optimised using the genetic algorithm approach, i.e. evolution via survival of the fittest. The GA used is a 'simple' genetic algorithm, which uses non-overlapping populations. In each generation the algorithm creates the whole new population of individuals. When simple GA is created, a population of individuals is specified, the new GA will clone the individuals that are previously identified, to make its own population. Elitism is optional, but should be turned on, as it means that the best individual from each generation is carried over to the next generation. The selection scheme used is tournament, with a shuffling technique for choosing random pairs for mating, which represents mixed sampling mechanism. The routine includes binary coding for the individuals, jump mutation, creep mutation and the option for single-point and uniform crossover. Two methods of hybridisation are added to improve performance and reduce the computing load. Niching (sharing) and an option for the number of children per pair of parents (to enlarge sampling space) have been added. An option to use micro-GA is also added, in order to significantly reduce the number of computations. The standard subroutines for: (i) randomising, (ii) selection, (iii) binary coding and decoding and (iv) parameter adjustment remain for all the trial runs and verification. The existing driver had a subroutine "func" (for optimising simple mathematical function) substituted by the author's function evaluators, for each stage and the verification approach. The whole process can be presented, as a series of blocks, containing description of modules, their role and function, method of calculating, type of data and ways of transferral between the modules. This is shown in Figure 8.5 (parts A and B, for stage one and two, consecutively).



The main tool for the optimisation using GA in FORTRAN. Data read from outside input file via typical FORTRAN reading and writing files.
Separate input file for stage one
External file with manually input values for npopsiz, nparam, pmutate, maxgen, pcross, pcreep, and max. and min. values for candidate robot's parameters, with number of possibilities for each parameter.
Referral to the separate 'restart' file containing set of population individuals, with coded values for each parameter, for each generation and calculating tool location and distance to each path point.
Data read from a file created automatically during first iteration.
Further data I/O via typical FORTRAN reading and writing files.
Location of robot's tool calculated from the values of parameters read from 'restart' file.
Original GA software for general function optimisation
The code initialises a random sample of individuals with different parameters to be optimised simultaneously GA approach.
The selection scheme used is tournament selection [Goldberg, 1990] with a shuffling technique for choosing pairs for mating.
The routine includes binary coding for the individuals, jump mutation, and the option for single point or uniform crossover, with fixed probabilities.
Extension for singularity and collision elimination
Added set of equations to original program, reading corner co-ordinates of collision envelope, calculating geometry of the collision planes and relating them to tool and elbow location, calculated using direct kinematics matrices, with the purpose of eliminating singularity and collision.
Added function evaluator for stage one
Subroutine 'Func' including the function evaluator added (Eqn.2), assessed for each member of the population, in each generation.
Geometry of the collision envelope
Provides data for singularity and collision elimination. Set of separately calculated co-ordinates of collision envelope's corner points.
Manually input values for all the variables, with separately calculated co-ordinates of collision envelope's corner points. Initial manual placement, further handled via typical FORTRAN reading and writing files.
Values added to input file after prior calculations of the 3-D co-ordinates based on the size of 'unit' workspace and bridge geometry. Initial manual placement, further handled via typical FORTRAN. reading and writing files.
Location of path points to be followed by end effector
Provides data for evaluation by 3D co-ordinates of the points on the paths.
Referral to the 'restart' file for parameter values, overwritten for each generation and calculating tool location distance to each path point. Reading path points locations from a separate input file. As above and via FORTRAN reading and writing files.
Outcome for stage one
For the best member of the population in the last generation - the values of parameters, the fitness, the standard deviation and average values for all the and fitness for this generation, and the array of the best fitness values for all
Numerical values written into separate output files. Received via typical FORTRAN writing files. Assessment done inspection and manual comparison.

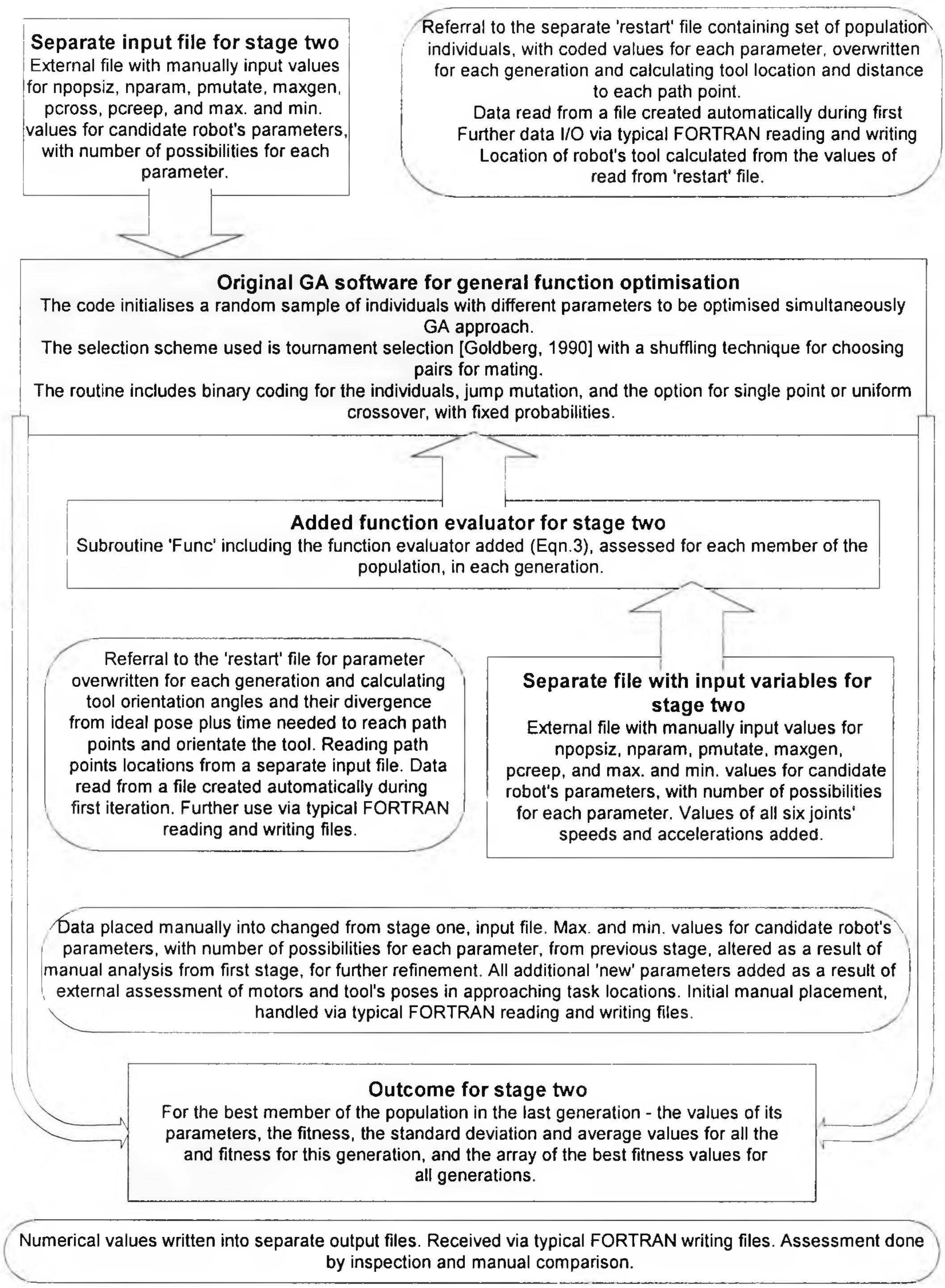


Figure 8.5a and Figure 8.5b. Modules and their Roles, Function and Interaction for Stages One and Two

The driver, as introduced by Carroll [1996], has many characteristics which guarantee more flexibility to available options: (i) input is in the form of minimum and maximum values of the parameters, with the number of possibilities for each parameter and (ii) specific parameters can be selected for niching. Appendix D gives the detailed algorithm with the distinguished parts, which are added by the author. It also shows the schematic diagram of the data flow. Appendix E gives the full version of the software listing.

8.7 Robot Simulation Software

The author felt that although, the developed method for kinematic parameters optimisation using GA is going to be verified on an independent robot, in the following Chapter, additional confirmation of the feasibility is needed. Therefore, she has investigated the commercially available robot simulation software and selected GRASP (from BYG) to model the optimised robot and re-check the feasibility of the optimisation (in Chapter 10).

GRASP (Graphical Robot Applications Simulation Package) is an interactive simulation package which models industrial robot arms and their workplace. It is development of SAMMIE CAD, a geometrical modeller, augmented by facilities such as a data base of kinematics models for different model configurations and robot operating systems [Dooner, 1983]. In GRASP, an independent entity named 'robot' can be defined which is a chained linked structure, stationary on one end (at the base of the robot arm) and open at the other (the tool tip). The structure and configuration of the robot is defined in two parts, the flesh of the robot and its kinematics. The flesh is in fact the bodywork of the robot, i.e., the geometrical solid model of the links, gripper and the base. The functional properties of the robot, such as the number and type of the joints, their relative position, and the kinematics constraints are collectively the kinematic model of robot. The software includes a 3D solid modeller for modelling the flesh of a robot and other objects in its workcell (workspace). The modeller can be used either interactively or through a descriptive high level programming language. Each object in the work place can be programmed to move. The motion of the objects can take place in parallel and they may be synchronised using variables and flags. In addition, the robot can also be instructed to drive as a single unit in joint mode, world mode and onto an object.

GRASP is mainly a kinematics simulation package and the dynamics of a robot are not modelled. The simulation of the robot, therefore, does not fully illustrate the operation of the real machine. In GRASP all the entities in the WORKPLACE can be programmed to move using the TRACK facilities. A separate track can be specified for each object and tracks can be synchronised and run simultaneously in a pseudoparallel environment. The operation of the tracks can be synchronised either by the clock built into GRASP or by using variables and flags. The track program can be entered interactively by defining the next location of the objects at each step. Alternatively, the high-level text programming language can be used to develop the tracks off-line. Similar to the assembly process, the development of a group of operational tasks to animate the movement of the robot is a lengthy and elaborate task. Nearly all the advanced techniques provided in GRASP have to be exploited to produce a satisfactory outcome.

8.8 Summary

This chapter has shown how the optimisation tool, GA, has been applied to the robot's optimisation problem (Chapter 7). Two stages have been introduced into the optimisation due to the characteristics of the criteria and computing limitations. The separate parameters and related criteria have resulted in simplicity of the outcome analysis and greater speed of assessment.

Using GA for the optimisation of the robot's parameters for selecting the first 3 DOF concentrated on selecting optimal out of two configurations, similarly in the second stage the last 3 DOF have also been selected out of two options.

Joints' working movement ranges related to the first 3 DOF have been optimised in the first stage, according to criteria of singularity and collision avoidance and percentage of coverage. In the second stage, wrist's joints' working ranges have been optimised according to singularity avoidance and dexterity. Additionally, the velocities and accelerations of all 6 DOF have been optimised for the maximum productivity.

Throughout the GA application development the main goal has been to come up with such a form of representation and evaluation, that finds 'reasonably' good (if not perfect solution) in a reasonable amount of time. This has governed the number of paths and number of points along those paths to follow by the end effector, as well as, the size of the step between the subsequent discrete values of the parameters. As the size of the transformation matrices and the length of the chromosomes both depend upon the configuration, two separate computations have been carried out for RRR and RRP configurations.

After the GA application method has been developed, the software (in FORTRAN) has been designed to reflect this. A general purpose FORTRAN encoded GA driver was further developed to suit the problem and the evaluation conditions. The fitness calculation functions and setting up environment for singularity and collision avoidance were added.

Then the robot simulation software was selected (GRASP), in order to carry out additional feasibility check at the later stage, after the optimisation results have been obtained.

This chapter, therefore, has presented the GA framework fully developed and ready to perform. The developmental part of this research is completed.

The following chapter tests the approach on the previously optimised (using alternative tools) configuration, for the method verification.

PART THREE

RESEARCH VERIFICATION, IMPLEMENTATION, EXPERIMENTS AND CONCLUSIONS

CHAPTER 9 - GA ALGORITHM VERIFICATION

9.1 Introduction

Now that the solution to kinematic optimisation of the robot has been worked out, as shown in Chapters 7 and 8, and the prospective tool has been analysed and applicability assessed, implementation is to be carried out. As a new and original method has been developed it has to be applied to a model, to which the solution is well known, or arrived at using the alternative methods.

Therefore, before this methodology can be introduced to the case study (robot restoration for steel bridges), the rate of success and the viability of the method have to be verified on already optimised robot. This is carried out on the example of the Multi-Purpose Interior Robot (MPIR), which was developed at Technion, Israel Institute of Technology by Dr. R. Navon, optimised using logical elimination and ROBCAD robot simulation software. Similarly, the kinematic parameters of MPIR were optimised, so it becomes straightforward to model the MPIR and the workspace and test the method, by comparing the results. The whole approach to verification using the methodology introduced in Chapter 7 and 8, has to be customised and modelled to suit a different problem (similarly to a case study in Chapter 7), which serves as a preliminary verification case study.

The original data about MPIR optimisation is included in Appendix G, in the form of photocopied journal papers reporting on the development.

9.2 Approach to Method Verification

The Multi-Purpose Interior Robot (MPIR) was developed in the laboratories of Technion, Israel Institute of Technology. The choice of best robot and optimisation of its parameters is aided by innovative usage of a commercial graphic simulation package. MPIR is designated to perform interior tasks, such as: (i) building partitions, (ii) plastering or painting walls and ceilings, (iii) finishing of floors and (iv) joining of structural components (walls, partitions and slabs).

The robot's development and optimisation is well documented in numerous publications: [Navon, 1989], [Navon, 1990b], [Warszawski and Navon, 1991] and [Navon and Warszawski, 1992]. The author takes the view that if MPIR's optimisation can be confirmed using GA, then the innovative use of GA (as introduced in Chapter 7) can provide credible results, when adapted to the research case study, a robot for restoration of steel bridges.

Confirming optimal values of common parameters of MPIR and bridge restoration robot, provide a good basis for the GA optimisation approach, as the resulting values can be checked against the previously verified ones.

9.2.1 MPIR Original Design Optimisation Approach

The main objectives of the graphic simulation are: (i) testing of functional stability, (ii) comparing performance among different alternative configurations, and (iii) determining optimal values of robot parameters.

The main simulation parameters are: (i) configuration of the robot's arm, (ii) dimensions of arm, links and carriage, (iii) trade-off between arm reach and the number of workstations needed to complete a task for a specified space and (iv) the velocities and accelerations of the actuators.

The robot is employed from static workstations, the dimensions of the carriage are determined by width of the doorways and manoeuvrability in narrow spaces. The typical environment (floor geometry of the building floor) pre-determined the size of 'unit' workspaces. The payload and the self-weight are defined as by the result of a task characteristics' analysis.

The robot's reach is optimised against the following criteria: (i) productivity, (ii) number of workstations and (iii) cost.

Simulation results are obtained from investigating two representative activities, masonry and painting. Robot configurations are reduced to two competing ones, spherical and articulated, both with three revolute minor axes (wrist). After analysing the volume of the effective work envelope, based on a floor (fully dimensioned) plan of a proposed building, shown in Figure 10 of Warszawski and Navon [1991] (Appendix G), the articulated robot is identified as optimal, producing the largest effective work envelope and therefore reducing the number of necessary workstations within the floor of the building. The arm length (or reach) is assessed for seven alternative values 1.6m, 1.8m, 2.0m, 2.3m, 2.5m, 2.7m and 2.9m. The criteria for the above are: (i) productivity, (ii) number of workstations, (iii) collision avoidance and (iv) percentage of coverage. The longer the arm, the more difficult (if not impossible) it is for the robot to manoeuvre in confined spaces, like small rooms. The productivity determined the reach of 2.3m as optimal for a variety of three-dimensional spaces that must be served by a single workstation location.

Actuators' velocities and accelerations are the parameters that have an obvious effect on the productivity. The actuators' acceleration affects the requirement for strength and rigidity of the links. The effect of these on weight and cost is also considered for MPIR. The balance between the velocities and accelerations (and the productivity, as a result) against weight and cost are considered for both point-to-point (PTP) and continuous motion using various velocity profiles. It is shown that, for masonry activity (representing PTP motion), the optimal range of the actuators' velocities for an articulated robot, with a reach of 2.3 m, is 100-200 deg/s. MPIR's productivity increases with the amplification of the actuators velocities, but reaches an asymptotic level around the mentioned range. Three velocity profiles are identified for experiments: (i) trapezoidal, (ii) trapezoidal with constant acceleration and deceleration, and changing velocities, and (iii) rectangular.

9.2.2 MPIR Optimisation Verification using GA Model

As the optimisation process for MPIR is analysed, it became apparent that optimal values for relating parameters can be confirmed using GA methodology. The whole GA approach is thus considered as corroborated, and can be employed in the optimisation of another applications.

The parameters from MPIR optimisation, which could be verified using GA, are the choice between spherical and articulated configuration, reach, and joint velocities and accelerations.

A representative model of standard dwelling with related workstation positions are analysed in order to establish the boundaries of the workspace and, at the same time, corners of the collision envelope. As stated in Warszawski and Navon [1991], the amount of workstations in a standard dewelling, with plan attached in Figure 10 of that paper, is six, and for this to happen, the robot had to be allocated centrally in each room. Three different size rooms (small, medium and large) are selected for trials. With the robot placed centrally, only half of each room is analysed (due to symmetry). The corners of the box-shaped workspace constitute the boundaries of the collision envelope and are input in the data file. Also, a robot's origin is placed 700 mm above the floor level due to MPIR's dimensions.

Then, two tasks are selected for MPIR's performance analysis. building a wall and applying surface coating. The activity of coating resembles steel surface cleaning, using blasting nozzle. The paths selected took into account the size of a typical spraying nozzle, its distance from the surface and the size of the area sprayed from a single nozzle position. All three workspaces, with the assumed paths for end effector to follow (shown schematically on Figures 9.1, 9.2 and 9.3 for small, medium and large workspace respectively) are 3D modelled, based on the description in Warszawski and Navon [1991].

Table 9.1. Path Points' Co-ordinates for Small Workspace

Path Points' Co-ordinates for Small Workspace							
Point	x	y	z	**Point**	x	y	z
P1	0	-1130	-680	**P10**	780	-727	1840
P2	390	-1130	-680	**P11**	780	-363	1840
P3	780	-1130	-680	**P12**	780	-20	1840
P4	780	-1130	160	**P13**	780	-20	1000
P5	780	-1130	1000	**P14**	780	-20	160
P6	780	-1130	1840	**P15**	780	-20	-680
P7	0	-1130	1840	**P16**	0	-20	-680
P8	260	-1130	1840	**P17**	260	-20	-680
P9	520	-1130	1840	**P18**	520	-20	-680

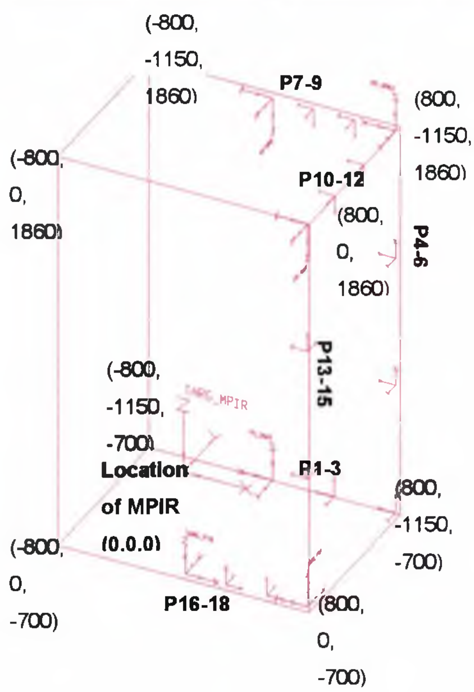


Figure 9.1 Small Workspace Model

Table 9.2. Path Points' Co-ordinates for Medium Workspace

Path Points' Co-ordinates for Medium Workspace							
Point	X	Y	z	**Point**	x	y	z
P1	0	-1280	-680	**P10**	1580	-1040	1840
P2	790	-1280	-680	**P11**	1580	-520	1840
P3	1580	-1280	-680	**P12**	1580	-20	1840
P4	1580	-1280	160	**P13**	1580	-20	1000
P5	1580	-1280	1000	**P14**	1580	-20	160
P6	1580	-1280	1840	**P15**	1580	-20	-680
P7	0	-1280	1840	**P16**	0	-20	-680
P8	526	-1280	1840	**P17**	526	-20	-680
P9	1053	-1280	1840	**P18**	1053	-20	-680

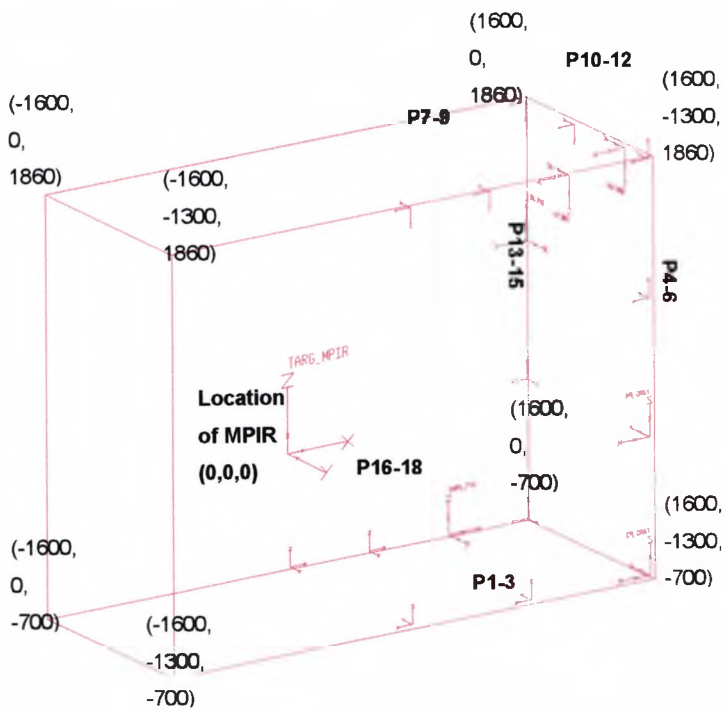


Figure 9.2 Medium Workspace Model

Table 9.3. Path Points' Co-ordinates for Large Workspace

Path Points' Co-ordinates for Large Workspace							
Point	x	Y	z	**Point**	x	y	z
P1	0	-1980	-680	**P10**	1580	-1326	1840
P2	790	-1980	-680	**P11**	1580	-673	1840
P3	1580	-1980	-680	**P12**	1580	-20	1840
P4	1580	-1980	160	**P13**	1580	-20	1000
P5	1580	-1980	1000	**P14**	1580	-20	160
P6	1580	-1980	1840	**P15**	1580	-20	-680
P7	0	-1980	1840	**P16**	0	-20	-680
P8	526	-1980	1840	**P17**	526	-20	-680
P9	1053	-1980	1840	**P18**	1053	-20	-680

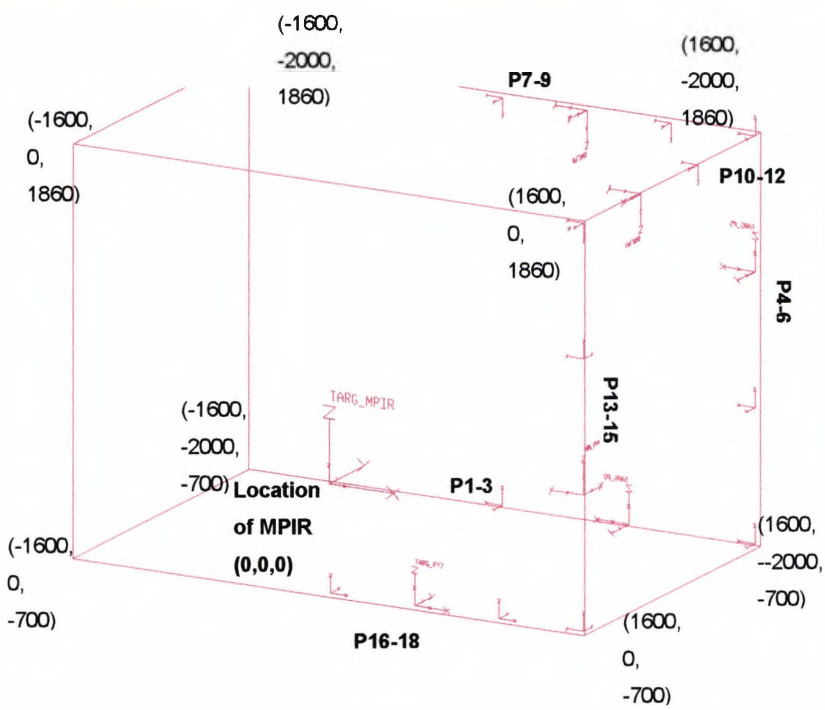


Figure 9.3 Large Workspace Model

Comparative trials are carried out in the same genetic environment (population size, number of generations and values of genetic operators). Restricted conditions for singularity avoidance are also employed throughout.

i. Comparison between spherical and articulated configuration.

Originally, the selection between the RRR and RRP alternatives is based on the spatial performance of the configurations and time required in executing the prescribed tasks. The extent of work that could be completed considering the arm's reach or the enclosed space constraints, is translated into boundary points of the collision envelope and purposefully placed points along the paths to be followed by the end effector. Applying the same conditions in the first stage of GA algorithm, as described in Chapter 7, for both configurations, the clearly higher fitness score is achieved by articulated configuration (RRR). The first stage measures the sum of distances between the position of the end effector (resulted from direct kinematics calculations) and the location of the path points (from task analysis).

ii. Reach for RRR.

The constrained conditions are set three-fold, small, medium and large workspaces, with relevant collision envelope co-ordinates and path positions. The same range of reach lengths as for MPIR are tested in all three environments. The fitness and performance values for each size of the workspace are plotted for each arms' length, to show the general efficiency of operations. As a final result, the best fitness values are added together from all three trials for reach. The largest value indicated the reach (through careful positioning of the path points including the extreme locations) which is the most efficient in all tested environments.

iii. Velocities and accelerations of actuators.

It becomes relatively straightforward to transfer the productivity / velocity relationship into GA model. Masonry activity comprises a number of relatively short movements and is imitated by the location of path points on the wall within certain area, as shown in Figure 9.4 (here approx. $0.2m^2$). The distance between the points of brick locations is assumed to be 150x150mm, and there is one location on the floor where the imaginary bricks are picked. So, the robot's tool movement is repetitive between 'Targ_BRICK' (Figure 9.5) and every point on the wall. Then, the overall productivity [m^2/h] can be computed. See Table 9.4 for input data and the outcome. The velocities are input at the exact values as in Navon and Warszawski [1992] (copy of the table is included in Appendix G).

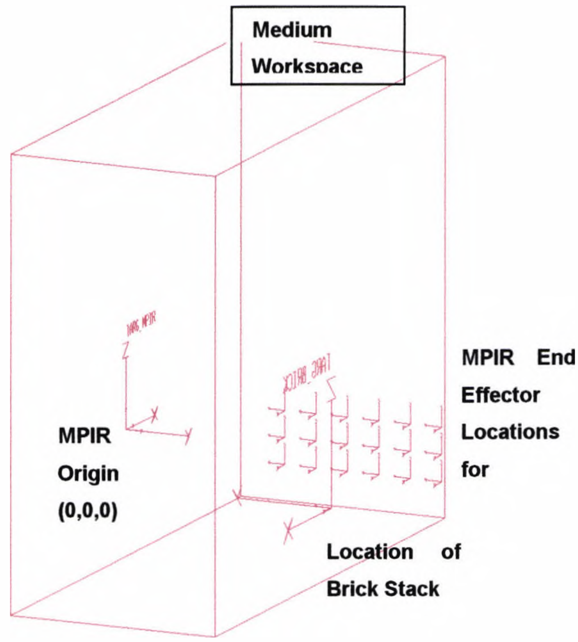


Figure 9.4 Workspace Model for Productivity Verification

9.2.3 Summary of Verification Procedure using MPIR

i. Comparison between spherical and articulated configuration.

The fitness performance and score for both configurations showed the clear advantage of the RRR configuration, for the same working environment and reach (here 2.0m) (see Figure 9.5).

ii. Reach for RRR.

As can be seen from the Figure 9.6, the longest reach is quite inefficient in small spaces (collision) and quite the opposite, the shortest reach does not reach far enough in the large spaces (too many workstations), as seen in the fitness behaviour in Figure 9.7. The medium size workspace is also tested for all arm lengths, and the lengths for the best performance as 2.0m, 2.3m and 2.5m (Figure 9.8). So they are isolated from the overall comparative check and analysed separately. The best performance is given by the robot's version with total reach of 2.3 in the medium size workspace (Figure 9.6), not only comparatively for all the other workspaces and arm lengths, but also from purely numerical value of the fitness function.

iii. Velocities and accelerations of actuators.

The profile of the graph from Figure 4 in Navon's paper [1990b] (see Appendix G for copy of the graph) resembles the shape of the graph in Figure 9.7. Additionally, the asymptotic behaviour of the productivity against raising velocities (above 200 deg/s) also becomes apparent, when considering results after implementing the same velocities and acceleration values. The productivity values are different because the geometrical details of the activity in Navon and Warszawski [1992] are unknown. The location of the robot, the details of the movement of the tool end-effector and the location for picking up bricks could not been confirmed. However, the overall trend is supported.

Table 9.4 Productivity for Alternative Velocity Profiles for 2.3m Reach Articulate Robot

Trial Number	Velocity [deg/s] according to Navon and Warszawski [1992] Velocities in () converted into [rad/s]	Acceleration/ Deceleration [deg/s^2] according to Navon and Warszawski [1992] Values in () are [rad/s^2]	Productivity [m^2/hr] according to Navon and Warszawski [1992]	Total Time for Activity in Figure 9.1d using GA [s]	Verification Productivity using GA [m^2/hr]
Productivity vs. Joint Velocity (Acc./Dec. 1/2 s.)					
1	500 (8.73)	1000 (17.45)	10.59	39.96	18.02
2	300 (5.23)	600 (10.47)	10.53	41.52	17.34
3	200 (3.5)	400 (6.98)	10.41	42.58	16.91
4	150 (2.62)	300 (5.24)	10.25	44.36	16.23
5	100 (1.75)	200 (3.49)	9.85	45.83	15.71
6	60 (1.05)	120 (2.09)	8.77	47.87	15.04
7	30 (0.52)	60 (1.05)	6.65	55.86	12.89
Productivity vs. Joint Velocity (Constant Acc./Dec.)					
1	500 (8.73)	220 (3.84)	10.14	44.20	16.29
2	300 (5.23)	220 (3.84)	10.14	44.36	16.23
3	200 (3.5)	220 (3.84)	10.14	44.64	16.13
4	150 (2.62)	220 (3.84)	10.10	45.03	15.99
5	100 (1.75)	220 (3.84)	9.90	45.95	15.67
6	60 (1.05)	220 (3.84)	9.09	48.06	14.98
7	30 (0.52)	220 (3.84)	7.04	69.77	10.32
Productivity vs. Joint Velocity (Rectangular Velocity Profile)					
1	500 (8.73)	infinite	10.64	37.48	19.21
2	300 (5.23)	infinite	10.63	37.68	19.11
3	200 (3.5)	infinite	10.61	37.84	19.03
4	150 (2.62)	infinite	10.55	37.88	19.01
5	100 (1.75)	infinite	10.33	41.79	17.23
6	60 (1.05)	infinite	9.45	47.62	15.12
7	30 (0.52)	infinite	7.19	58.02	12.41

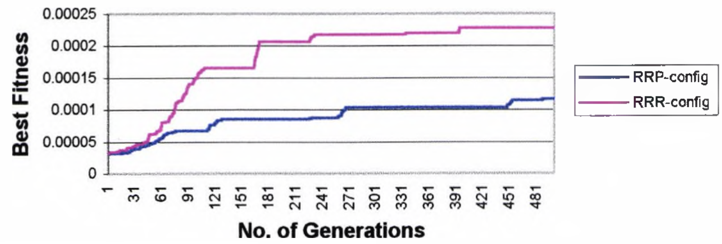


Figure 9.5 Performance of RRR vs. RRP Configuration for MPIR

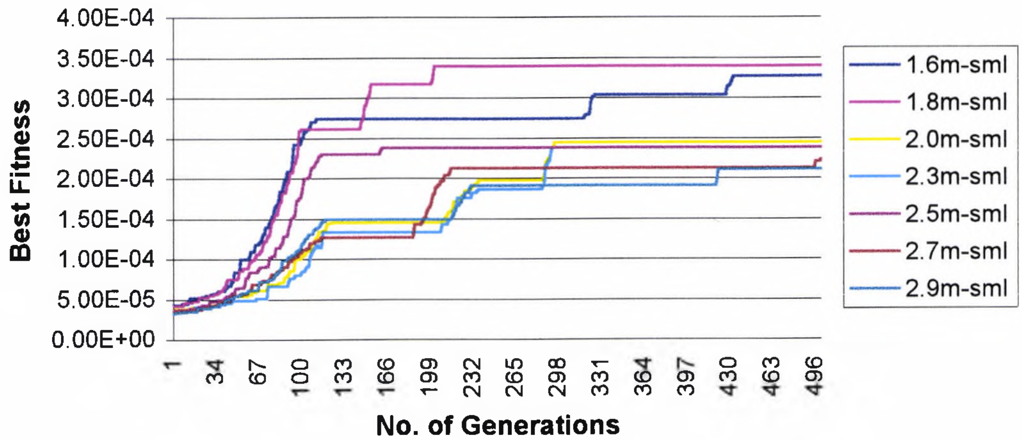


Figure 9.6 Performance of MPIR with Variation of Reach in Small Size Workspace

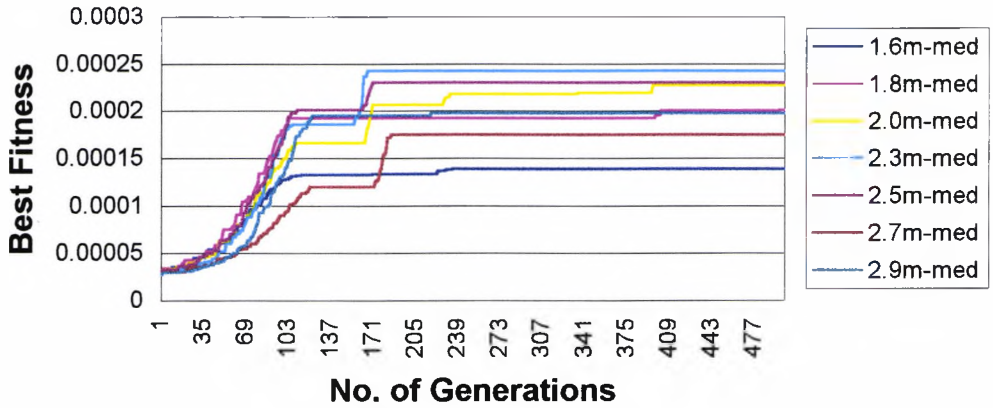


Figure 9.7 Performance of MPIR with Variety of Reach in Medium Size Workspace

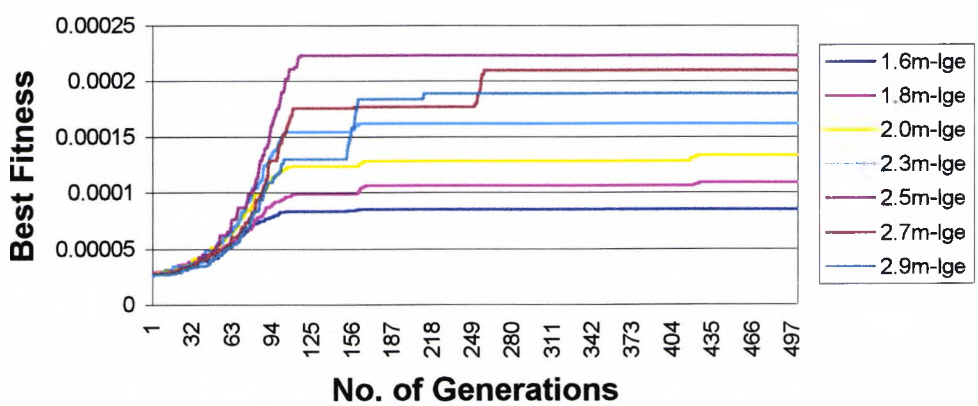


Figure 9.8 Performance of MPIR with Variety of Reach in Large Size Workspace

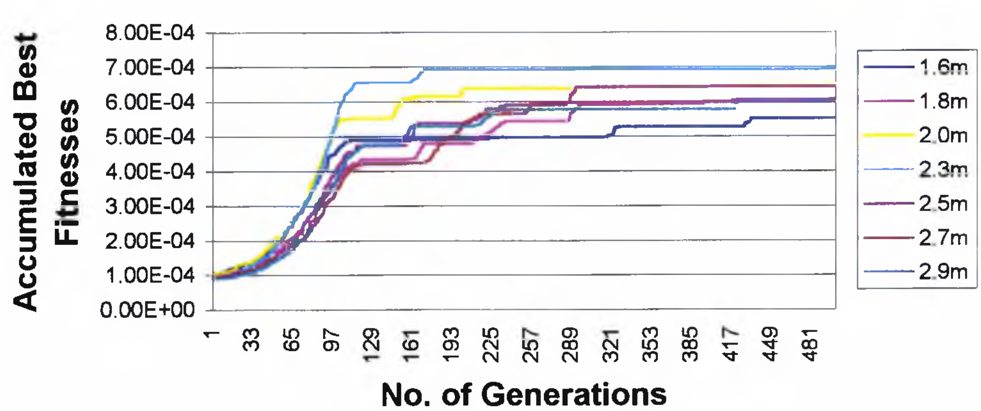


Figure 9.9 Accumulated Performance of MPIR for Variety of Reach in Different Size Workspaces

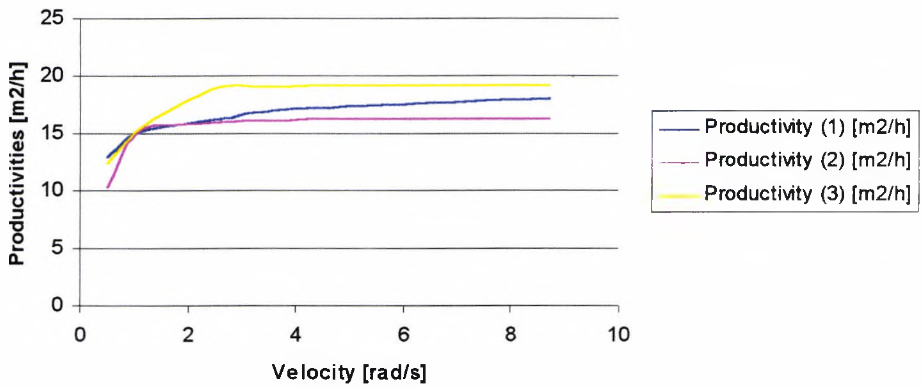


Figure 9.10 Productivity of MPIR with 2.3m Reach for Varying Velocity Profiles

As can be seen above, the robot's configuration and number of DOF, together with actuator's velocities, when GA optimised, provides similar outcomes as in the optimisation using the original computer simulation of MPIR by Navon and Warszawski [1992]. The parameters the original MPIR and GA modelled version, have in common, confirm the same behavioural tendencies and final outcomes. The preferred configuration is a six DOF articulated robot, which is also selected using GA. Additionally, it is found that the optimal range of actuator's velocities for an articulated robot with a reach of 2.3m is 100-200 deg/s, based on the assumptions of the robotic masonry activity. Therefore, it is assumed that sufficient verification is carried out that the GA method produces an optimal solution, hence additional parameters which the GA algorithm may optimise, will also arrive at optimal values. The further development of the optimisation approach is based on this finding.

9.3 Summary

This chapter has verified that the optimisation method developed for the bridge restoration robot and the GA application as an optimisation tool, are able to optimise the alternative case study robot and yield results comparable to those previously obtained using alternative optimisation methods and with well-confirmed results. MPIR from Technion has been selected as the verification case, for the similarity of final aim and well-publicised description of the process and outcome. Similar workspace to that of MPIR's has been modelled and the percentage of coverage and collision avoidance were addressed for three different size workspaces. The results confirmed that the optimal performance was obtained for medium size and this has minimised the amount of workstations within the workspace. The velocities and accelerations, where similar values were input also showed similar relative behaviour.

Once verification has been carried out, it became apparent that the method applied to the target case study (restoration robot for steel bridges) could generate useful results.

Therefore, the following chapter implements the proposed methodology to the case study using GA technique and reflects upon the outcome.

CHAPTER 10 - GA ALGORITHM IMPLEMENTATION TO THE CASE STUDY

10.1 Introduction

In the previous chapter, the author has applied the GA optimisation concept and computations developed in Chapters 7 and 8 to the well-developed and publicised construction robot (MPIR) from Technion, in order to confirm the results obtained using other optimisation methods. The optimisation of kinematic parameters of MPIR using GA, confirmed the results and indicated that the GA approach is viable and generates optimal or near optimal results. Therefore, it may be justifiably used in the selected case study for the steel bridge restoration to generate the optimal values of the kinematic parameters, for the given tasks, as described in Chapters 7 and 8.

This chapter covers the performance of the previously developed method for optimising the kinematic parameters of the robot, using GA as a tool, on the prototype assembled in Chapter 7. One of the bridges' workcells identified on steel bridges, given in Table 4.3 and shown in Figure 4.1, is selected. The task is added in the form of the required locations of the end effector. The tool selected for the trials is a blasting nozzle and locations of the paths reflect this tool operational requirement. Two minor and two major configurations are tried for superior suitability, with the kinematic parameters optimised for the same task. The results are assembled in the form of charts showing the behaviour of the fitness through the generations, and tables assembling detailed values of the parameters.

Designing a kinematically optimised robot for a given task, using established GA techniques, takes a lot of computing time. As the assessment of results and number of generations, population size, and values of the other genetic parameters are set up and adjusted 'on-line', several trial runs are needed. Therefore, further research into GA was required in order to make the whole approach more practical to use. The micro-GA significantly reduces running time and makes it more practical, and niching also improves the performance and, therefore, are researched and utilised. The alternative selection methods are also investigated. All the above are shown in the form of graphs, figures and charts. Then, analyses of the contents of the optimal fitness function, in terms of the optimal boundaries of all the parameters, shows the outcome on the arm's schematic representation of the optimised version of the robot. These results are then input into GRASP in order to further confirm the viability of the outcome. The fitting of the method into the ED process is then described, and all links within the process established. The optimisation results further confirmed by GRASP performance are assembled as the final and original findings of this research.

10.2 Solution Algorithm

Genetic Algorithm is a very accessible optimisation tool and has its effectiveness easy to assess, even at early stages, and in different ways. As suggested by Holland [1975], when searching large finite spaces, convergence is not the most useful performance measure, as there is always a danger that the optimum is not the global, but a local one. To avoid the search algorithm being entrapped in a local optimum, various methods are available, such as (i) improvements to the searching mechanisms, (ii) observing the speed with which the optimum is arrived at, (iii) analysing the efficiency of the fitness function with which it approaches the optimum or (iv) the analysis of the quality of the optimum solution at the intermediate stages.

As the problem under investigation is complex and involves searching large spaces, an approach of monitoring the performance throughout and analysing the current optima is adopted. From the typical form of the evolution curve [Bullock et al., 1995], it is evident that major improvements tend to occur during the early stages of searching. A maximum number of iterations must be chosen wisely, in case, the goal is too ambitious. The quality of the solution can be improved with more searches, but it is up to the designer to decide how long to wait. After running the computation for an initial 2000 generations, and observing the behaviour of the best fitness, it became apparent that no significant improvements happen beyond the first 400 generations (Figure 10.1) for almost a thousand more generations. This value is arbitrary and needs to be re-evaluated with every new evaluation condition. The improvement, which occurs here beyond 1500th generations, may be ignored on the initial assumption, that the result at 500th generations is "optimal enough", if the method has to yield quick and reliable results. However, before further investigations can take place, the values of the parameters at 2000th generations are compared with the ones obtained at the 500th generation. As the boundaries of the kinematic parameters remain the same and the improvement of the fitness must come from better utilisation of the parameters within the boundaries.

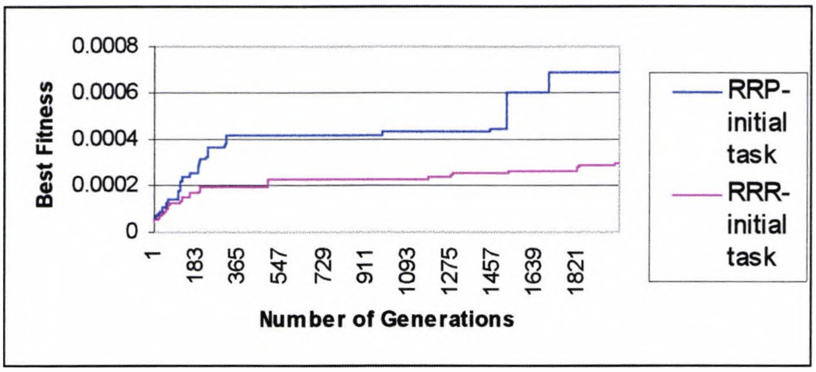


Figure 10.1 RRR and RRP Performance for 2000 Generations for Initial Task

The computations reported in this chapter are carried out for 'full' 2000 generations. The earlier debate about the cut-off level in the number of generations and the profile of the evolution curve is assumed to be useful when the problem under optimisation becomes more complex. Complexity may be induced through an elaborate task (many multi-location paths) or complex workspace geometry, when obtaining results in realistic computing time is of essence.

The 'best fitness' on the vertical axis (Figure 10.1) shows the numerical value of the performance evaluation of each generation. Each plot represents the best fitness for the combination of all the variables of the given stage. The scale of the computations restricts the simplicity and efficiency in obtaining results at this stage. Therefore, further study is carried out into the micro-GA [Krishnakumar, 1989] population and niching, which not only improves the performance but also significantly reduces the computing requirements.

Task description, as introduced in Chapter 7, involved geometrical positioning of the collision envelope on the chosen structure and locating the path for the robot's end effector. The approach to the underside of every bridge is considered to be difficult therefore, a bridge detail A1 from Table 4.1 is selected as a workspace and four paths of three points are located in a manner representing the task, as shown on Figure 10.2.

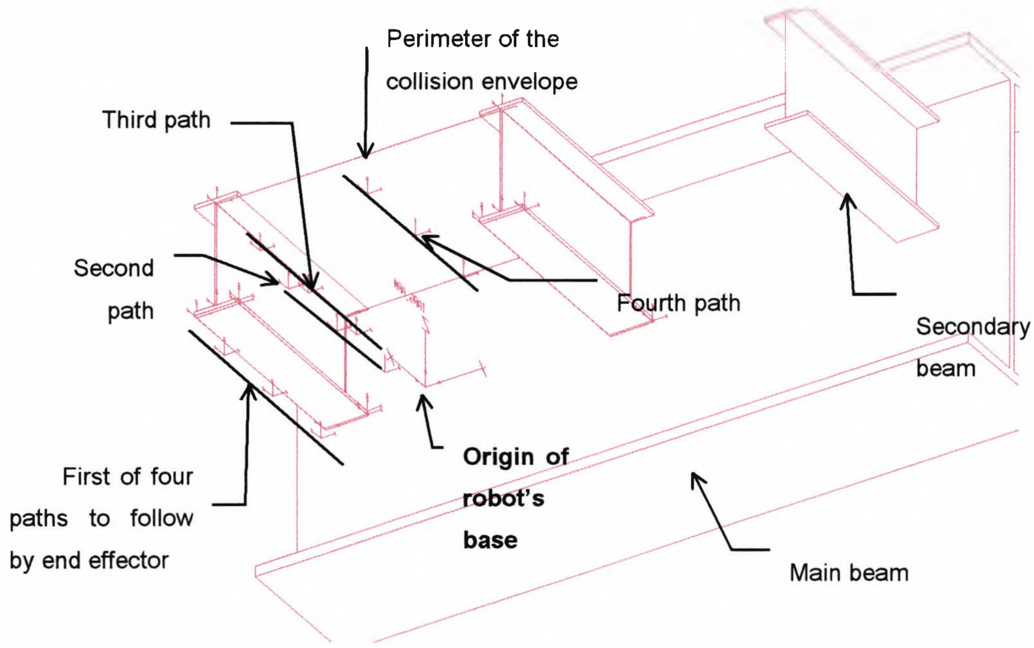


Figure 10.2 Graphical Task Description

10.2.1 Enhancement Techniques

Due to the extensive computation time, a number of different scale improvement procedures have been looked into for use in a genetic adaptive search. Their power varies from extreme to marginal [Goldberg, 1989]. The author felt that the action of two of them could enormously enhance the performance and reduces the computing time, so both niching and micro-GA [Haupt and Haupt, 1998] are investigated and applied in the search.

10.2.1.1 Niching

Niching methods promote the formation and maintenance of stable sub-populations in GA, and allow the GA to better cope with complex domains. Stable sub-populations of strings (species) serve different sub-domains of a function (niches) and creating niche-like and species-like behaviour can be practically used in the artificial genetic search. The reason for this is to discourage the ultimate convergence on the highest peak, without differential advantage and without identifying other peaks in the other regions of space. In other words, methods should be found which assure more fruitful mating patterns. Several methods have been introduced in order to induce niche formation in genetic algorithms, such as parallel niching (niches are formed and maintained within a single population) with crowding or through sharing function [Goldberg and Richardson, 1987], sequential niching and parallel hill-climbing. After careful consideration [Mahfound, 1995], the parallel niching with sharing is added to the original algorithm in order to improve the performance. Parallel niching methods conceptually form and maintain niches simultaneously within a single population. Sequential niching methods, on the other hand, locate multiple niches temporally.

In natural settings, sharing occurs through crowding and conflict, when a habitat becomes full of certain organisms and individuals have to share available resources. In artificial genetic search, sharing de-rates each population element's fitness by an amount related to the number of similar individuals in the population. The theory defines a sharing function (triangular) specifying the neighbourhood and degree of sharing for each string in the population, setting up the phenotypic (parameter set) sharing scheme. For a given individual, the degree of sharing is determined by summing the sharing function values contributed by all other strings in the population. Strings close to an individual require a high degree of sharing, while strings far from the individual require a very small degree of sharing. After collecting the total number of shares, an individual's de-rated fitness is calculated, by taking the potential, unshared fitness and divided by the accumulated number of shares. This limits the uncontrolled growth of particular species within a population. Upon convergence, local optima are occupied by a number of individuals proportional to their fitness values.

Conventional fitness sharing techniques [Deb and Goldberg, 1989] have been shown to be quite effective in preventing genetic drift, in multi-modal function optimisation, therefore, the basic GA algorithm is improved by implementing niching through sharing scheme with a triangular sharing function.

10.2.1.2 Micro-GA

If the functions for optimisation are well defined and do not change faster than the time it takes for GA to reach the optimum, then they are classified as stationary functions and 'simple' GA is used. On the other hand, there are many problems where the optimised functions evolve at a rate faster than the SGA can find an optimum, making the application and performance of SGA not immediately effective. An investigation by Goldberg [1989b] shows that for serial implementation of binary coded GA, the optimal population choice is small. Clearly, simply taking a small population size and letting them converge is not very useful, due to insufficient information processing and early convergence to non-optimal results. However, a scheme is put forward [Goldberg, 1989b], in which small population GA (micro-GA) can be implemented. Krishnakumar [1989] found that a micro-GA avoided premature convergence and demonstrated faster convergence to the near-optimal region than did SGA, for the multi-modal problems he studied. The procedure is as follows: (i) fix the population size as (here) N=100, (ii) randomly generate small population, (iii) perform genetic operations until nominal convergence, (iv) generate a new population by transferring the best individuals of the converged population to the new population (one good string from previous search - elitist strategy), and then generate the remaining individuals based on a deterministic tournament selection strategy, (v) perform genetic operations with crossover rate 1 (for high order of schema processing) and mutation 0 (to avoid more diversity), (vi) check for nominal convergence, or alternative termination conditions and (vii) go back to step (iv). The initial size of the population is purely empirical, against the traditional GA, with the population at 100 individuals. The termination conditions are set as 95% convergence for Stage One and 90% convergence for Stage Two, due to the size of the population and unnecessary loss of time, between 90% and 95%. Predictably, the micro-GA application substantially reduces computing time and allows easy monitoring of the individual developments, through generations.

10.3 Results of the Optimisation

10.3.1 Commentary on the First Stage Performance

Monitoring the output through the distribution of the best fitness only gives information about the speed and convergence profile towards the optimal solution. Assessment of the best individuals proves to be a better indicator. It is vital to notice that even tiny improvements in the representation's maximum fitness can bring significant changes in the values of the parameters, and vice-versa, the significant improvement in the best fitness value, may not necessary change the optimal boundaries of the parameters. Also, introducing standard deviation calculations gives additional information about the quality of the solution. Very often, the improvement in fitness is accompanied by an increase in standard deviation, calculated for the distances between tool positions and a pre-determined path, which may need further study confirming the quality of the best representation. However, first of all, careful analysis of the well performing individuals in the population in both computations for RRR and RRP configuration, has to determine the more suitable configuration for the task. The calculations for both configurations relate numerically, as the total link length in RRR and maximum value of the reach parameter in RRP are both 1.0, revolute joints having the same movement sector ranges, workspace and position of robot. Therefore, it is rational to compare both performances numerically.

The spherical configuration shows (Figure 10.1) significant and consistent superiority throughout, and the improvement in the fitness can be noticed already at an early stage. The RRP shows better performance and, therefore, is perceived as more suitable for the task. Initial verification, using computer simulation and inverse kinematics shows that RRP is clearly the better choice of the two, for the specified task. Then, the limits on the ranges of the movement sectors of the major configuration's joint variables are identified. This information not only helps to calculate the percentage of manual involvement, but also assist the kinematic design and choice of actuators. Hence, the RRP is the one, which is admitted to the second stage. As explained in Section 10.2, the number of generations is purely empirical. The improvement in the fitness logically is expected every time the micro-population is started. The level of improvement varies. The fitness is continuously improving and, with such a large problem rational decision about how much computation effort to expend in trying to improve the design and performance of a particular system, can be made only on the economic and obvious basis.

Table 10.1 Optimised Parameters for Initial Task in Stage One for RRP

Parameters (1)	Values (2)	Best Fitness (3)	Generation No. (4)	Standard Deviation (5)
J-1	40^{deg} - 326^{deg}	0.0003682	299	176.616
J-2	40^{deg} - 354^{deg}			
J-3 (prismatic)	(0.19-0.84)x2000mm			
J-1	34^{deg} - 326^{deg}	0.0004137	479	169.247
J-2	40^{deg} - 354^{deg}			
J-3 (prismatic)	(0.22-0.78)x2000mm			
J-1	40^{deg} - 326^{deg}	0.0004187	937	157.752
J-2	40^{deg} - 354^{deg}			
J-3 (prismatic)	(0.22-0.78)x2000mm			
J-1	34^{deg} - 326^{deg}	0.0004315	1302	164.713
J-2	34^{deg} - 354^{deg}			
J-3 (prismatic)	(0.24-0.78)x2000mm			
J-1	34^{deg} - 326^{deg}	0.0006040	1696	60.857
J-2	40^{deg} - 354^{deg}			
J-3 (prismatic)	(0.43-0.78)x2000mm			
J-1	34^{deg} - 326^{deg}	0.0006856	2000	56.074
J-2	40^{deg} - 354^{deg}			
J-3 (prismatic)	(0.44-0.78)x2000mm			

The last representation giving the maximum fitness is examined, and the boundary values of all three parameters outlined to show the movement sectors for all three joints, in order to determine the percentage of coverage. This information may additionally be used to determine the choice of motors and for the construction purposes. The information assembled in Table 10.1 is graphically illustrated in Figure 10.5.

10.3.2 Commentary on the Second Stage Performance

As the RRP configuration was identified as more suitable for the given task in stage one, it is admitted to the second stage with the optimised ranges of the parameters from the first stage computations (Table 10.1). Additional parameters include option for the choice of two wrist configurations (the Euler wrist (RPR) and PRY), the movement sectors of the joints within wrist, which in turn, can help the choice of the actuators and the geometry of the end effector.

The second stage representation uses the reduced boundaries of the joints' movement sectors for the first 3 DOF (four values within +/- 2^{deg} from the optimum), taken from the first stage and run also for 2000 generations, with similar singularity prevention restrictions as in the first stage. The total fitness function consists of two independent criteria and Pareto approach is not applied (due to insufficient data about productivity-dexterity polarisation).

As the size of micro-population is a very much trial and error assessment, a 2000 generation run was executed for population sizes npopsiz=100, npopsiz=200 and npopsiz=300, for dexterity. The results are as shown in Table 10.2.

Table 10.2 Trial Run for Micro-population Size Assessment for RRP Configuration in Stage Two

Population Size	**Best Fitness at 2000th Generation**
100	0.000000227272615542
200	0.000000227271832169
300	0.000000226764225678

The trial run shows that the population size of 100 is optimal, however the two factors influenced the decision of running the second stage optimisation for the population size of 200. Firstly, the amount of parameters increased from 36/37 to 217/272 and keeping the same population size could lead to premature convergence. Secondly, niching (crowding) needed bigger sample to perform satisfactorily and therefore, all the further trials are carried out for npopsiz=200. Section 10.4.1.1 offers further comments on the population size for micro-GA.

Therefore, it is beneficial to plot separately the best fitness performance for dexterity and productivity (see Figure 10.5 and Figure 10.6), for better ability to analyse the GA performance, instead of applying Eqn.7.6 and imposing artificially selected values of weighting factors A and B. Similar to the first stage, the performance of the best individual at the last computation is closely investigated.

Motion for the last (minor) joints after superposition of ranges fulfilling dexterity and productivity criteria is shown in Table 10.3. The working range for the last roll of the wrist (J-6) is not optimised for dexterity (although certain angle values are shown), as the relevant locations in transformation matrix are ignored because the rotation of the blasting nozzle along its longitudinal axis has no bearing on tool's performance. The information from Tables 10.1 and 10.3 is graphically illustrated in Figure 10.5.

Table 10.3 Optimised Joints' Parameters for Initial Task in Stage Two for RRP and Wrist RPR

Parameters	**Values (2)**		**Best Fitness (3)**		**Generation No.**
(1)	**Dexterity**	**Productivity**	**Dexterity**	**Productivity**	**(4)**
J-4(roll)	6^{deg} - 269^{deg}	142^{deg} - 291^{deg}			
J-5 (pitch)	34^{deg} - 354^{deg}	97^{deg} - 354^{deg}	2.27271832 E-06	2.22288085 E-06	2000
J-6 (roll)	N/A	46^{deg} - 354^{deg}			

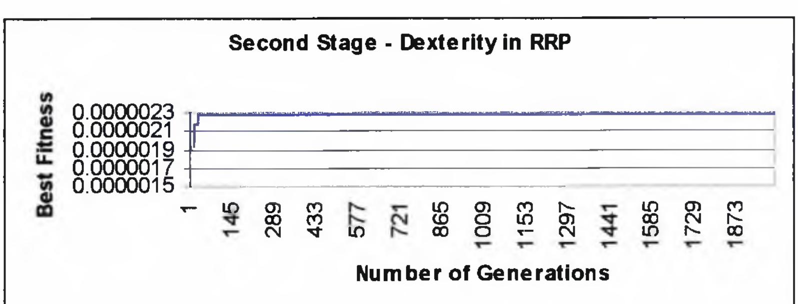


Figure 10.3 Spherical Configuration Performance for Dexterity

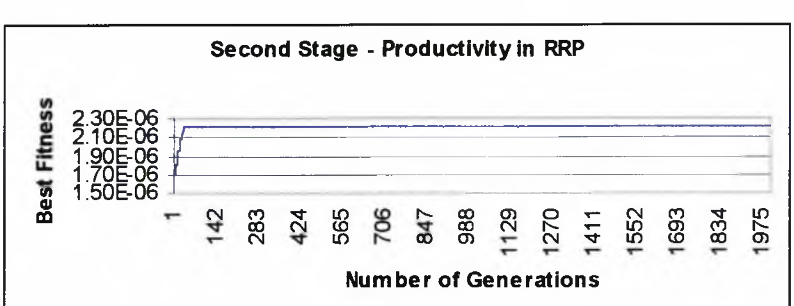


Figure 10.4 Spherical Configuration Performance for Productivity

The different values of the optimised ranges of the last three parameters for dexterity and productivity are considered further. The final values should be estimated according to the designer's additional knowledge about dexterity and productivity relationship and priorities. In this research, the logical sum of both optimal ranges is assumed as the final optimal range. The most economical values for the velocities and accelerations should reinforce the preference for the joints' actuators and are shown in Table 10.4. They are used to calculate the cost of running the robot.

Table 10.4 Values of Velocities and Accelerations for Optimal Productivity

Joints (1)	Velocity [RPM] - except prismatic J-3 (2)		Acceleration [RPM^2] - except prismatic J-3 (3)	
	Dexterity	Productivity	Dexterity	Productivity
J-1	0.17-1.50	1.48-1.50	0.10-2.48	0.14-2.81
J-2	0.07-1.36	0.04-1.41	0.00-3.00	0.0-2.91
J-3	0.48-1.43 [mm/s]	0.31-1.45 [mm/s]	0.00-2.81 [mm/s^2]	0.19-2.81 [mm/s^2]
J-4	0.05-1.50	0.05-1.45	0.00-2.52	0.10-2.33
J-5	0.12-1.29	0.19-1.5	0.23-2.86	0.33-2.33
J-6	0.12-1.19	0.0-1.47	0.33-2.86	0.0-2.48

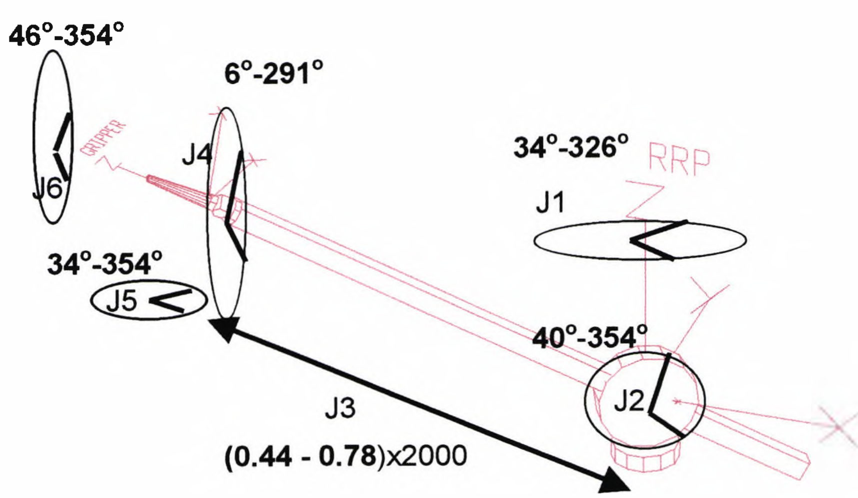


Figure 10.5 Optimal Robot Selected for Specified Initial Task (four paths)

10.3.3 GA Performance for Extended Task

When examining the task, it became apparent that the results obtained, although logical for selected tool activity on a given task, may not provide the kinematic parameters for a versatile robot. As the optimisation is of the robot' s parameters for the given task, the selection of tasks on the bridge has to show more variety of locations, for the robot to be more adaptable to a variety of bridges. In order to address this consideration and further test the sensitivity of the GA method, a fifth path is added to the task. The path is added in line of the first one, but at the level of the base of the robot (0,0,0). On the chosen bridge workpiece, the additional path does not have a practical use, as there is no structural element, where the path is located. It is just added theoretically in order to adjoin versatility to the robot's activity. Again, both configurations are tested over an initial 2000 generations and this time RRR appeared to be superior (Figure 10.6). This outcome is predictable and can be confirmed directly by the shapes of the working envelopes of both spherical and articulated configurations [Spong and Vidyasagar, 1999].

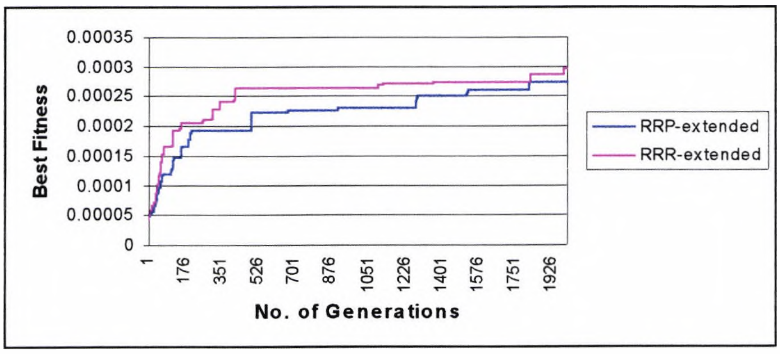


Figure 10.6 RRP and RRR Performance for Extended Task in Stage One

The behaviour of the fitness function over 2000 generations is again investigated (for prospective use on the complex problem) in order to observe the profile of the evolution curve and to determine at which generation to stop the process, if more quick trials are required. It can be seen in Table 10.5 that at the 894th generation there is the last improvement before the long pause, so cut-off value can be reasonably assumed as 900th generation, if required. Table 10.5 shows the parameters' optimisation progress.

Table 10.5 Parameters' Optimisations for Extended Task for RRR

Parameters (1)	Values (2)	Best Fitness (3)	Generation No. (4)
Link proportion	0.54		
J-1	45^{deg}-309^{deg}	0.000192418	330
J-2	23^{deg}-331^{deg}		
J-3	63^{deg}-303^{deg}		
J-1	86^{deg}-326^{deg}	0.000225317	894
J-2	23^{deg}-314^{deg}		
J-3	86^{deg}-303^{deg}		
J-1	86^{deg}-326^{deg}	0.000229784	1055
J-2	23^{deg}-297^{deg}		
J-3	86^{deg}-303^{deg}		
J-1	74^{deg}-326^{deg}	0.000252026	1479
J-2	23^{deg}-314^{deg}		
J-3	85^{deg}-274^{deg}		
J-1	74^{deg}-326^{deg}	0.000297903	2000
J-2	34^{deg}-314^{deg}		
J-3	74^{deg}-286^{deg}		

These results show that improvement in fitness is reflected either in refinement or shifting of the extreme boundaries of the parameters. The intermediate values of parameters change as the fitness is improved, which indicates more precise location of the tool for the intermediate points.

The second stage calculations are approached in the same manner as for the simplified task. The choice of wrists and the orientation of the wrist is added, as the tool is expected to face downwards, so Figure 7.3 is enhanced by additional conditions for orientation.

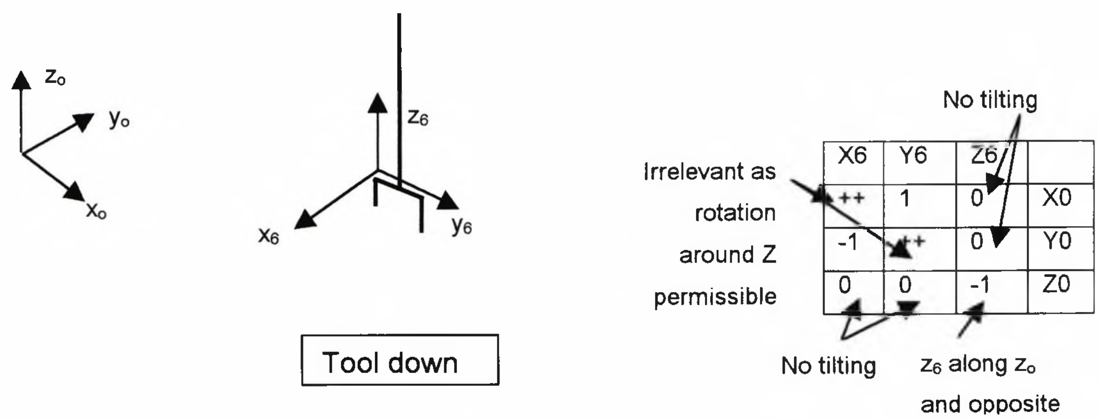


Figure 10.7 Dexterity Matrix Compatibility for Extended Task

Motion for the last (minor) joints after superposition of ranges fulfilling dexterity and productivity criteria is shown in Table 10.6. The information from Tables 10.5 and 10.6 is graphically illustrated in Figure 10.8.

Table 10.6 Optimised Joints' Parameters for Extended Task in Stage Two for RRR and Wrist RPR

Parameters (1)	Values (2)		Best Fitness (3)		Generation No. (4)
	Dexterity	Productivity	Dexterity	Productivity	
J-4(roll)	68^{deg} - 360^{deg}	34^{deg} - 280^{deg}			
J-5 (pitch)	0^{deg} - 360^{deg}	40^{deg} - 348^{deg}	0.96153430 E-06	0.96138184 E-06	2000
J-6 (roll)	NA	0^{deg} - 251^{deg}			

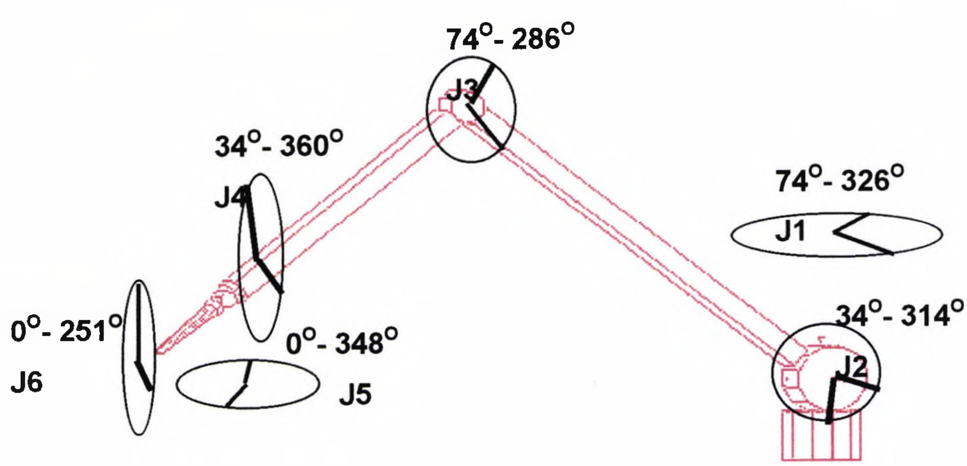


Figure 10.8 Optimal Robot Selected for Extended Task (five paths)

The optimised values for the velocities and accelerations are shown in Table 10.7. No further comments can be offered at this stage regarding the velocity and acceleration outcome, as no assumptions were made regarding the type of motors. The approach to velocity and acceleration optimisation serves a s a guidance for further investigation, while dynamics of a robot are included in the consideration.

Table 10.7 Optimal Values of Velocities and Accelerations for Extended Task

Joints (1)	**Velocity [RPM] (2)**		**Acceleration [RPM^2] (3)**	
	Dexterity	Productivity	Dexterity	Productivity
J-1	0.00-1.45	0.95-1.50	0.52-3.00	0.91-3.00
J-2	0.02-1.19	0.41-1.50	0.38-2.95	0.05-2.48
J-3	0.00-1.47	0.36-1.50	0.48-3.00	0.00-2.67
J-4	0.07-1.50	0.48-1.50	0.14-3.00	0.43-2.76
J-5	0.05-1.38	0.36-1.50	0.24-3.00	0.24-2.86
J-6	0.05-1.41	0.64-1.50	0.43-2.91	0.05-3.00

10.4 Experiments

Once the basic computations and results are obtained, one other interesting aspect remains, the performance of GA on the formulated problem [Bach, 1996]. Therefore, in the following, various types control parameters are going to be altered and the performance and outcome checked for signs of improvement.

10.4.1 Varying Control Parameters

10.4.1.1 Population Size

The population size has been halved and then doubled for stage one, in order to observe the time increase, the best fitness value behaviour and the corresponding parameters' values after [Goldberg, 1989b]. The population size has been 'quickly' tested for the value of the best fitness, similar to stage two, in Section 10.3.2, in order to find the most economical population size for further trials. Here the fitness function as well as values of optimised parameters is closely investigated for stage one. The computational time varies marginally, but the fitness function shows considerable changes, as shown in Figure 10.9. The population size of 100 proves to be optimal for size selected. Therefore, it became vital to check the boundary values of the parameters.

The optimisation trial for the population size displaying the best value of the fitness function at the 2000th generation refined the movement range of the first revolute joint and extended the movement range of the second and third revolute joint, as shown in Table 10.8.

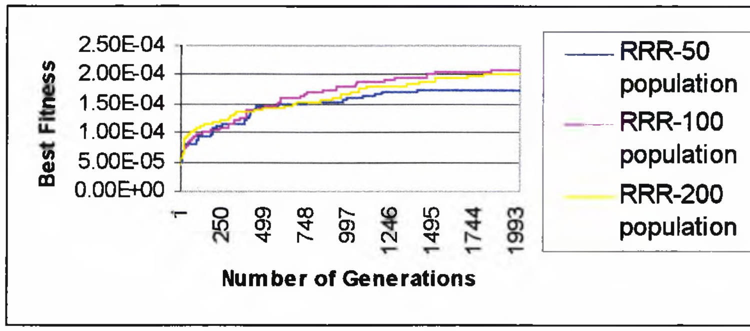


Figure 10.9 RRR Performance for Extended Task and Varied Population for Stage One

Table 10.8 Parameters' Optimal Values for Extended Task (RRR) and Varied Population Size

Parameters (1)	Values (2)			Best Fitness (3)			Generation No. (4)
	50 popsiz	100 popsiz	200 popsiz	50 popsiz	100 popsiz	200 popsiz	
Link proportion	0.38	0.48	0.56				2000
J-1 (deg)	34-320	6-325	57-360	0.000175	0.0002067	0.0002008	
J-2 (deg)	57-360	0-348	6-354				
J-3 (deg)	0-297	57-280	91-274				

Therefore, when micro-GA is introduced, the population size should be carefully investigated, as not always just the increase of the population brings superior results. For the small population, the convergence is too rapid to explore different regions before the new population starts, and for the large population, the convergence takes too long, for enough new populations to be searched within 2000 generations.

10.4.1.2 Rate and Type of Crossover

The rate of crossover, applied throughout, is taken as 0.5, as recommended by some practitioners [Wu et al., 1997]. However, seeing that micro-GA is adopted, Krishnakumar [1989], for example, suggests the crossover rate as 1.0. Therefore, the GA performance is observed for two rates of crossover and compared (0.5 and 0.9) and relative performance assessed. As it is observed in Figure 10.10, the quality of the fitness function varies dramatically.

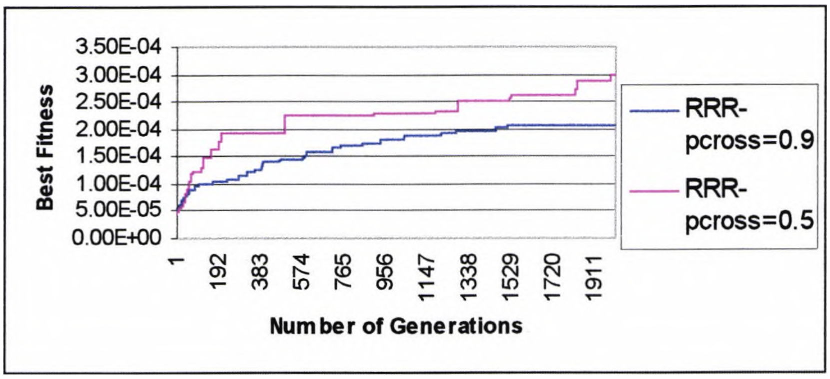


Figure 10.10 GA Performance for Increased Crossover Rate

Table 10.9 Parameters' Optimisations for Extended Task for RRR and Crossover Rate of 0.9

Parameters (1)	Values (2)	Best Fitness (3)	Generation No. (4)
Link proportion	0.48		
J-1	6-325 deg	0.0002067	2000
J-2	0-348 deg		
J-3	57-280 deg		

First of all, the above parameter values serve as further confirmation that the values of control parameters have to be selected by "trail and error", or the "good practice" approach and experience from other practitioners. Deceptively, the higher rate of crossover feels like it should yield better quality solution, but obviously high-performance structures are discarded faster than the selection could produce improvement.

Another interesting aspect is, the optimal range in the motion of the major configuration. They show wider ranges of allowed movement, which if compared with results in Table 10.4, further confirms that the improved fitness carries the refinement in the ranges of the parameters, as well as better parameter utilisation within the boundaries.

Carroll [1996] found that uniform crossover [Syswerda, 1989] tended to preserve more alleles than a single-point crossover. On this basis, uniform crossover was the preferred choice in the research. However, it was considered worth checking whether or not there is a significant difference between the two different crossover choices for this application. Both trials are administered to RRR configuration, first stage and extended task. Figure 10.11 shows that the uniform crossover case (iunifrm=1, iniche=1) approaches the superior solution more rapidly than the single point crossover case (iunifrm=0, iniche=1) up till approx. 1000 generations. Then both improve at the same rate, with their best fitness values inter-changing, but actually they both reach the similar level of optimum at the same time. At the time of the process termination, the single point crossover shows better performance, but from observing the accomplishment of both types, it remains unclear which one produces ultimately better outcome. As for the allele pool, both cases end up oscillating around an average, therefore neither crossover scheme appears to show more or less allele preservation for this problem. As for parameters' values, as indicated in Table 10.10, the ranges seem to be shifted at the moment of current optimum.

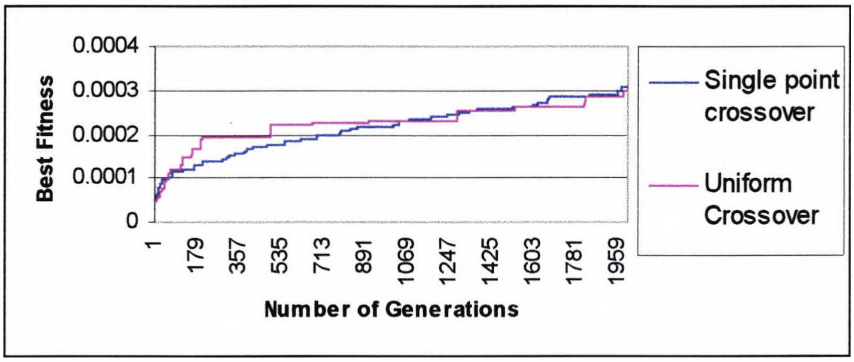


Figure 10.11 GA Performance for Two Crossover Types in Stage One

Table 10.10 Parameters' Optimisations for Extended Task for RRR and Two Crossover Types

Parameters (1)	Values (2)		Best Fitness (3)		Generation No. (4)
	Uniform Crossover iunifrm=1	Single Point Crossover iunifrm=0	Uniform Crossover	Single Point Crossover	
Link proportion	0.54	0.48			
J-1	74 ^{deg} -326 ^{deg}	11 ^{deg} -309 ^{deg}	0.000297903	0.000308419	2000
J-2	34 ^{deg} -314 ^{deg}	29 ^{deg} -331 ^{deg}			
J-3	74 ^{deg} -286 ^{deg}	91 ^{deg} -280 ^{deg}			

10.4.2 Performance Influence of Creep Mutation

Creep mutations can be useful, as they can slide gene pool towards the optimal solution, rather than just jump towards it. Figure 10.12 compares the non-uniform, non-niching, creeping case (iunifrm=0, iniche=0, icreep =1, ielite=1) with the non-uniform, non-niching, non-creeping case (iunifrm=0, iniche=0, icreep =0, ielite=1). With creep mutations removed, the GA did not find the similar level of 'optimisation' until approx. 1000th generation, but still managed to catch up with the similar level of optimality at the approx. 1600th generation. Since the addition of creep mutation found the near-optimal solution marginally sooner, than without them, creep mutations appear to be of slight benefit to the GA. Additionally, the values of the parameters also show improvement, as indicated in Table 10.11.

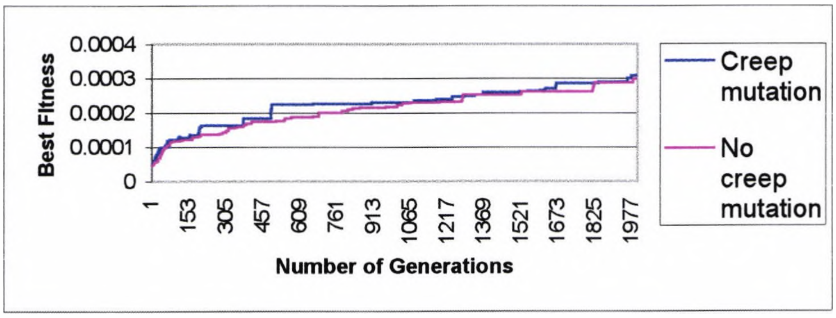


Figure 10.12 GA Behaviour with and without Creep Mutation

Table 10.11 Parameters' Optimisations for Extended Task for RRR with and without Creep Mutation

Parameters (1)	Values (2)		Best Fitness (3)		Generation No. (4)
	Creep mutation	No creep mutation	Creep mutation	No creep mutation	
Link proportion	0.54	0.48			
J-1	74 ^{deg} -326 ^{deg}	11 ^{deg} -326 ^{deg}	0.000297903	0.000278322	2000
J-2	34 ^{deg} -314 ^{deg}	29 ^{deg} -331 ^{deg}			
J-3	74 ^{deg} -286 ^{deg}	74 ^{deg} -280 ^{deg}			

10.4.3 Elitism and GA Behaviour

Elitism forces the best individual to passed to the subsequent generations, until someone (from the best individual's progeny or an unrelated individual) indicates better fitness and replaces the 'winner'. When elitism is removed, the best individual must develop through generations and 'hope' to be selected for further breeding. The basic tournament selection with elitism (iuniform=0, iniche=0, icreep=0, ielite=1) is compared with basic tournament selection without elitism (iuniform=0, iniche=0, icreep=0, ielite=0) in Figure 10.13. When elitism is removed, the GA performance is inhibited. The value of best fitness was well below the one with elitism, at the 2000th generation, when search was terminated. Clearly elitism is beneficial for this application.

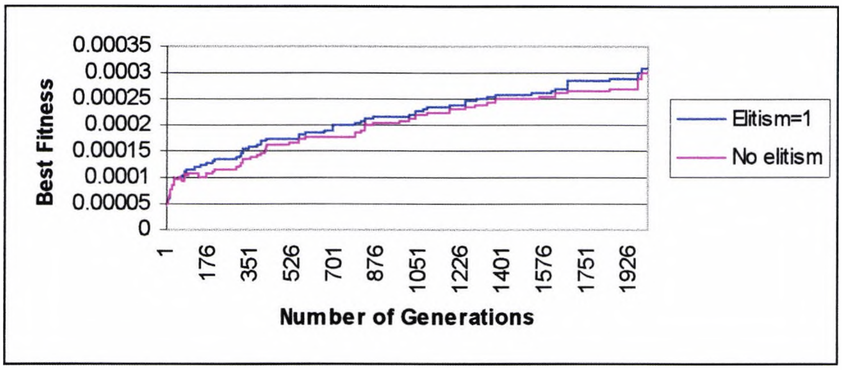


Figure 10.13 Influence of Elitism on GA Performance

Table 10.12 Parameters' Optimisations for Extended Task for RRR and with and without Elitism

Parameters (1)	Values (2)		Best Fitness (3)		Generation No. (4)
	Elitism	No elitism	Elitism	No elitism	
Link proportion	0.54	0.48			
J-1	74 deg -326 deg	34 deg -326 deg	0.000297903	0.000278419	2000
J-2	34 deg -314 deg	34 deg -331 deg			
J-3	74 deg -286 deg	91 deg -309 deg			

10.5 Computer Simulation (GRASP) Results

The results from the optimisation were input into GRASP. All optimised values of the kinematic parameters were input into the software according to the results obtained with GA technique. The GRASP files can be found in Appendix E. The path points were input in the form of tracks and collision envelope was added for potential collision detection. Then the Grasp was run for both cases - RRP and four paths and RRR and five paths. On both occasions, the tracks were run without any warnings, such as either 'collision detected' or 'joints limits out of range'. This result confirmed that the results obtained from GA optimisation were viable and realistic. The author would like to stress that GRASP did not address the optimality of the parameters' values.

Each position of the arm, during running of the track, was frame-captured and two sets (RRP and RRR) are assembled at the end of this chapter in Figures10.16 and 10.17. The trolley and the platform were removed for clarity. Therefore, the full graphical representation of the workspace and the robot input into GRASP is shown in Figure 10.14 below.

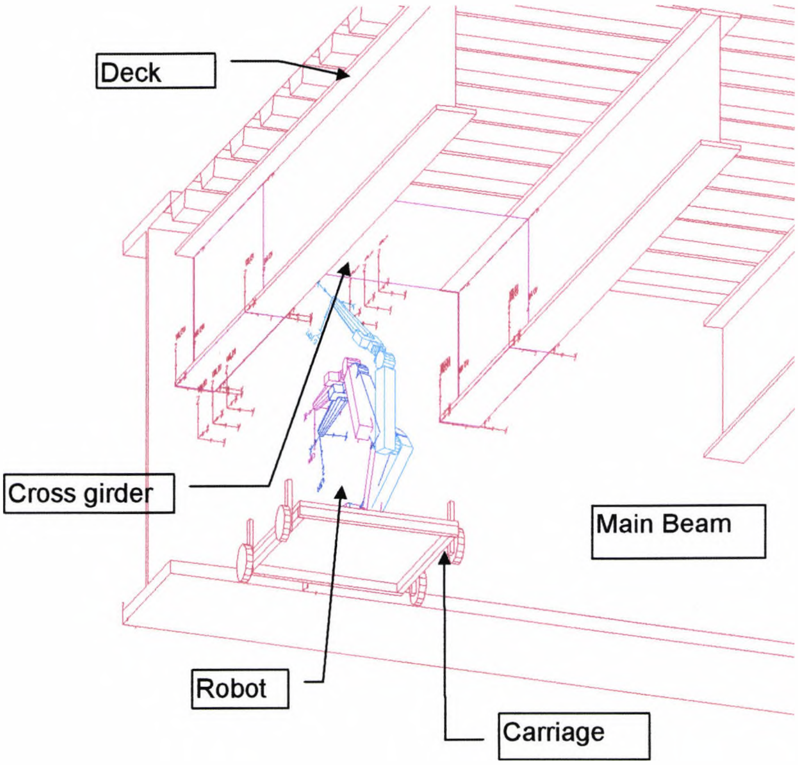


Figure 10.14 Graphical Representation of the Workplace and the Robot

10.6 Final Configuration Models

Necessary modifications to robot link lengths and configurations have been established using genetic-based techniques. Further, the modified version, assembled on a mobile platform (carriage) and combined with a modular suspended access platform, is tested against the pre-selected bridge geometry. Prior to the calculations and analysis of the representative example, a graphical description of the working environment is presented (Figures 10.14 and 10.15), in order to further clarify the nature of workplace and the robot.

Figure 10.14 shows an example of a section of a steel bridge with composite decking under restoration, and the representation of an RRR arm on a mobile access platform positioned under the bridge. The global platform, suspended from the bridge and supporting the robot, is omitted for clarity. Figure 10.15 shows the complete environment.

Figure 10.15 General View of the Proposed System

10.7 Positioning within ED and Extensions to Other Phases

The approach introduced in Chapters 7 and 8 and tested in this Chapter clearly positioned the method within the conceptual phase of the design with links to the embodiment phase. At the completion of the conceptual phase the most appropriate combinations of the item were concretised as primary solutions variants, so they could be evaluated and assessed. Use of QFD setting, analysis of restoration tasks, bridge geometry, existing access systems, and allowing for use of commercially available enabling technologies, identified the overall direction and distinguished set of schemes for evaluation, which was within the conceptual part of ED. The GA refinement initiated the complete compilation of the techno-economic structure of device. Optimal values of the kinematic parameters initiated and guided the analysis of the design principle feature of design and indirectly of all other principle features as listed in Table 3.2. New designs frequently involve only certain modules or system components, or adjustment to different circumstances, without altering the solution principle. This produces fluid boundaries between the phases of the ED. As identified in Chapter 3, the ED problem addressed in this research lies between selection design, redesign and variant design, and therefore the realisation of the general objective of the embodiment design was focused at reliability, simplicity and non-ambiguity.

10.8 Conclusions

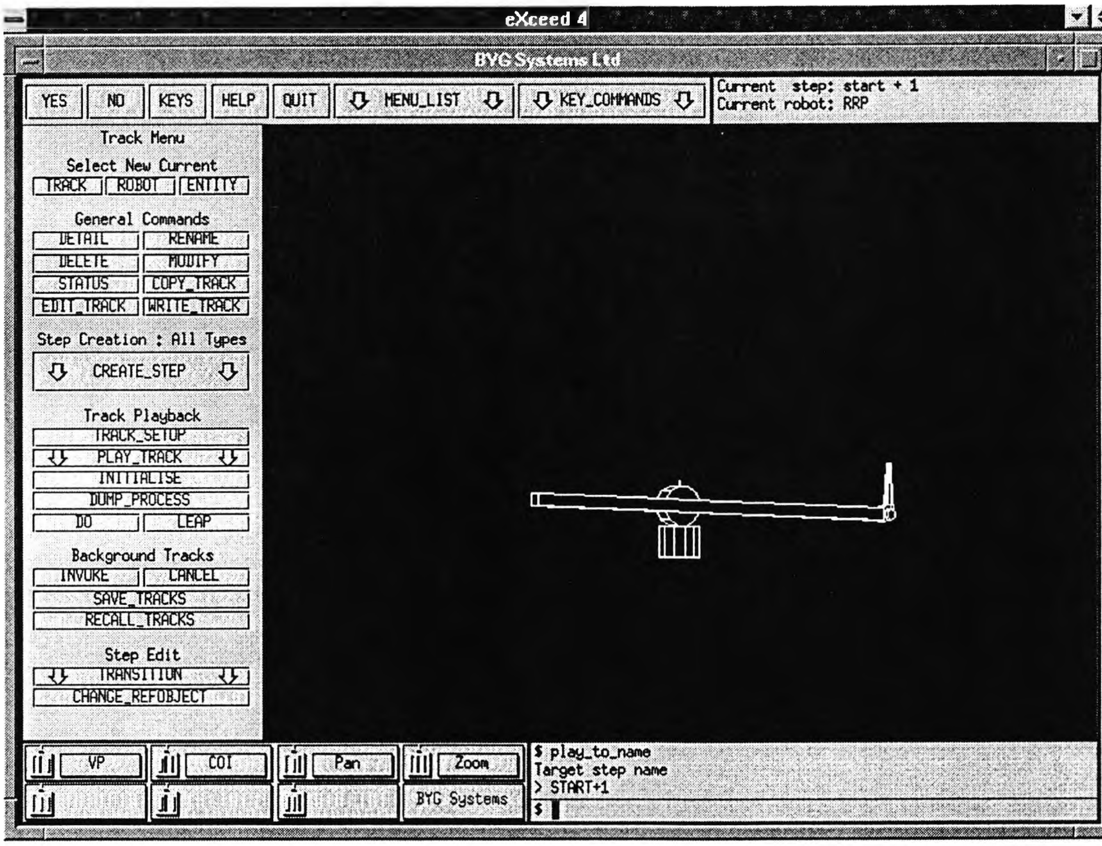
In chapters 7 and 8, the method was established, using an innovative and efficient GA application approach to optimising kinematic parameters of the robot. The previous chapter tested the methodology on a construction robot MPIR, optimised using logical elimination and computer simulation. Using the GA approach confirmed these results and tendencies observed with MPIR. Therefore, it became feasible to apply the method to the case study of a steel bridge restoration robot and task of cleaning the surface off paint with blasting. The procedure was set up as described in Chapter 7 and the GA performance and behaviour in relation to the results was observed. Initially, a traditional approach to GA was adopted, which yielded very large population and an efficient way of obtaining results was lost in excessive computing time. Therefore, enhancement methods were researched and applied, such as niching and micro-GA. This made the observations much easier and faster. All the experiments were run based on 2000 generations, as it was decided that convergence was not the best method of terminating the search and treating the outcome as absolute optimum.

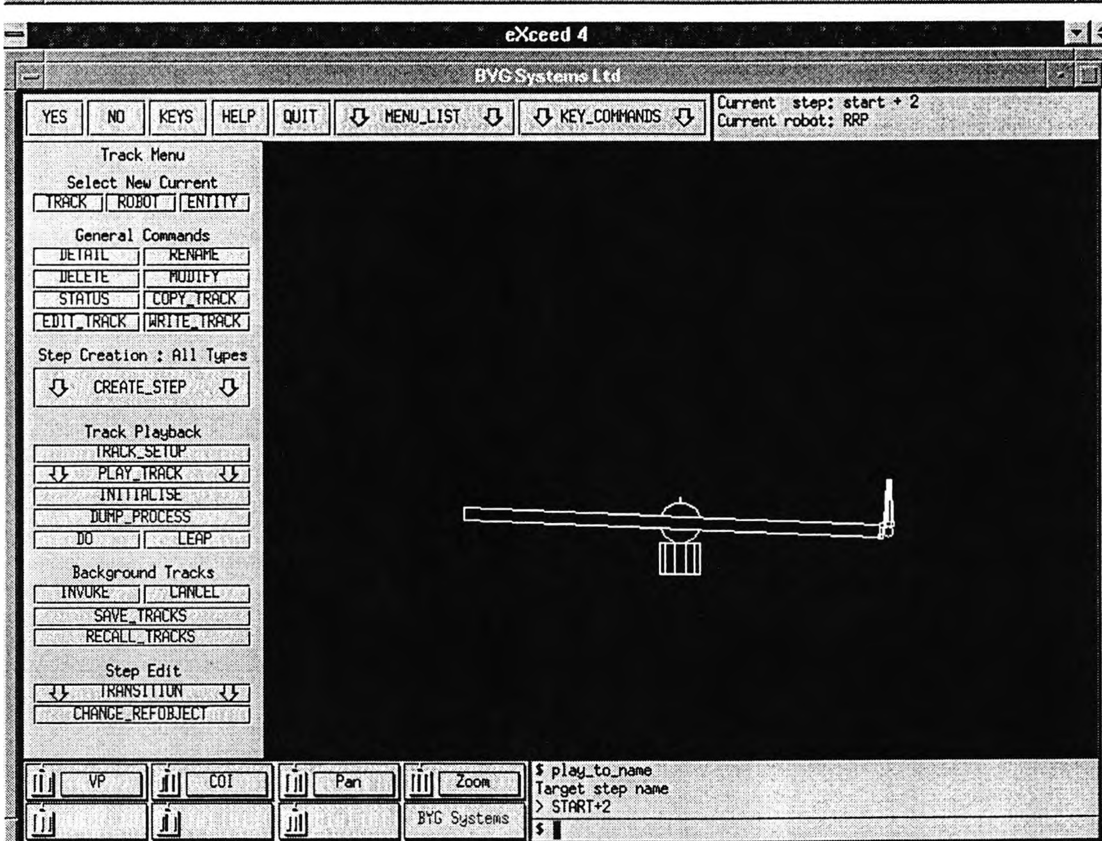
In cases of large numbers of parameters and several criteria, it is more beneficial to observe the behaviour of the solutions throughout generations and determine when to stop and, therefore, assume the result as optimal. The shape of evolution curve and the behaviour of the best fitness led to the conclusion that, in some cases, as little as 500 generations could yield satisfactory enough results. So in complex optimisation problems the analysis of the evolution curve on a trial run may provide the designer with the information, where to terminate the search, in order to save time and still obtain 'near-optimal' solution. Experiments confirmed sensitivity of the GA search and correctness of the approach. For the task positioned in the "unit workspace", in a location centrally within the working envelope of the spherical configuration, the RRP was the one selected as superior for the pre-set conditions. The articulated robot could also perform the task, but it was faster and more precise to use a least sophisticated configuration if possible. When, another path was added, located within the typical region of articulated robots, but outside the reach of the RRP, the RRR configuration was selected. The boundaries of the motion regions of all the joints were refined over the generations of the search. The preferred wrist type for the activity was also selected according to the predictions. Then, the GRASP simulation closely matched the time for the activity, which was the reciprocal of the fitness function, when the entire robot's parameters' boundaries were the same as in GA.

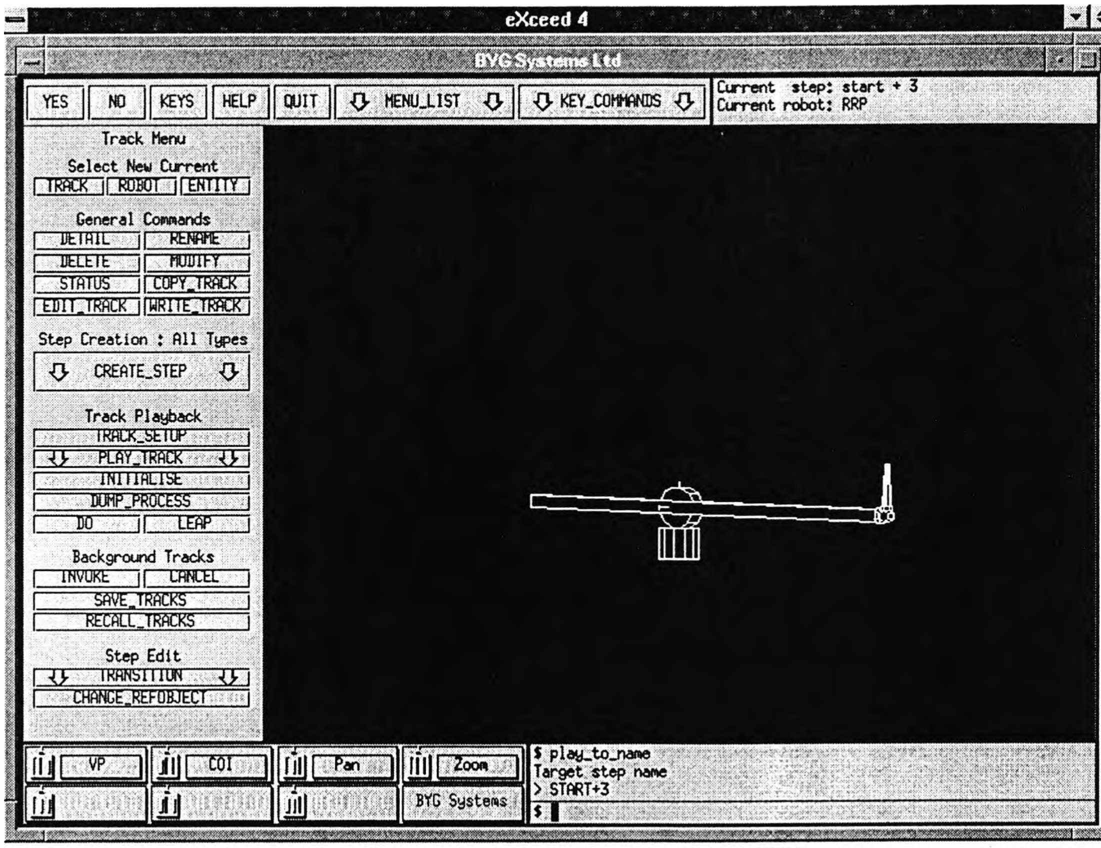
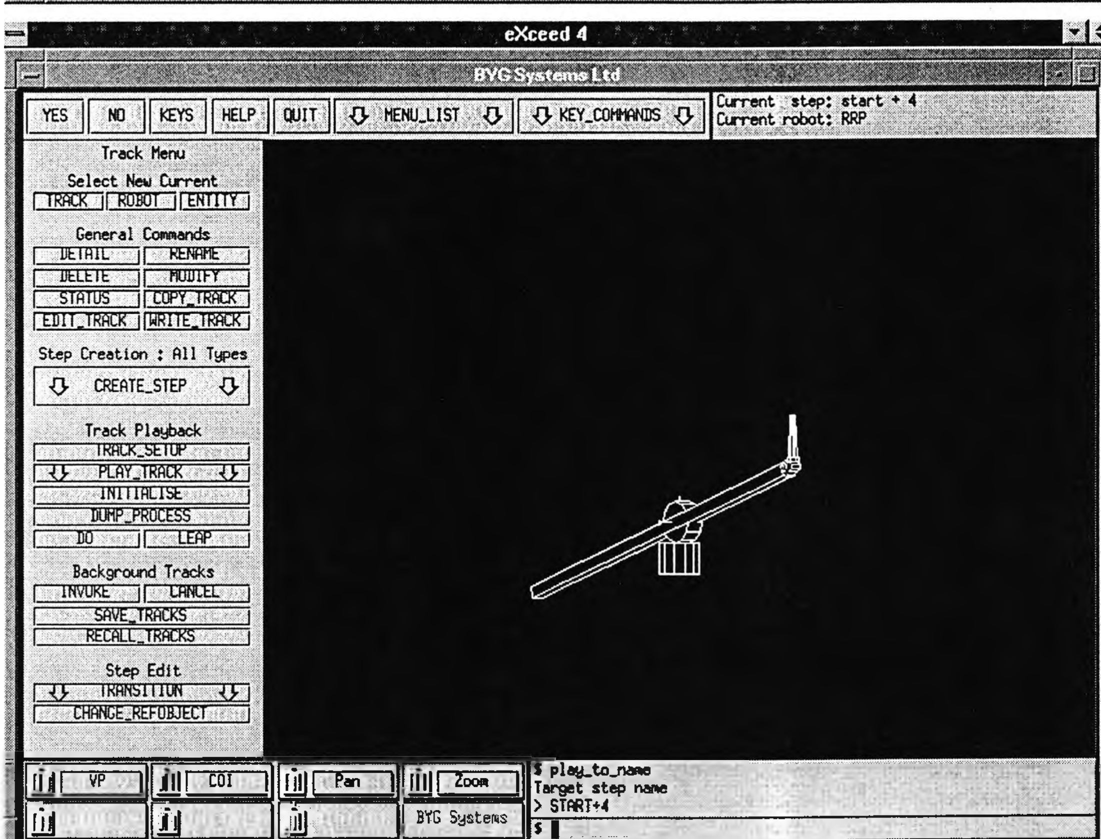
Further experiments were run to test the sensitivity of the GA parameters. The size of the population did not improve the level of the fitness function, or the boundaries of the parameters. Increasing the probability of the crossover did not improve the results, as too much diversity causes loss of good schema. The uniform crossover and the single-point crossover both reached similar level of optimality at the same time and neither crossover scheme showed different level of allele presentation.

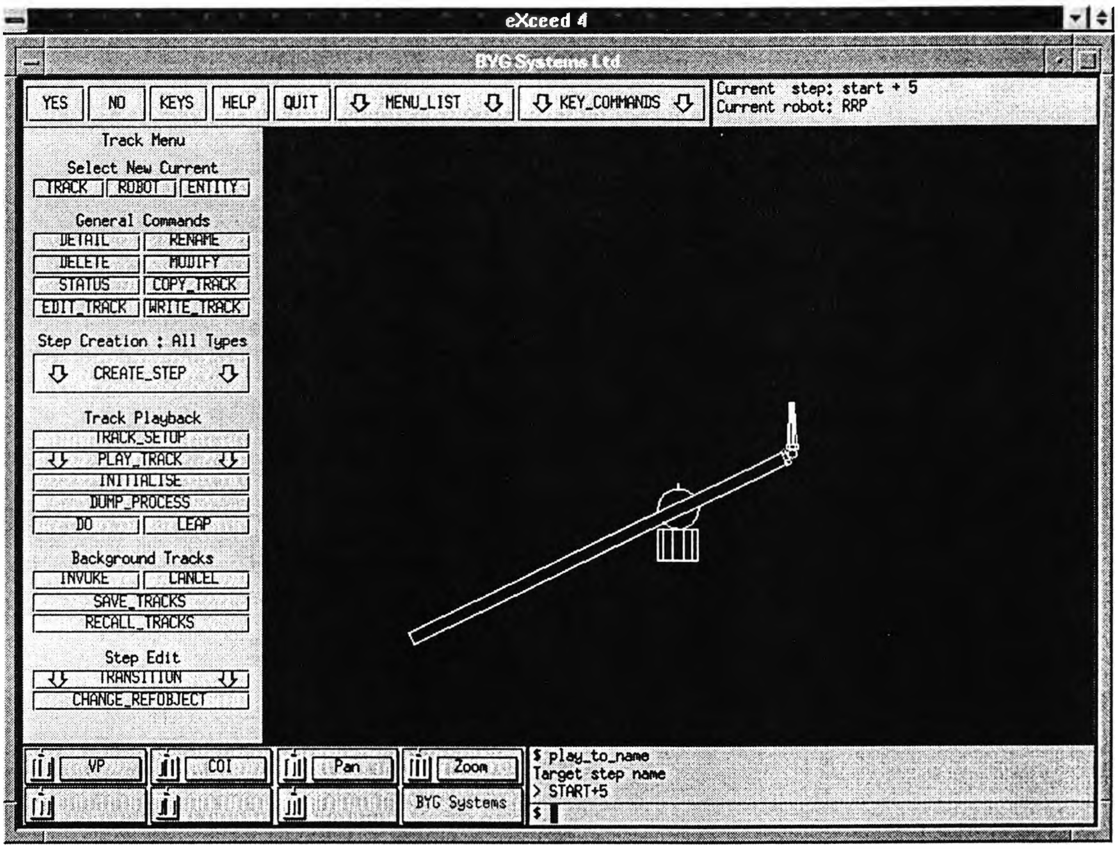
The influence of the creep mutation was significant, time-wise, as the optimisation process would need more generations to arrive at similar level of optimality. Also removal of elitism indicated GA performance inhibition.

Cumulatively, the results and application confirmed the method's logic. It could be, therefore, further expanded into different workspaces (bridges), tasks (tool manipulations) and link to other phases of the ED, as indicated in Section 10.7.

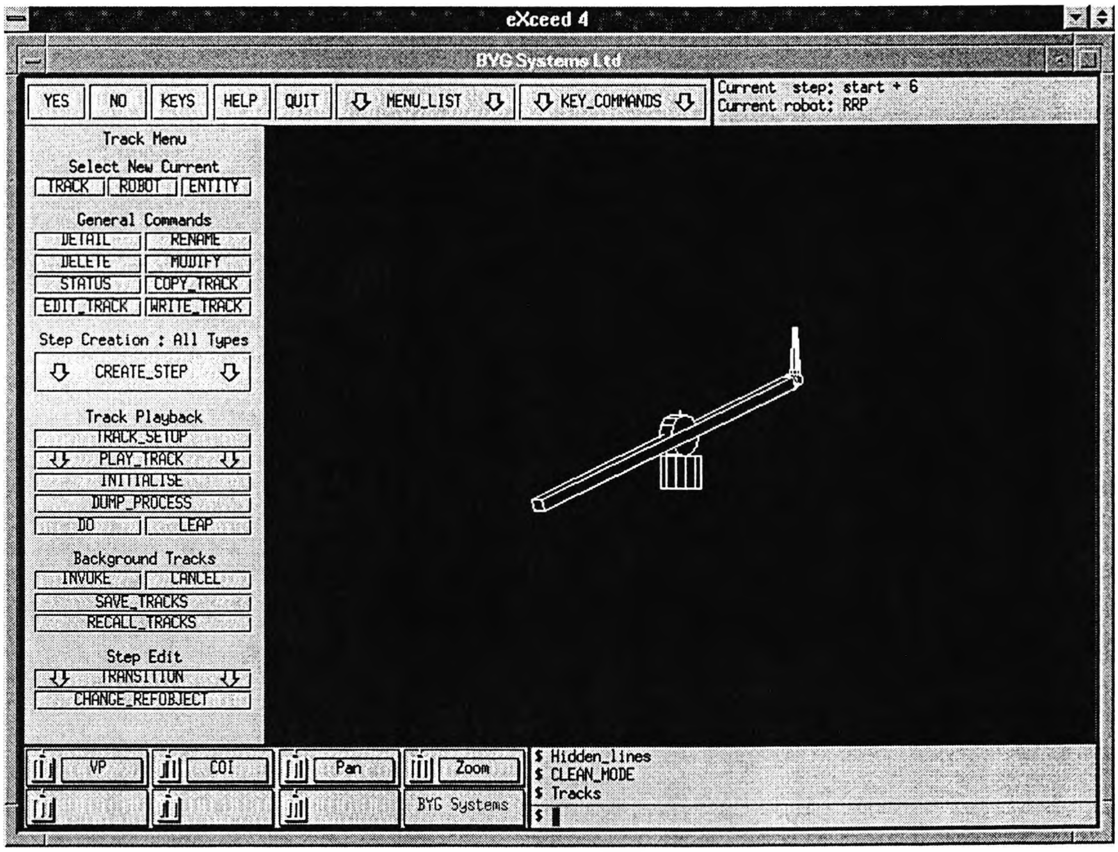

eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 1
Current robot: RRP
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ play_to_name
Target step name
> START+1
$


eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 2
Current robot: RRP
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ play_to_name
Target step name
> START+2
$

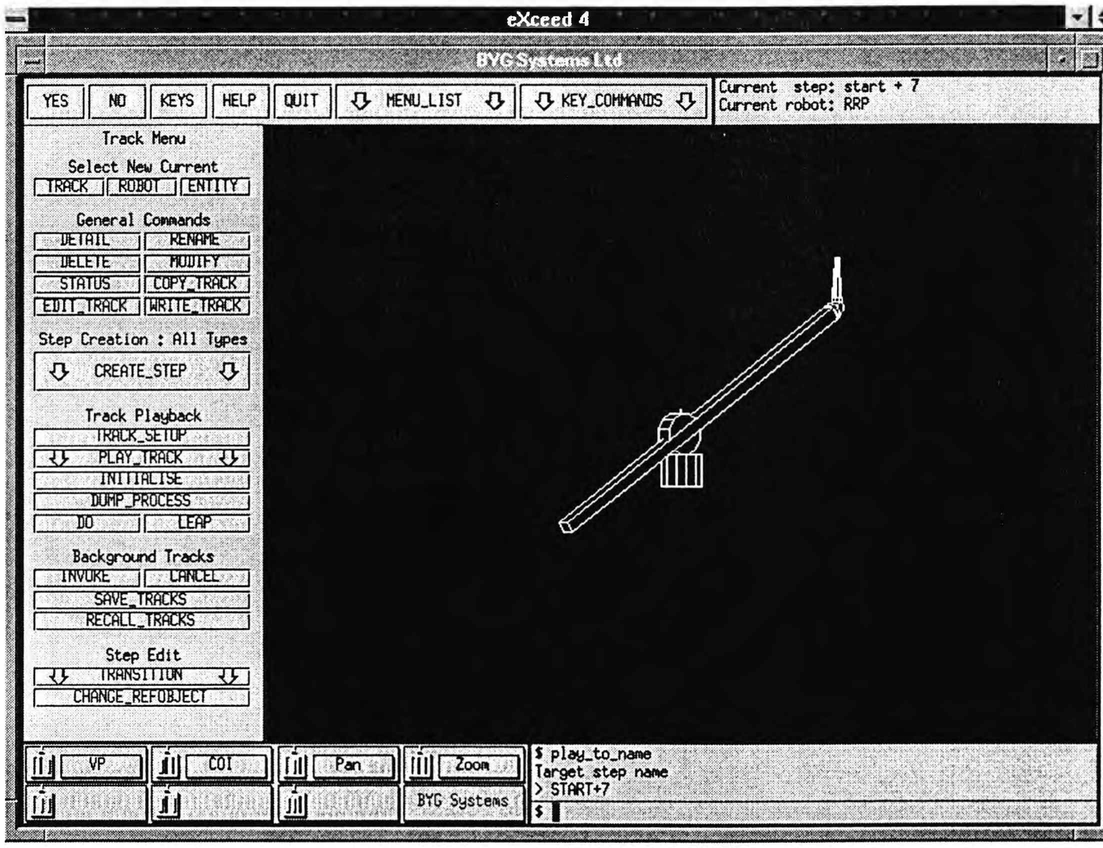

eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 3
Current robot: RRP
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ play_to_name
Target step name
> START+3
$

eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 4
Current robot: RRP
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ play_to_name
Target step name
> START+4
$

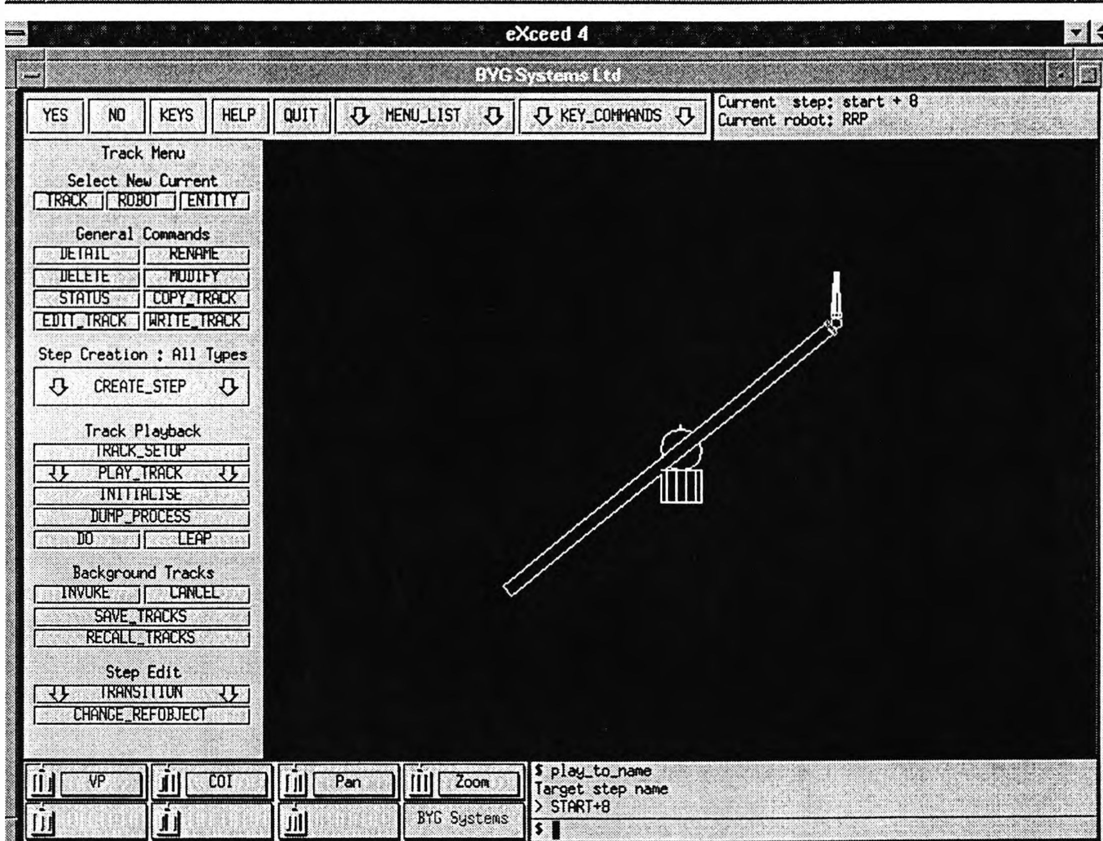


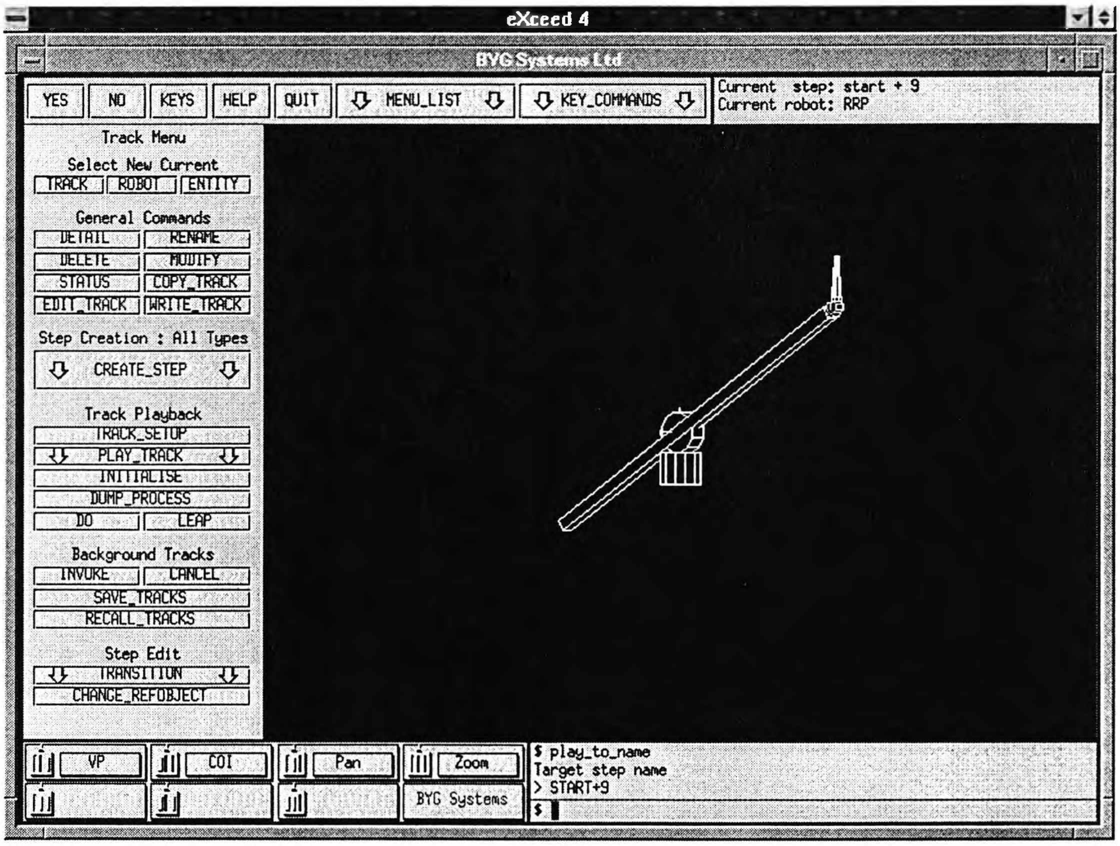
eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 5
Current robot: RRP
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ play_to_name
Target step name
> START+5
$

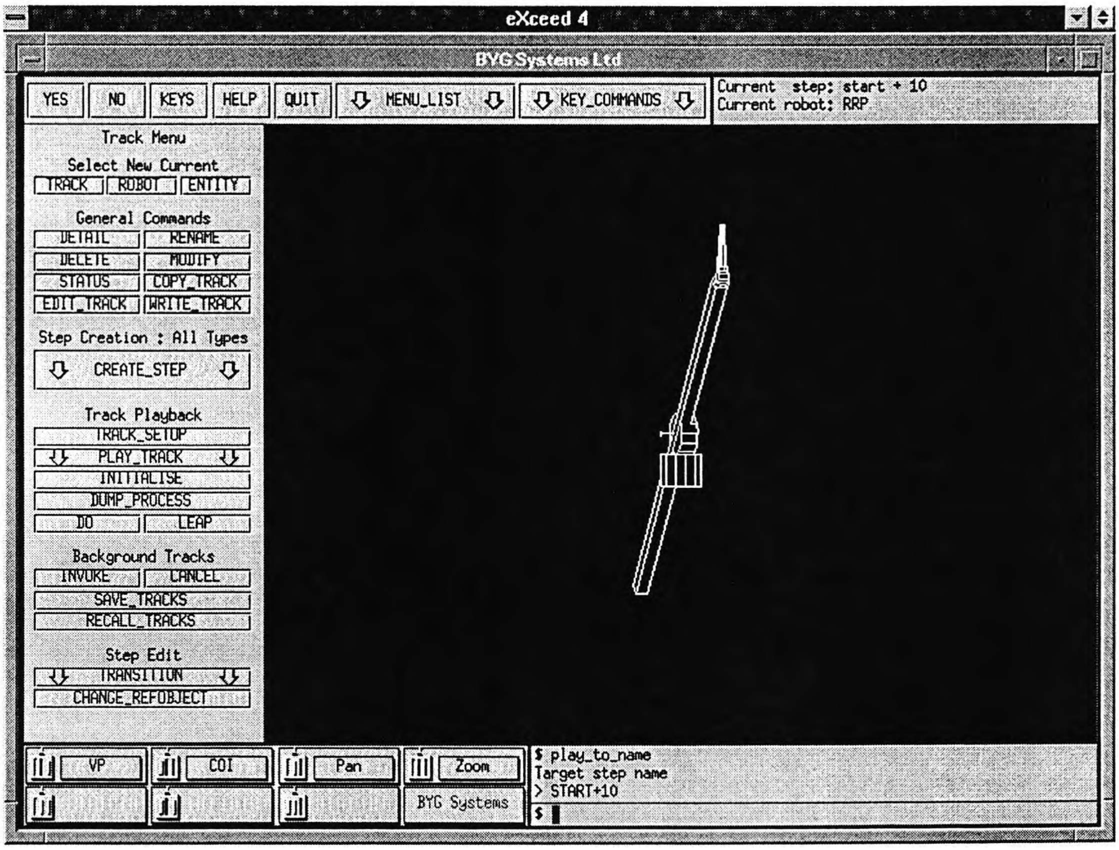


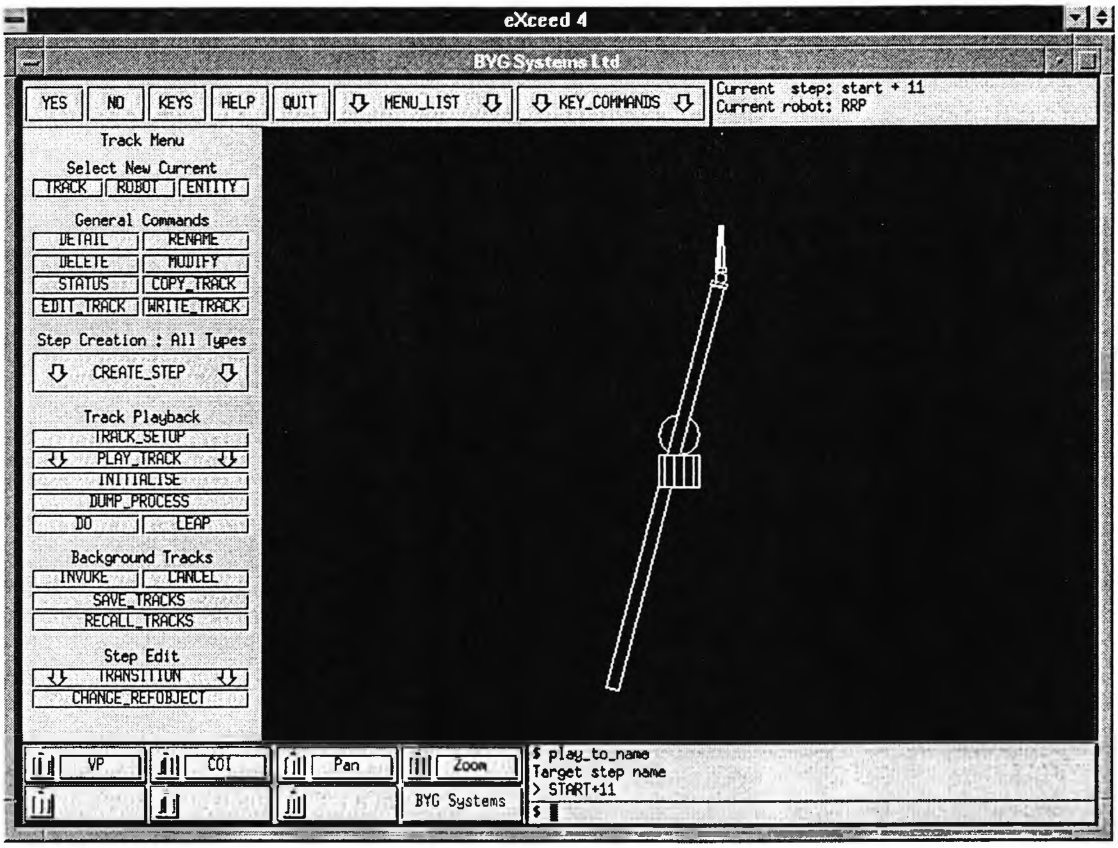
eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 6
Current robot: RRP
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ Hidden_lines
$ CLEAN_MODE
$ Tracks
$

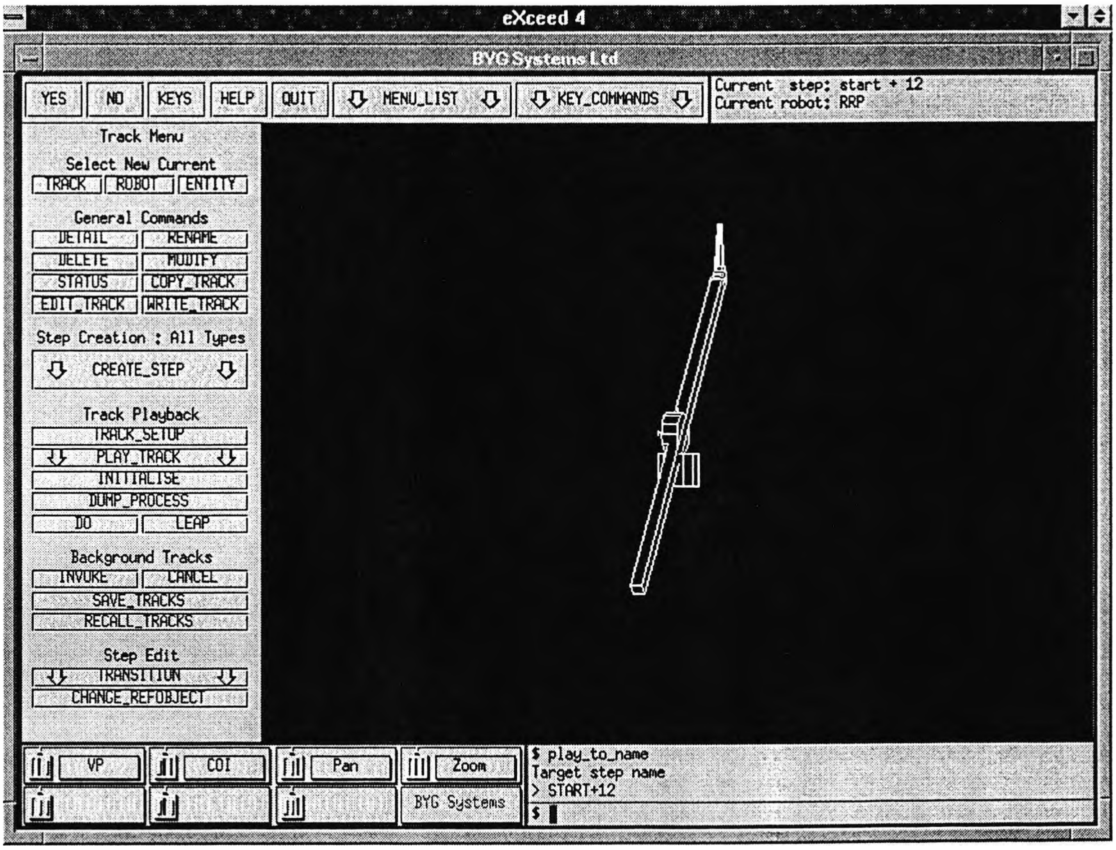

eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 7
Current robot: RRP
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ play_to_name
Target step name
> START+7
$


eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 8
Current robot: RRP
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ play_to_name
Target step name
> START+8
$


eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 9
Current robot: RRP
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ play_to_name
Target step name
> START+9
$


eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 10
Current robot: RRP
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ play_to_name
Target step name
> START+10
$


eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 11
Current robot: RRP
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ play_to_name
Target step name
> START+11
$


eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 12
Current robot: RRP
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ play_to_name
Target step name
> START+12
$

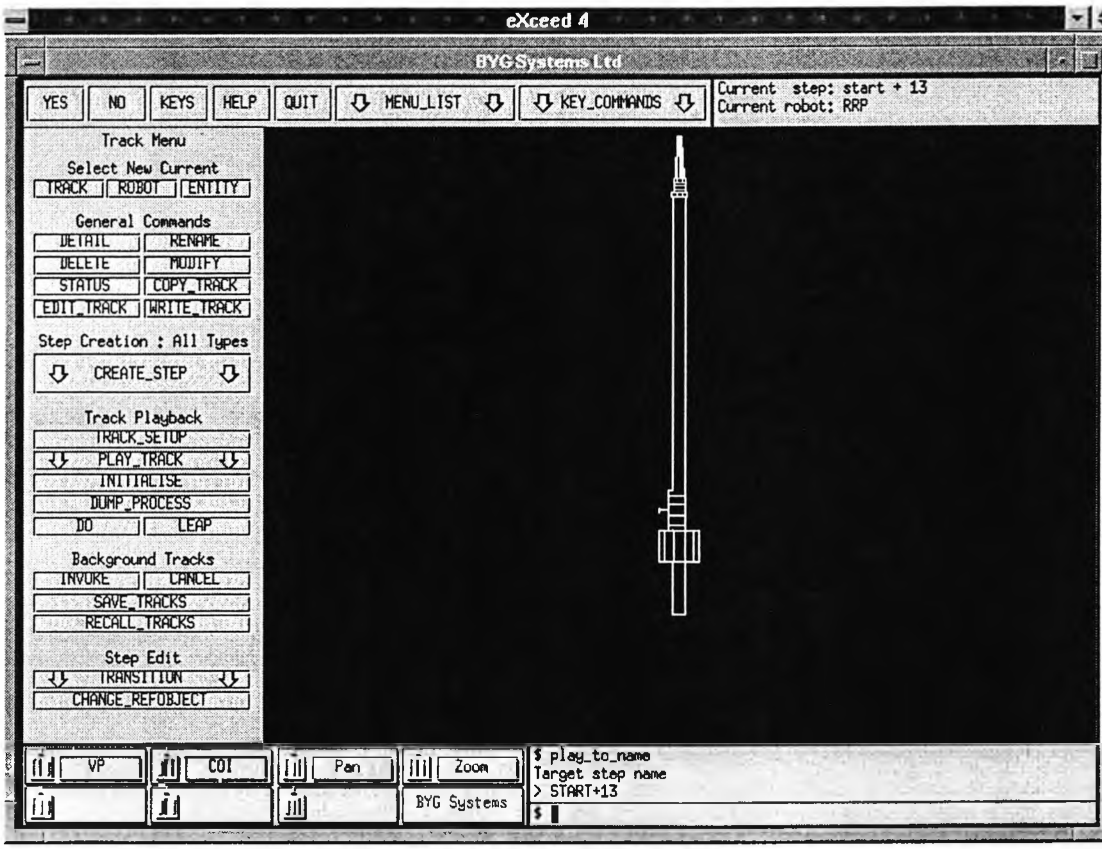
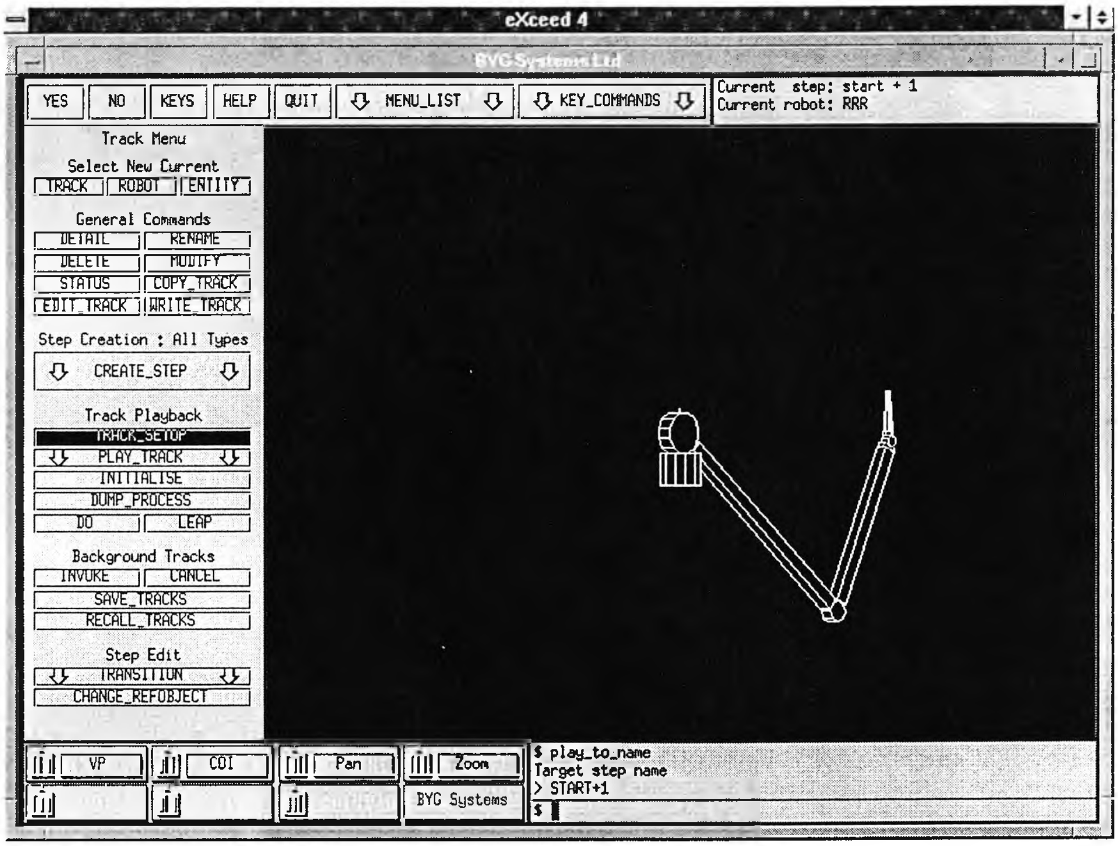
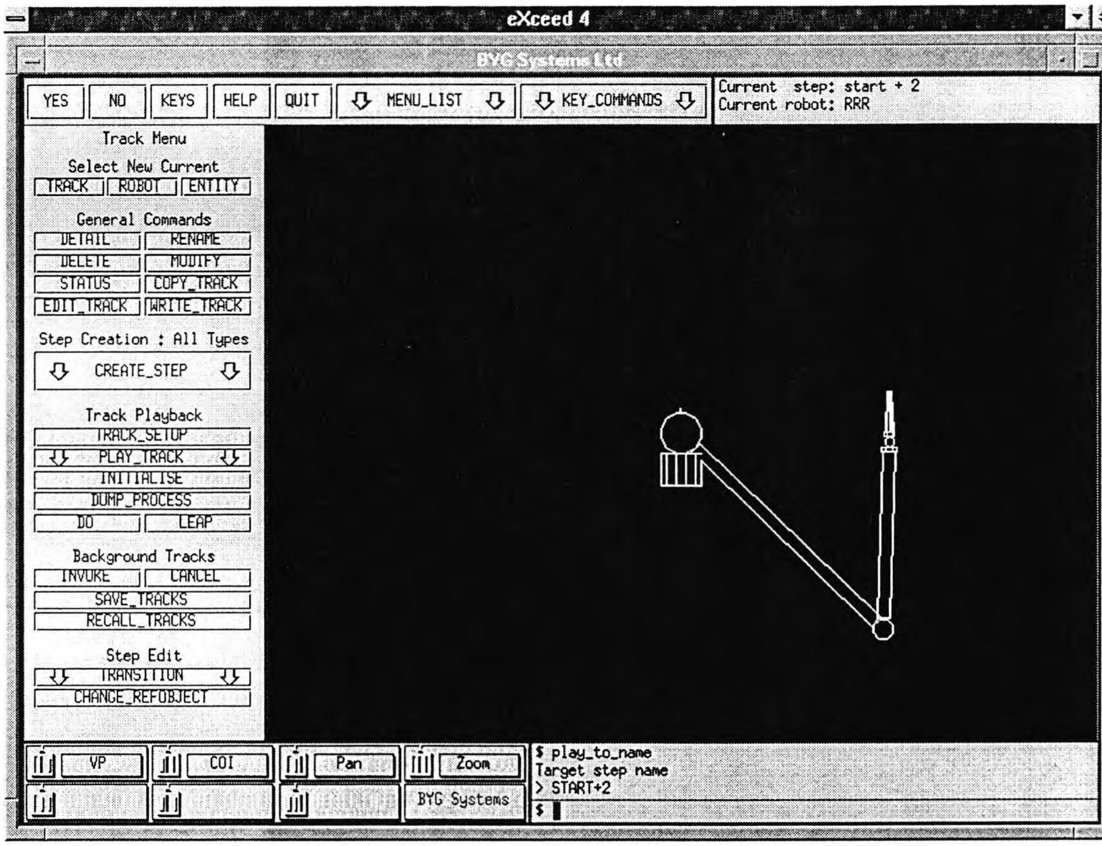
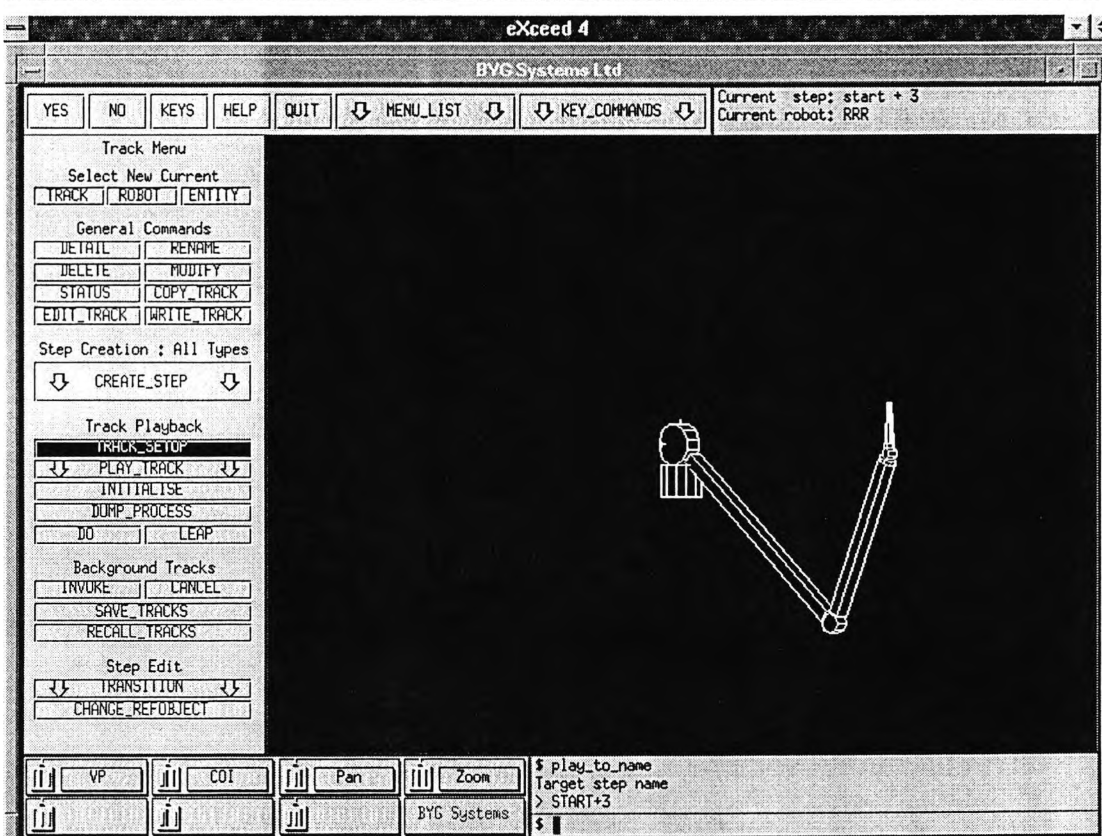
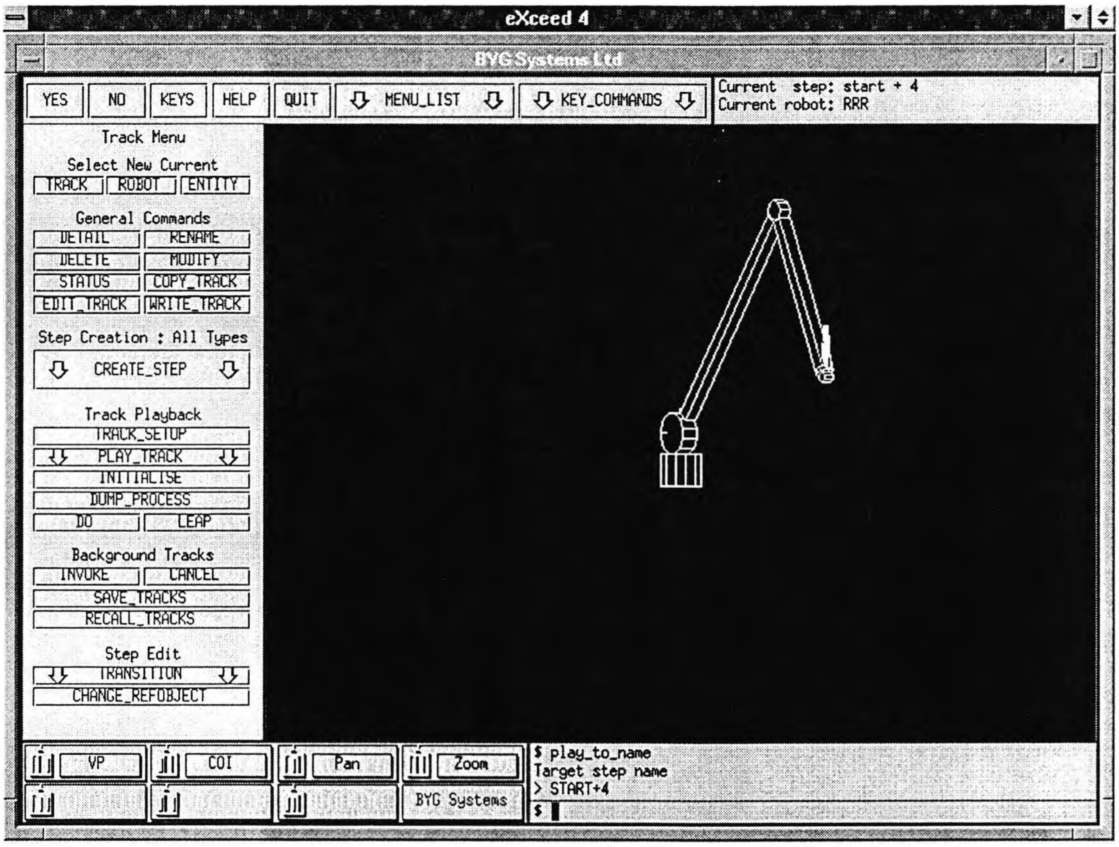


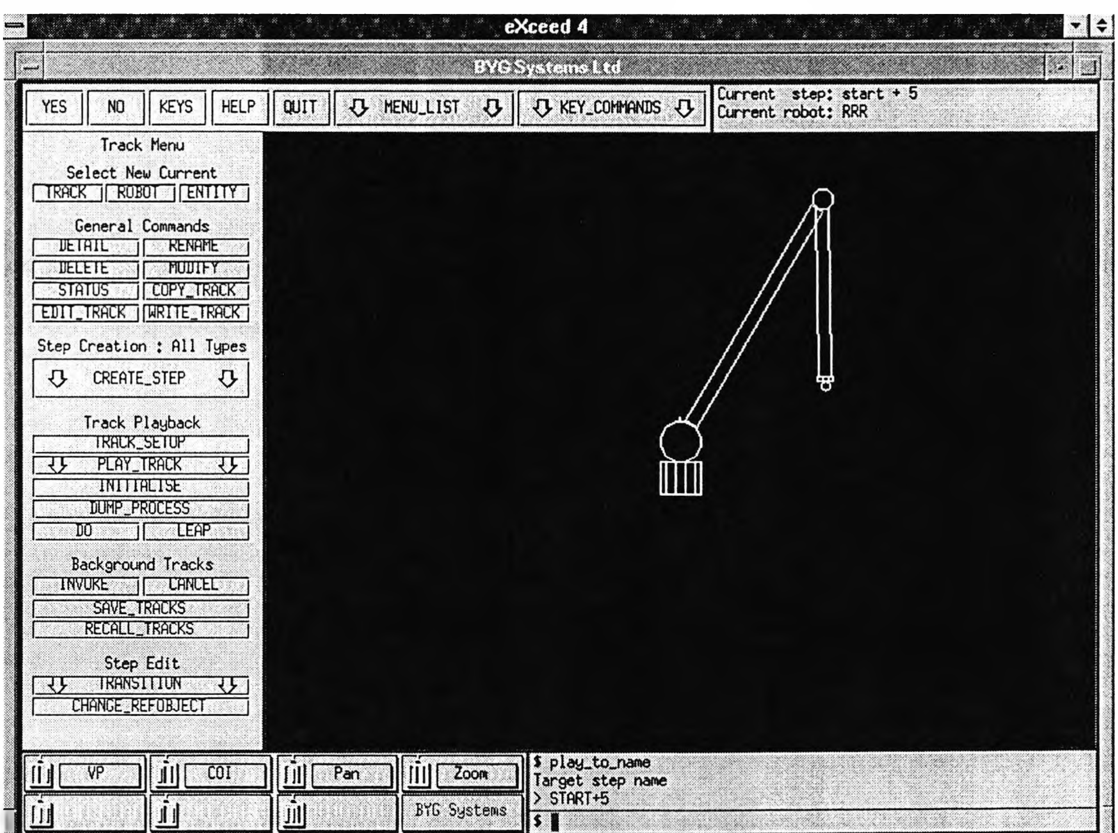
Figure 10.16 Sequence of 12 Steps of Track of RRP Robot using GRASP

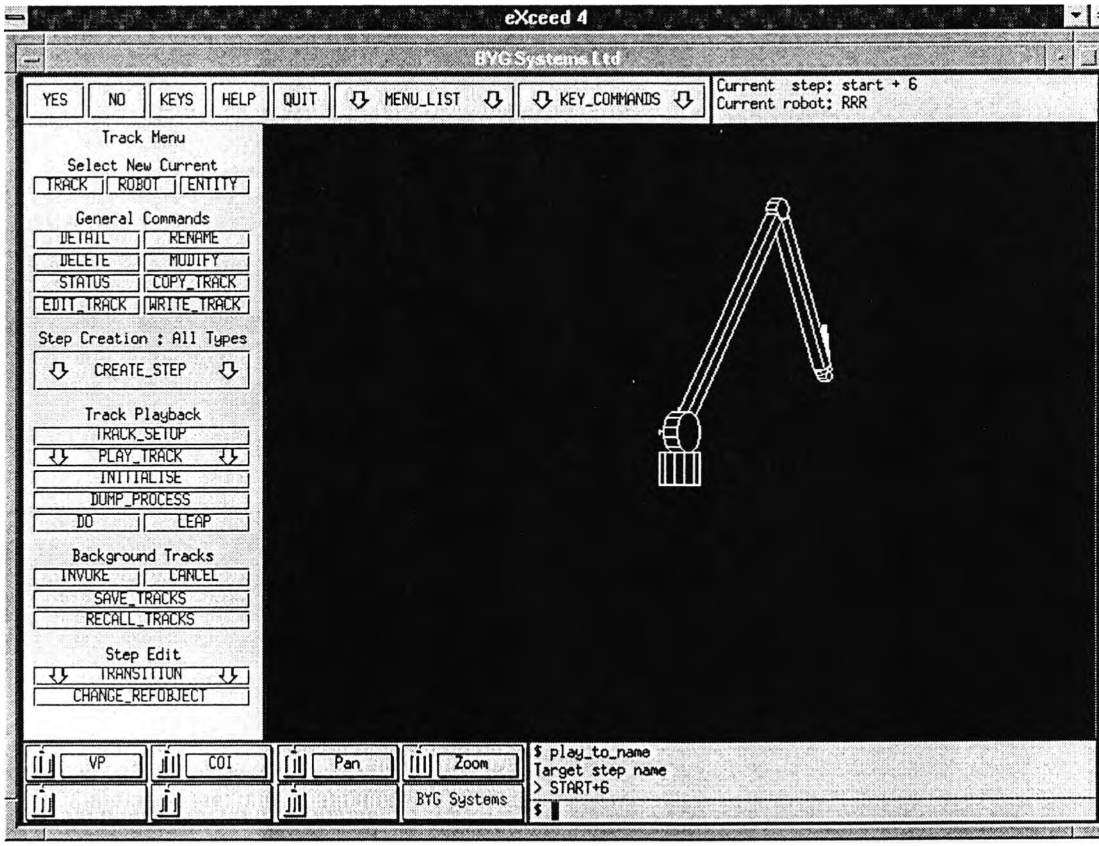

eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 1
Current robot: RRR
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ play_to_name
Target step name
> START+1
$

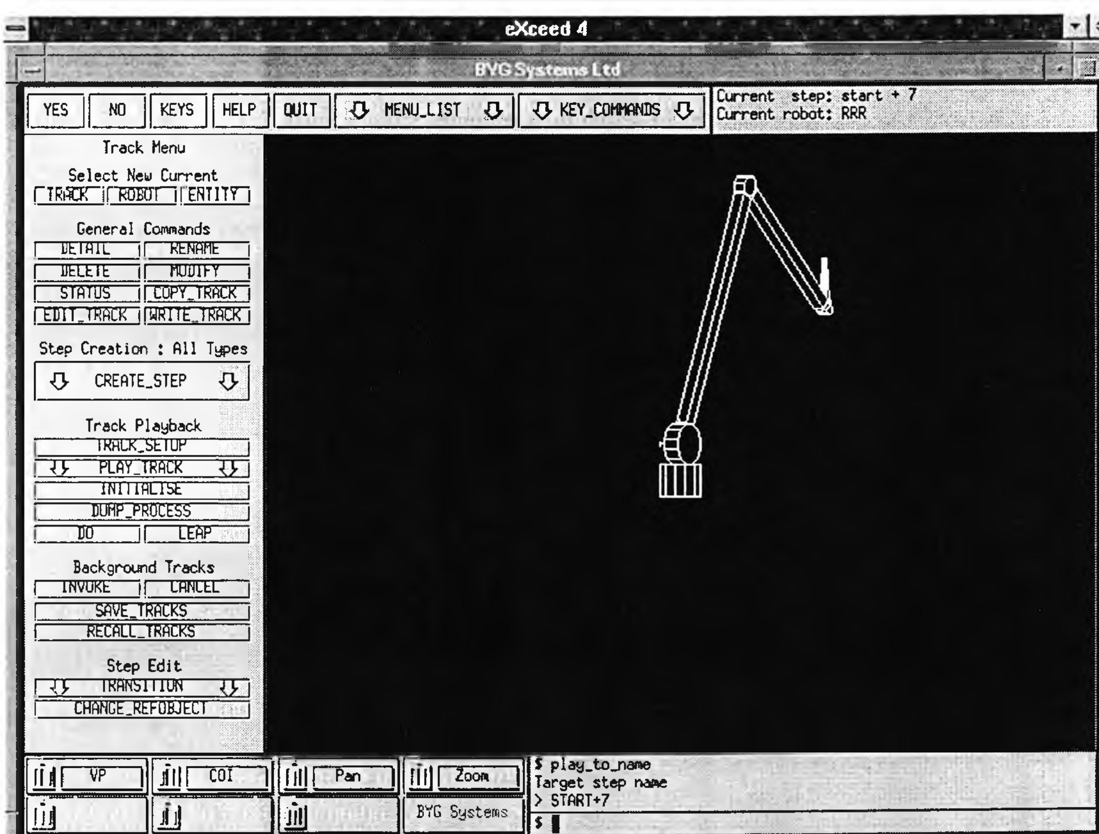

eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 2
Current robot: RRR
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ play_to_name
Target step name
> START+2
$

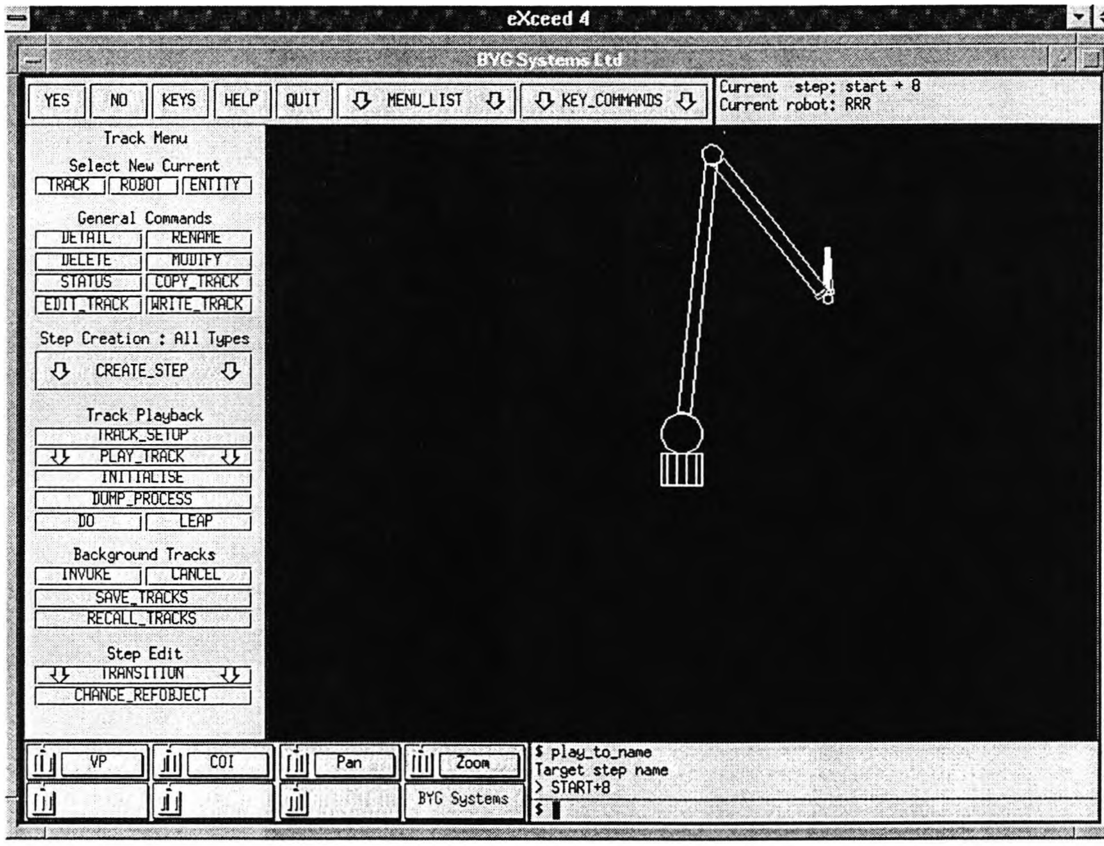

eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 3
Current robot: RRR
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ play_to_name
Target step name
> START+3
$

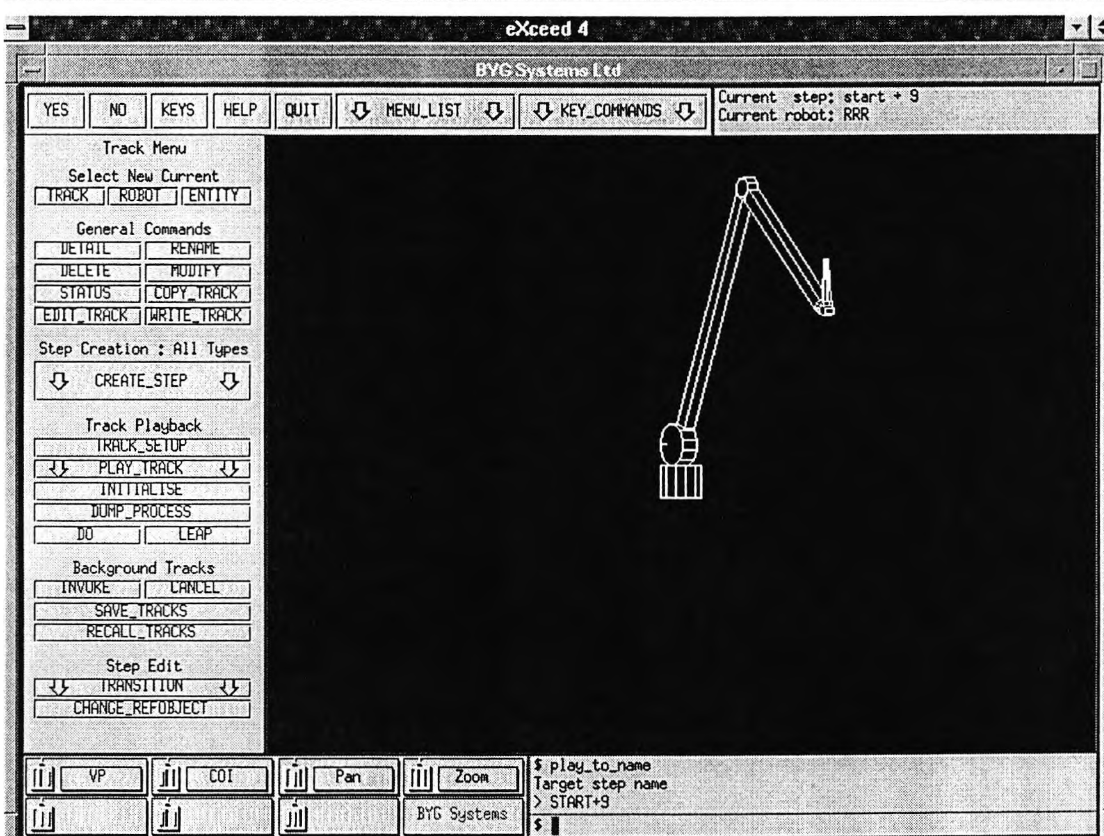

eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 4
Current robot: RRR
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ play_to_name
Target step name
> START+4
$

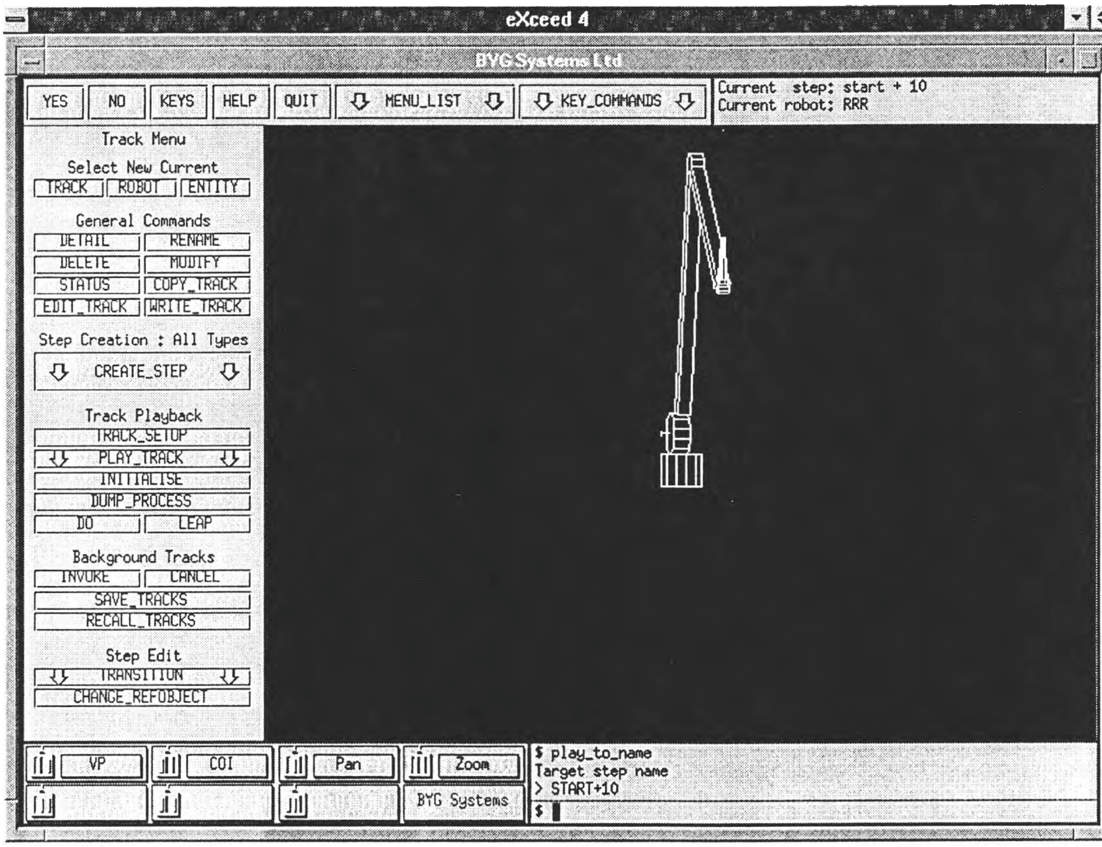

eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 5
Current robot: RRR
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ play_to_name
Target step name
> START+5
$

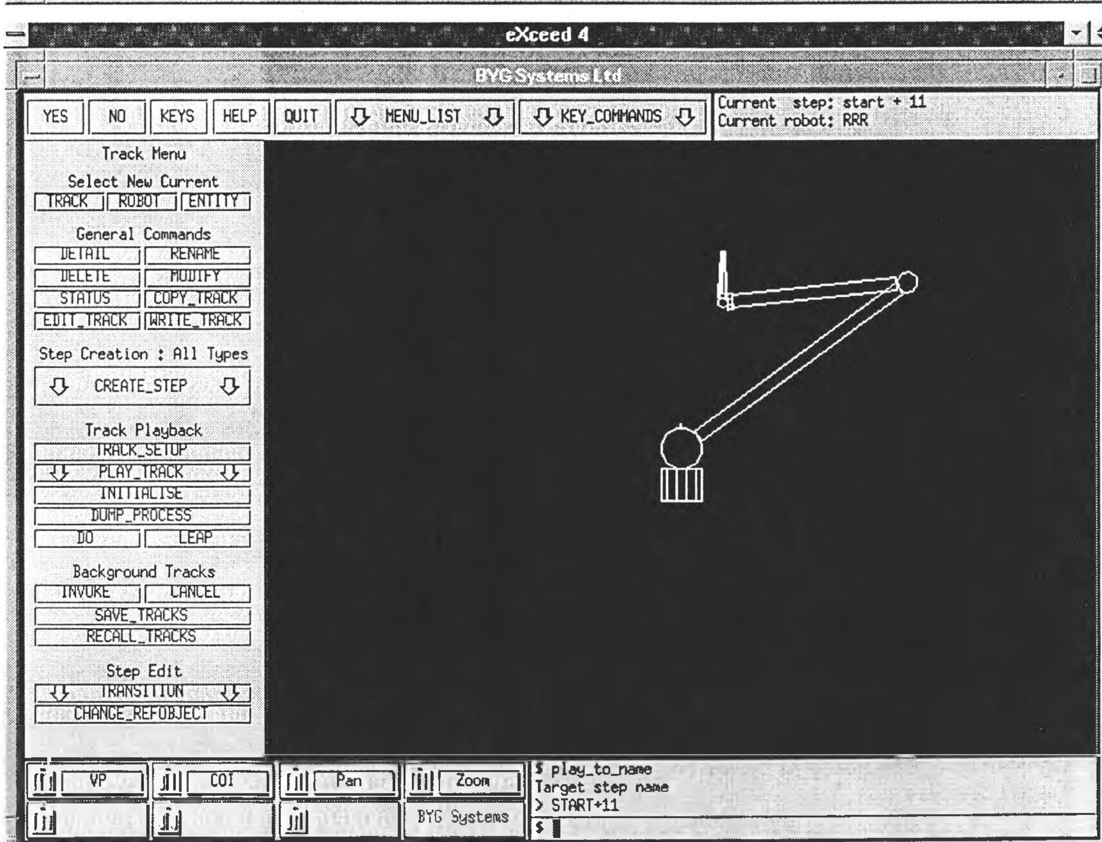

eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 6
Current robot: RRR
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ play_to_name
Target step name
> START+6
$

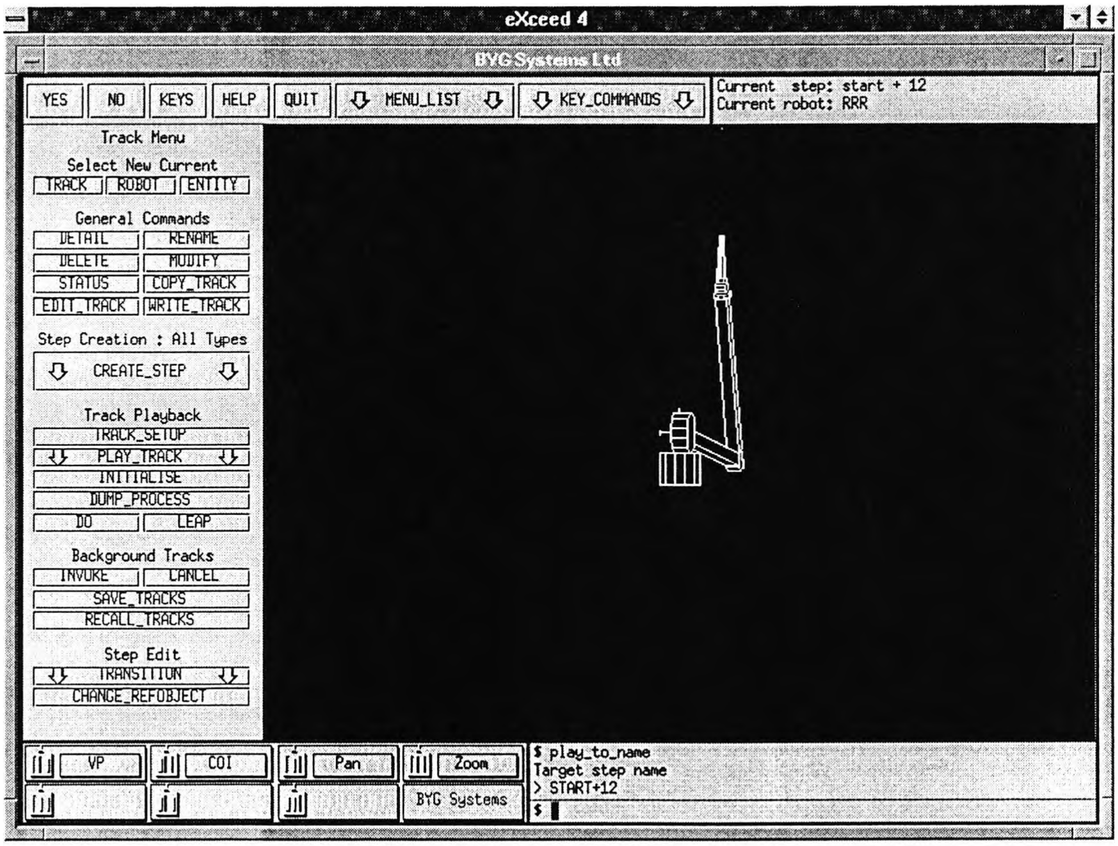

eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 7
Current robot: RRR
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ play_to_name
Target step name
> START+7
$

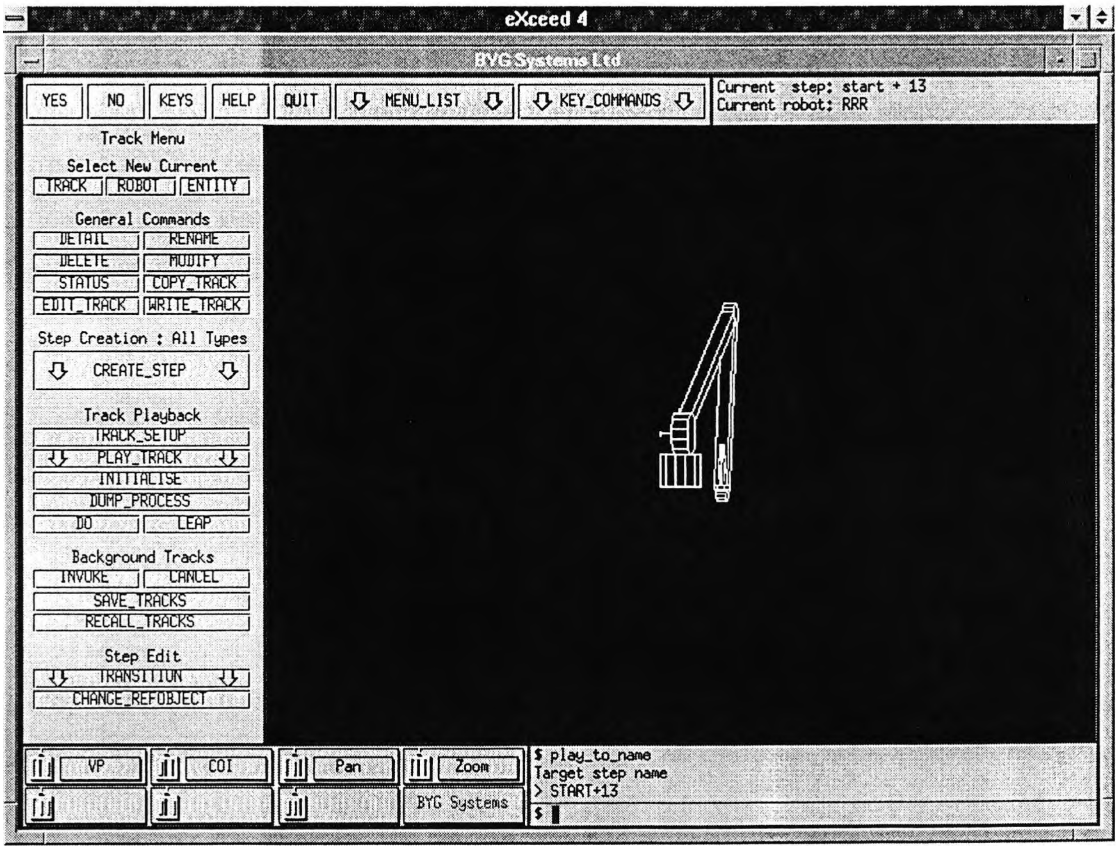

eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 8
Current robot: RRR
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ play_to_name
Target step name
> START+8
$

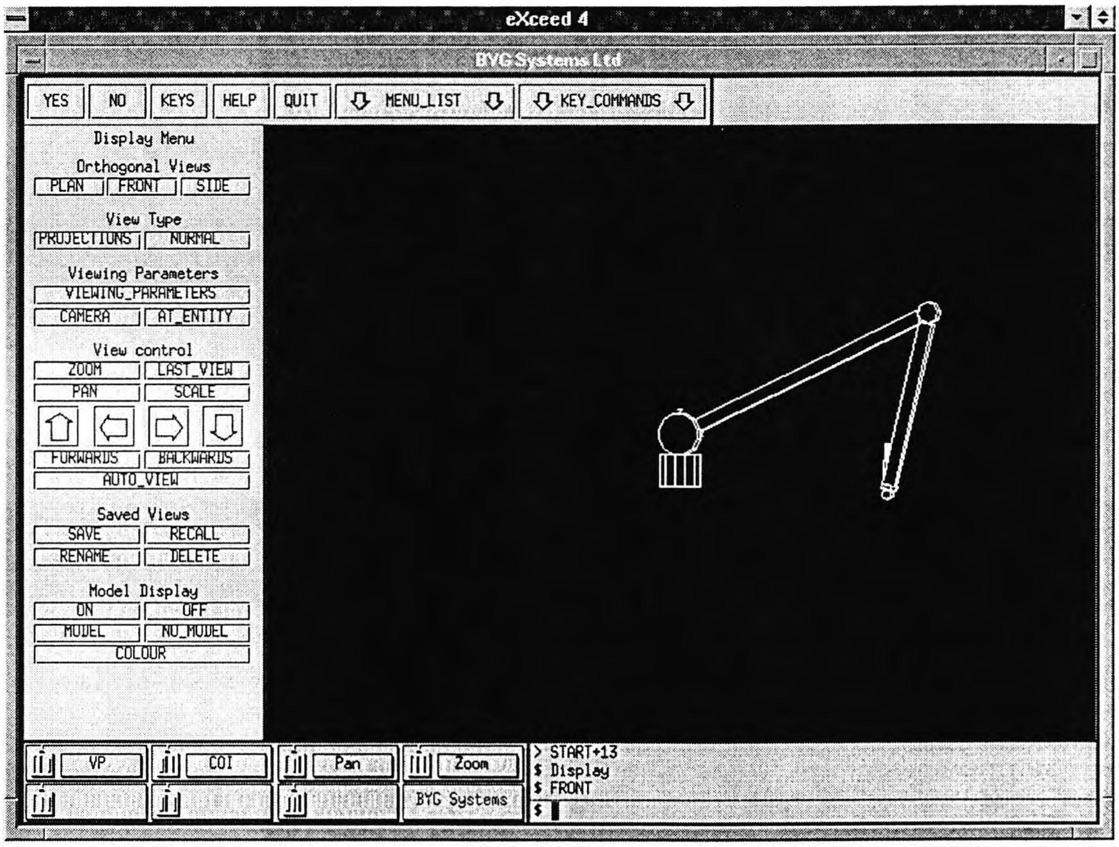

eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 9
Current robot: RRR
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ play_to_name
Target step name
> START+9
$

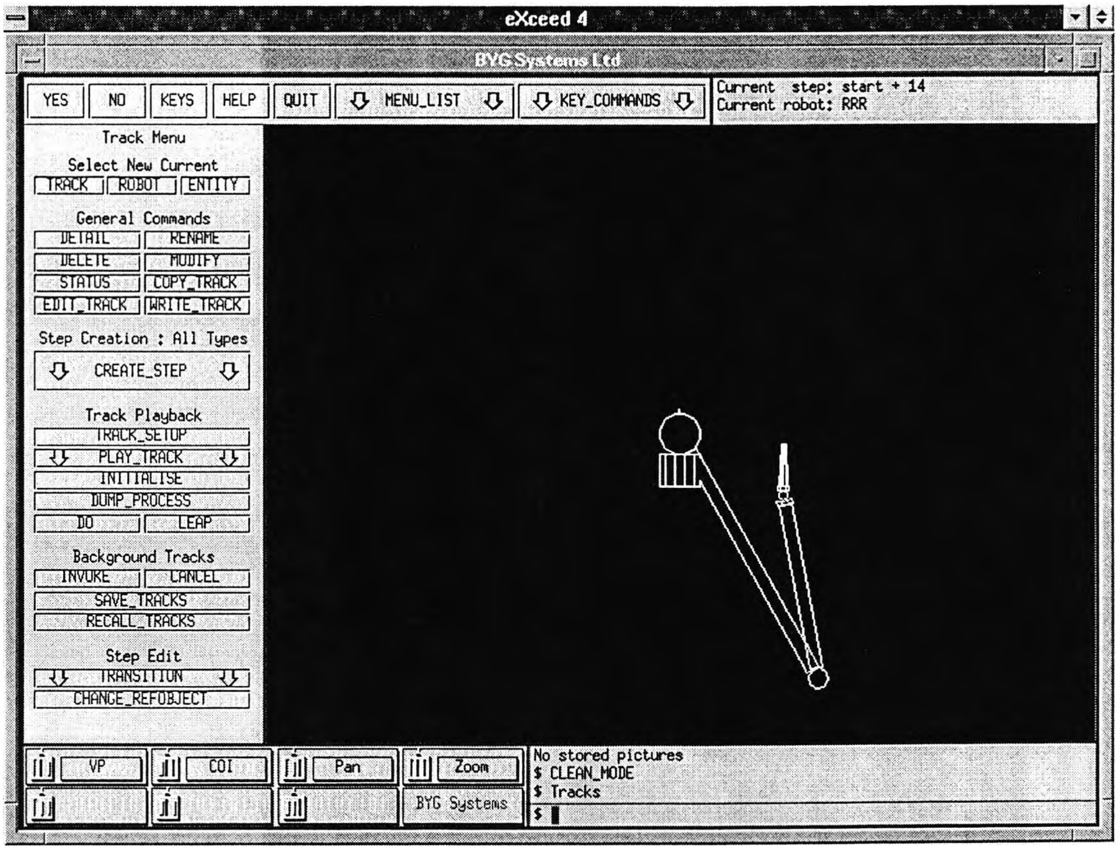

eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 10
Current robot: RRR
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ play_to_name
Target step name
> START+10
$

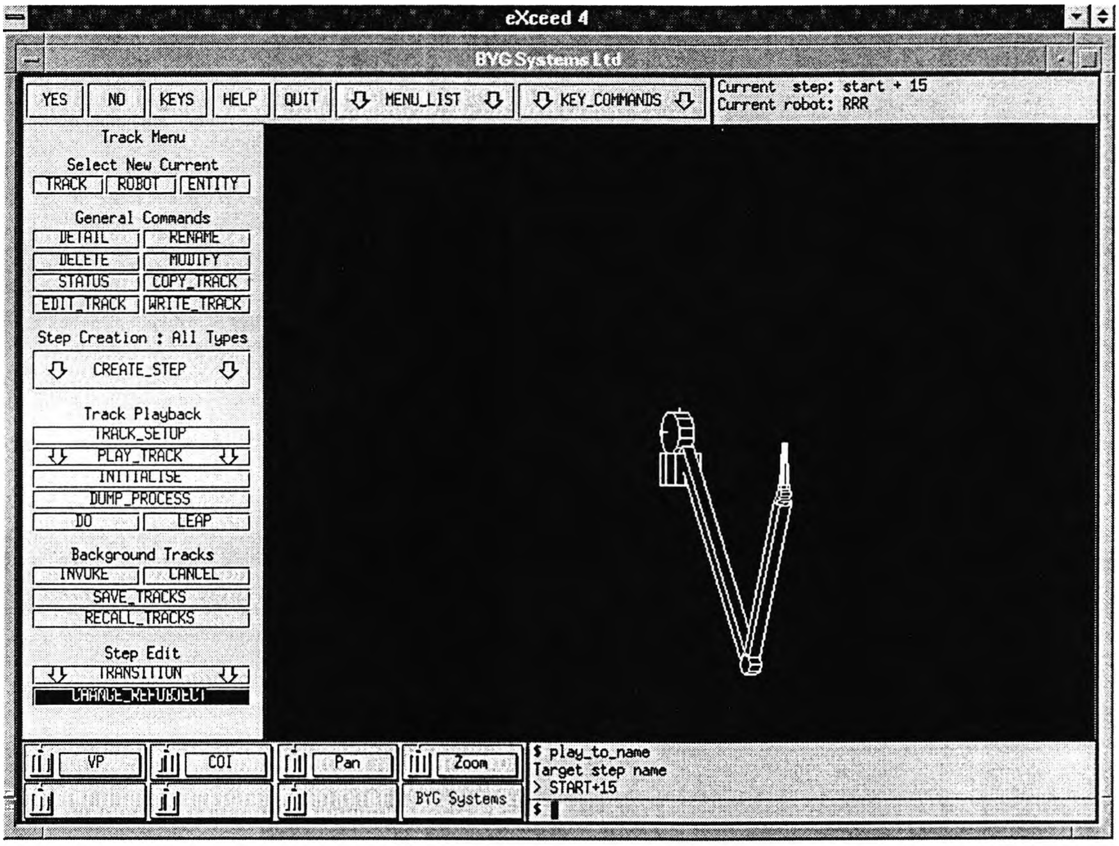

eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 11
Current robot: RRR
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ play_to_name
Target step name
> START+11
$


eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 12
Current robot: RRR
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ play_to_name
Target step name
> START+12
$


eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 13
Current robot: RRR
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ play_to_name
Target step name
> START+13
$


eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Display Menu
Orthogonal Views
PLAN
FRONT
SIDE
View Type
PROJECTIONS
NORMAL
Viewing Parameters
VIEWING_PARAMETERS
CAMERA
AT_ENTITY
View control
ZOOM
LAST_VIEW
PAN
SCALE
FORWARDS
BACKWARDS
AUTO_VIEW
Saved Views
SAVE
RECALL
RENAME
DELETE
Model Display
ON
OFF
MODEL
NO_MODEL
COLOUR
VP
COI
Pan
Zoom
BYG Systems
> START+13
$ Display
$ FRONT
$


eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 14
Current robot: RRR
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
No stored pictures
$ CLEAN_MODE
$ Tracks
$


eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 15
Current robot: RRR
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ play_to_name
Target step name
> START+15
$

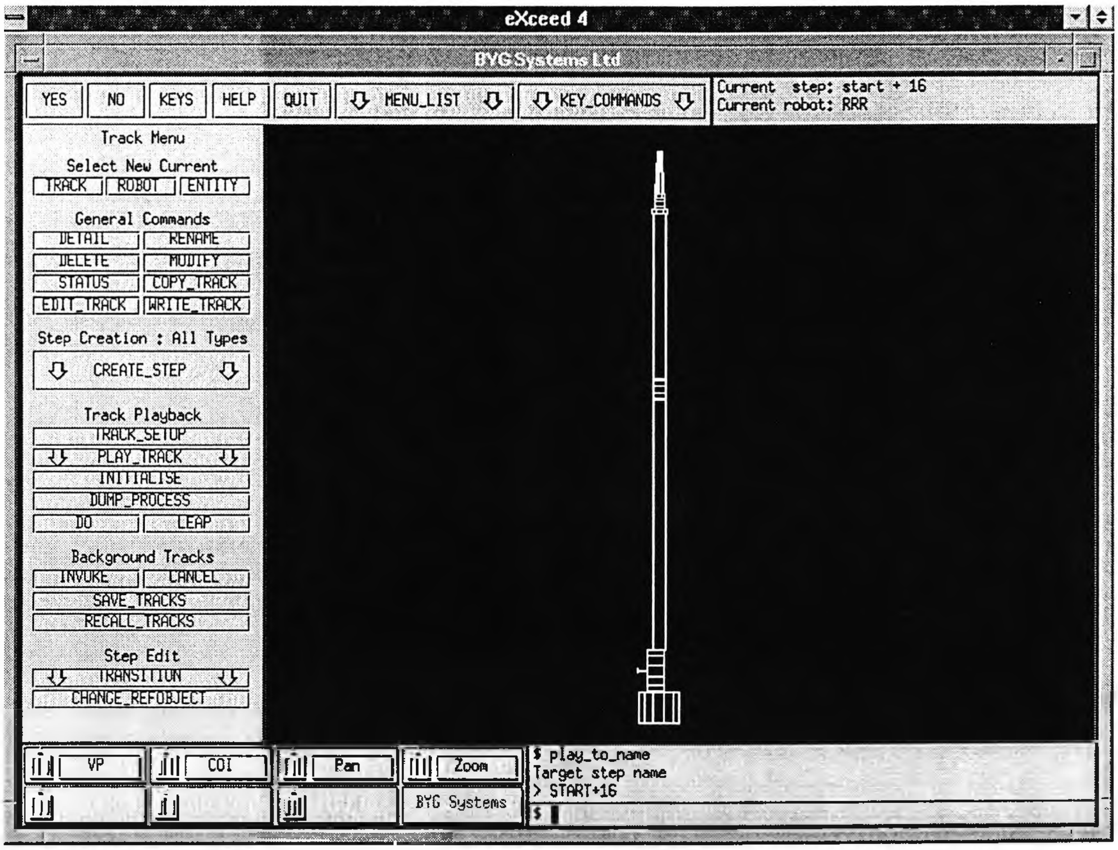

eXceed 4
BYG Systems Ltd
YES
NO
KEYS
HELP
QUIT
MENU_LIST
KEY_COMMANDS
Current step: start + 16
Current robot: RRR
Track Menu
Select New Current
TRACK
ROBOT
ENTITY
General Commands
DETAIL
RENAME
DELETE
MODIFY
STATUS
COPY_TRACK
EDIT_TRACK
WRITE_TRACK
Step Creation : All Types
CREATE_STEP
Track Playback
TRACK_SETUP
PLAY_TRACK
INITIALISE
DUMP_PROCESS
DO
LEAP
Background Tracks
INVOKE
CANCEL
SAVE_TRACKS
RECALL_TRACKS
Step Edit
TRANSITION
CHANGE_REFOBJECT
VP
COI
Pan
Zoom
BYG Systems
$ play_to_name
Target step name
> START+16
$

Figure 10.17 Sequence of 15 Steps of Track of RRR Robot using GRASP

CHAPTER 11 - CONCLUSIONS

11.1 Overview

Restoration (inspection, surface preparation and re-painting) of steel bridges remains a costly and hazardous activity. High cost is incurred mainly through expensive, complex access systems and environmental protection provisions. The large scale and variable geometry of bridges complicates this. The aggressive nature of the surface preparation activity (pollution and abrasivness) adds to the factors, which are typical conditions for the automation application and the level of worldwide research and development points to the requirement for automated systems. Present systems have one major drawback - they are not applicable to the whole structure, due to limited versatility. Therefore, after studying the existing automated and semi-automated devices, market requirements and with support of the literature review, the author identified a gap in this knowledge.

The quality of the design process of the engineering system is more likely to be successful if it follows certain set of rules and steps. These were found in the engineering design process, which the author adopted as a framework.

After further study, the most promising optimisation tool was identified as Genetic Algorithms and selected for use, mainly due to its ability to search large spaces and succeed in multi-parameter and multi-constraint optimisation in other areas of ED. The author has also identified a niche in the continuation through phases of ED, where using the same tool and approach, but with different conditions and constraints imposed through stages, provide clearer evidence of the design development and therefore, possible alterations or detours.

This thesis has presented a new methodology for robot kinematic selection. Although, there already exists a large number of design methodologies, GA is unique in its ability to model complex systems, to automatically evaluate these models and to present the designer with an array of design alternatives, ultimately converging to the optimum. This is done by using a binary representation of the object being modelled, applying GA techniques for model manipulation and optimisation.

The GA approach has been tested on the existing optimised MPIR (from Technion) and then used to aid the selection of the restoration robot for steel bridges. In verification, the methodology provided good confirmation, and was able to come up with a viable solution to the specific design problem (kinematic optimisation). This was indicative of the approach's ability to solve more complex problems.

In the course of the research, contributions in the following fields were worked at:

i. The applicability of the same GA approaches to different stages of the Engineering Design process.

ii. A GA based technique for the simultaneous optimisation of all the robot's kinematic parameters.

iii. The task-based GA technique for evaluation.

This was achieved through the following phases:

i. The engineering design process was adopted as a skeleton for the study of the new system, firstly, in order not to miss any important stages in development, and secondly, to span the developed methodology across the phases of the design process.

ii. Reduced QFD was used to specify the clients', users' and manufacturers' requirements.

iii. The task was analysed, through the research into the variety of steel bridge structures, with synthesis in the graphical form of a benchmark library.

iv. The results from QFD were isolated, hierarchically assembled and transformed into engineering design criteria. Several of these criteria were selected for further simultaneous use in the optimisation.

v. Parameters contributing to fulfilment of the relevant criteria were selected and justified.

vi. Once the methodology was set up, the most suitable tool was selected (GA), developed and applied according to the pre-developed concepts.

vii. Then, the computations have followed and a verification case (MPIR) was used as initial checking point.

viii. Based on the successful outcome, the optimised device was developed for the selected bridge workspace (out of the benchmark library).

ix. The final version was finally applied to GRASP and the general operativness and productivity further confirmed.

11.2 Objectives Evaluation

The following conclusions correspond to the order of stated objectives in Section 1.3. Each conclusion is preceded by the objective, for clarity.

i. *Clarify and apply the engineering design process to the robot's selection and explain the positioning of the optimisation method within ED with the correlation between the expected outcome of consecutive stages and the application integrity.*

Investigation into ED and its role as a framework for this research and constant referral to the procedures and standard methods, made it apparent that the unified procedure within parametric design spanning through conceptual and embodiment stages of design would be of value. The whole of the Section 3.3 addresses this objective. Therefore, the consecutive stages of this research were aimed at this target and hence the objective can be considered as totally accomplished.

ii. *Use the QFD method to assemble a list of customers' and manufacturers' requirements, completing benchmarking and generate engineering requirements.*

Only reduced QFD was applied, however it helped focus on the essential users' and designers' requirements and transformation into engineering design criteria, which can be seen in Section 3.3.1. Also research project limitations such lack of deadline and budget made completing of the clarification of the task phase not fully completed. Although the objective fulfilment is complete within the realistic scope of the project, it is only partially fulfilled in general ED terms.

iii. *Critically analyse the inspection and restoration tasks on large span steel bridges, through combining bridge geometry with the specification of restoration techniques.*

The NDT, in Section 5.2 and restoration tools and procedures, in Section 5.4 and Appendix C.1 were fully investigated and assessed from automation perspective. One of many bridge classifications was assumed and a typical geometry for each representative bridge was combined with the task. The choice of bridge classification as well as choice of one task (blasting nozzle) was the practical limitation. The choice of fifteen model workspaces may seem limited in general terms, therefore I would consider only partial fulfilment of this criterion, although it was realistic within the scope of the research.

iv. *Review existing enabling technologies for possible application in the automated device.*

All possible sources of information and data were exhausted and the most up-to-date compilation of commercially available was assembled and then critically reviewed for possible application in the proposed restoration robot. The objective was fully completed and is reported in Chapter 6.

v. *List, assess and grade the choice of criteria (emerging from engineering requirements) for the optimal design of an inspection and maintenance robot. Analyse and establish design parameters, based on the criteria selection.*

The initial and crucial criteria for any robot's performance were identified as ability to avoid collision with the workspace, singularity avoidance during task performance, percentage of coverage, ability to reach the path points in the tool-required orientation (dexterity), and in the shortest time (productivity) (Section 7.5). All these criteria had to be re-evaluated due to not including the dynamics of the robot into the optimisation. Additionally, they had to be logically and mathematically related to the kinematic parameters. A lot of original thought went into this. The achievement of this objective relied on the choice resulting from the reduced QFD application and although the objective was fully completed, the subjectivenes of the results (only partial QFD and no dynamics included) leaves some uncertainty.

vi. *Analyse and apply Genetic Algorithm (GA) as an optimisation tool, appropriately representing robot's parameters and employing effective evaluation methods, based on the criteria.*

The research into the GA, further reinforced its choice as the optimisation tool for the problem of the optimisation of the configuration and the kinematic parameters of the restoration robot. The development of the functions for optimisation (Eqns 8.1-8.6) was a laborious and risky task, as the verification could only confirm its correctness at the later stages. Major part of the original method development was highly intuitive. I consider this objective almost accomplished, as a specific approach to representation and evaluation was assumed and there is no objective proof or reassurance that the one developed in this research is the most effective.

vii. *Computerise the optimisation procedure with FORTRAN.*

Although the ready-made, basic FORTRAN driver was used, the main effort went to into the design of the evaluation procedure (within GA), which would correspond to the developed method of optimisation. All the parameters for optimisation were assembled into functions (Subroutine "evalout" in Appendix E) and the constraints of the optimisation were reflecting the criteria. The optimisation process was divided into two stages, for computational simplification and ease to observe the performance and interpret results. The software development fully supported the designed model and optimisation approach, so the above objective is considered to be fully completed.

viii. *Test the results on the verified case study of an automated system and based on the success of the verification, apply the method to the bridge inspection and restoration AF (automated facility).*

The verification on the previously optimised robot (MPIR) proved to be successful and yielded similar results to those obtained originally using logical elimination and computer simulation. As the verification showed that the method was producing viable results, it was applied to the prototype model. It optimised the configuration type and the working ranges of the parameters. These are modelled in Figures 9.5 to 9.10. Further tests on alternative tasks confirmed the expected outcome. Using improved GA techniques indicated the routes for improving the GA performance. This objective was completely fulfilled and is supported by the results of the extensive verification.

ix. *Confirm the optimisation outcome using robot simulation software (GRASP).*

The prototype and the workspace were modelled using robot simulation software, GRASP. The design, dimensions and the optimised ranges of the parameters were in-built into the robot's model. The workspace geometry, collision envelope and trajectories were also modelled according to values assumed for the GA optimisation. The robot was able to execute the task without any clashes with the environment, or exceeding pre-determined ranges of the parameters. This further confirmed that the method yielded realistic results. The objective was fully accomplished from the modelling and performance aspects. The shortcoming of the software were not considered or taken into account.

x. *Anticipate the continuation to the method.*

This is addressed in detail in Section 11.5 of this chapter and was fully achieved as an objective.

11.3 Hypothesis Testing

Achieving the objectives set in the beginning of the thesis in Section 1.3 enable to address the hypothesis. The main hypothesis stated in Section 1.4 is: "*It is feasible and beneficial to use customised Genetic Algorithm (GA) as part of the engineering design process, as a technique for generating and evaluating concepts and optimisation of the most task suitable, automated systems, which considers: (i) alternative discrete values of design kinematic parameters, within specified range, for given configuration, (ii) simultaneous changes in values of a number of parameters under specified conditions (criteria), and (iii) mutual influence of these parameters.*".

This is further broken down into three sub-hypotheses as stated in Section 1.4:

- *"The proposed methodology for selecting and optimising an automated device spans over several phases of engineering design process and therefore only requires single framework to carry the out the design into an advanced stage."*
  As the whole approach to selective design and optimisation was based on the ED framework (Ch.3), it became apparent that the approach, could be further expanded and applied through other ED phases, which allow work in a single framework, just changing evaluation conditions, appropriately (Ch.11).

- *"An innovative, task-based evaluation within the GA results in more accurate and at the same time versatile device."*
  In-depth task analysis was assembled from the operational specification of NDT probes and restoration methods and the assessment of their level of automation applicability (Ch.5). This was combined with consideration of variety of geometries of unit workspaces from various bridge types (Ch.4). This allows for precise sequence and location of the end effector positions and therefore valid evaluation against these trajectories (Ch.10).

- *"Possibilities exist to match the existing enabling technologies with the results of the optimisation, which can leads to assembly of the proposed automated facility (AF) from existing components."*
  Research into the existing enabling technologies was carried out in the fields of access systems for hazardous access structures (Ch.6), commercially available five and six DOF articulated manipulators (Ch.6), together with variety of steel bridges'

geometries (Ch.4). This served as a source for constraint setting for the optimisation and was later compared against the GA optimisation outcome (Ch.10) show the feasibility to use the available systems.

11.4 Additional Aspects of Outcome Assessment

In order to assess the practicality and effectiveness of the use of GA into parametric design of the construction robot, the following aspects have been additionally addressed:

i. To what degree were the criteria of the optimisation fulfilled?

The selected criteria covered in the first stage of the optimisation, singularity, collision avoidance and percentage of coverage. The imposed conditions for the singularity avoidance were input according to theory of kinematics of robots and they were complied with. The criterion of collision avoidance was set up according to the rules of geometry and intersection of the straight line and a plane. The planes were isolated from the geometry of the unit workspace on the bridge, so again this condition was also fully satisfied. In both cases, the robot representations were penalised severely for the not complying with the pre-set conditions, and according to the rules of genetic search the non-complying representations did not have any chance of survival. The third criterion for the percentage of coverage was approached from the angle that if the choice of better configuration for the task and relevant motion ranges of the kinematic parameters relating to the preferred major configuration were task-optimised, then it is possible to assess the percentage of manual involvement required to complete the task.

The criteria of productivity and dexterity determined the approach to parameter optimisation in the second stage. And, here the criteria were fully fulfilled. The productivity calculations gave the shortest time for which the task was completed for the simultaneous movement of all the motors, and the result included also the ranges of velocities and accelerations needed for this time scale. Dexterity criterion meant selecting the most suitable minor configuration, and the corresponding joints' motion ranges for the type of manipulation and tool orientation required by the tool.

ii. Do the generated designs accomplish the specified task?

The results of the optimisation were further input into GRASP - robot simulation software, and it was confirmed that the solution was feasible, task execution can be simulated and does not produce any violations.

iii. Were the generated designs related to the predicted ones?

The initially generated results were tested further and the location of another task (and the tool) added. The outcome of the optimisation reflected the expected consequences of the addition and showed the sensitivity of the method. The ranges of the kinematic parameters were optimised according to roughly predicted values, and the remaining aspects of the optimisation process, such as feasibility, time of completion, collision with the environment, were further authenticated by results obtained from GRASP simulation.

iv. How practical was the method (time and complexity of setting up and running)?

The typical stages of operation specified in Table 5.1 were encompassing a range of activities and occupy certain time. The results of the optimisation did not add any significant amount of time or effort once the GA optimisation software was extended and linked to the databases of: (i) the portfolio of the bridge geometries, (ii) operational conditions of the of the inspection probes and restoration tools, and (iii) manipulator's major configurations and their direct kinematics.

11.5 Recommendation for Further Studies

The main thrust of this thesis has been to present a new kinematic design methodology and demonstrate its versatile application in engineering design of a robot. This methodology was tested on an existing interior finishing robot and used to develop a conceptual steel bridge restoration robot. These case studies were selected for two reasons: (i) in the micro-perspective, the similarity of the tasks and (ii) globally, the environmental, human and economic need for their services. In order to be more widely used, this methodology requires further testing and must be applied to more complex problems. In addition, the benchmark library must be further developed to help predict computing requirements for potential applications as well as add versatility to the optimised device.

Further experimentation is necessary to determine if the demonstrated methodology is truly capable of handling versatile environments and a variety of tasks. Future experimentation can be roughly divided into three categories:

The first category includes all the work, which follows directly from that presented here. Model complexity should incrementally be increased to ensure that the methodology continues to function properly. The second is the research, which represents a new application field in which the GA approach can be applied. To effectively apply GA to other problem domains, new tool sets, which allow the methodology to be broadly applied, will need to be developed (object oriented programming applications) and innovative evaluation methods for extended sets of criteria. The third is to further investigate and detail the cross-phase application within ED.

In continuing this body of research, there are many viable courses of action that can be pursued.

11.5.1 Further Testing

One of the first extensions of this work could be to extend the robot's configuration to allow for obstacle avoidance. Introducing redundant DOF could simplify the conditions for collision avoidance and potentially increase the versatility 'unit' workspaces on bridges. Research into redundant DOF, obstacle avoidance and binding them by relevant evaluation criteria would increase the mobility of the robot within more geometrically complex workspaces.

11.5.2 Robot Design

The next phase of development could be to incorporate other configurations than RRR and RRP into the model. This phase was potentially the most difficult, because different configurations will require different evaluation models. But, if successful, this means that the methodology can be applied at the earliest stages of ED, which makes it even more versatile and cross-phase spanning. This may mean setting one model and applying progressively complex evaluation criteria, according to the developing design.

11.5.3 GA Methodology Development

In the course of further developing these models, two fundamental questions need to be addressed: (i) what are the appropriate functions to evaluate and (ii) how are input disturbances handled? For the kinematic parameters, the evaluation-incorporated functions relate to the robot's performance, size and speed. For one of the selected workspaces, the choice of these functions was straightforward. However, in the real world of complex environments and variable tasks, the choice of evaluation functions may be far more difficult.

For the system to be robust, it must be capable of handling input disturbances. This involves extending the research into sensors and dynamics of a robot. Alternative approaches to evaluation will have to be developed, which do not built on the logic of kinematic parameter's evaluation.

11.5.4 Evaluation Methods

One of the central issues in this thesis, has been the function for evaluating the robot. In nature, evaluation is the ability of the organism to procreate. In design, it is the user-defined function. The difficulty comes in determining not only what robot outputs should be evaluated, but how the multiple objectives are to be combined into a single evaluation function. While it is probable that determining general rules to apply to the evaluation of all conceivable robotic configurations may not be possible. Work should be done to determine if some general guidelines could be discovered. One of the limitations of GA is the large computing resource required to solve even moderately complex problems. In addition, as problems tend to become more complex, the size of the design space increases, and it may become necessary to work with larger populations to ensure proper sampling of the design space. Hence, simultaneous research into micro-GA may offer a solution.

11.5.5 Application of Advanced GA Techniques

Research into GA, their schema behaviour and the assessment of the genetic information carried, has developed extensively and in many directions. Setting up the problem and the evaluation method is no longer the sole path to successful optimisation, as more genetic knowledge embedded in GA has been unveiled and therefore, the GA behaviour can be closely monitored and guided in the desirable direction. Additionally, Koza [92] has been very successful in introducing the Genetic Programming (GP) into dynamic optimisation of systems. Therefore, it could be beneficial to look in-depth at alternative selection methods, self-evaluating crossover and mutation probabilities, and a hybrid GA to further improve and even guide the outcome.

11.5.6 Redundancy and Collision Avoidance

Kinematically redundant manipulators possess at least one DOF more than the number of variables that are necessary to describe a given task. Redundancy is a concept relative to the task, assigned to the manipulator. Redundancy can provide the manipulator with dexterity and versatility in its motion. Some of the extra capabilities include the ability to avoid internal singularities or external obstacles while operating entire workspace. Further, if a joint of a redundant manipulator reaches its mechanical limit, there might be other joints that allow execution of the prescribed end-effector motion. The obvious extension to the method was modelling the direct kinematics matrix for 7 DOF or even 8 DOF manipulator with variety of configurations, pre-set variety of tasks on bridge geometry, selecting the best redundant manipulator. A useful robot might operate 6 DOF to position and path and further DOF on force control for example. This could protect sensors and inspected structure.

11.5.7 Long-range Goals

A final area for further research might consider alternate types of genetic operators. In the biological world, the result of a genetic operation produces a viable organism of the same species as its parents. This behaviour is also seen in GA, where the 'organisms' that evolve tend to do so in such a manner as to make the 'organism' more capable of surviving genetic operations. This tendency limits the design space examined by the genetic operators. To overcome this tendency, either a new operator should be developed or some alternative ways of exploring wider areas of the design space employed.

The ultimate design tool would allow the tool itself to evolve. This idea is not as far-fetched as it may seem, as this is what occurs in the biological world. By using the generally sufficient means of representation, such as programs, which represent the means to build a robot, as opposed to representing the robot itself, the future capabilities of GA are extensive.

APPENDIX A. THEORY OF CORROSION

The increasing number of bridges requiring rehabilitation is a common feature and leads to loss or reduction of structural safety. Steel bridges can be susceptible to damage and deterioration from a number of causes, which can be grouped as follows: (i) chemical and biological (corrosion, fire, frost, chemically or biologically aggressive environment), (ii) physical or structural (fatigue, structural deficiencies, faulty design, defective materials, poor workmanship during erection or coating application, poor detailing, poor maintenance), (iii) mechanical (impact, explosion) and (iv) geometrical (geometrics of the roadway as it approaches and traverses the bridge, vertical and horizontal alignment, roadway width and clearance, vehicle sight distance, traffic capacity). Out of all the causes listed above, corrosion is the one which is the most prevailing and therefore, needs further insight.

Corrosion can occur only if steel is subjected, in general terms, to air and water [Burstein, et al., 1994], therefore, the only steelwork that is necessary to protect, in order to maintain its structural integrity, is that exposed to external environments, potential damp conditions and highly aggressive or marine environments. Corrosion can span a range of forms between surface rusting, pitting corrosion, through to complete laminar breakdown of the steel. The rate can vary, depending on the concentration of two contaminants in the atmosphere - chlorides and sulphates.

The first line of protection against corrosion, is the adoption of the design details that avoid water traps and allow for drainage and air circulation, with access for maintenance. The second line of defence is the protective coating to the surface of the steel. While choosing the paint coating, the following aspects must be considered: (i) quality of steel surface preparation, (ii) chemical composition of the paints, (iii) inter-coat compatibility and (iv) subsequent maintenance treatments. The most common and efficient (also cost-wise) maintenance method on a large scale is cleaning the steel surface by shot blasting and repainting it. It is a repetitive process over the length of the useful life of the structure, because corrosion, by its very nature, will cause deterioration from the date of the structure's exposure.

Ever since steel has been applied as structural material and corrosion became associated with it, the study into corrosion, its nature and prevention has grown. The chemistry of corrosion of steel is the formation of hydrated ferric oxide (rust), as a result of an electrochemical reaction between the steel and the environment [Shreir, 1993]. For corrosion to occur, requires the presence of oxygen and water. If the water is pure, the corrosion rate is low, otherwise, it acts as an electrolyte and the corrosion rate increases significantly. Unprotected steel in the presence of oxygen and moisture and in the absence of contaminants, corrodes approx. 0.2 mm per year. Thus, except over very long periods, this type of uniform corrosion on a surface does not have dramatic consequences. However, with steel the concerning factor is that of deleterious effects of corrosion.

A.1 Types of Corrosion

Classification of corrosion depend upon the particular environment, steel is exposed to and in general four classes can be identified: (i) atmospheric, (ii) gaseous, (iii) immersed and (iv) underground. Gaseous type of corrosion is concerned with the formulation of a surface film and the other three rely on the presence of water.

Atmospheric Corrosion

Steel that is freely exposed to the atmosphere receives an unlimited supply of oxygen. The attack depends on the presence of water and the impurities dissolved in it. In London the amount of impurities found in precipitation is exceedingly high, therefore, any exposed steel, such as in the various London Bridges, is particularly vulnerable to corrosion. The dissolved impurities, which are mainly pollutants such as sulphur dioxide which is emitted primarily from the industrial sector, and also salts, which can be found in marine environments, form efficient electrolytes thus promoting corrosion.

In atmospheric corrosion relative humidity is highly influential in predicting both the vulnerability and rate of corrosion. It has been shown by Shreir [1993] that there is a sudden rise in the rate of corrosion above a certain critical humidity. Above this humidity atmospheric pollution becomes the decisive factor. The critical humidity level for serious corrosion is above 70%. Different humidity patterns across the UK show that relative humidity only falls below the critical value of 70% for comparatively short periods during the year. It should also be borne in mind that ambient temperature can also have an affect, as well as diurnal fluctuations in temperature, which wholly determine the incidence and duration of condensation. Lastly, the presence of deliquescent particles can also be highly injurious and corrosion will therefore, take place below the critical value of relative humidity.

Immersed Corrosion

In comparison, immersed corrosion totally depends upon the availability of oxygen which is liable to wide variations of dissolved oxygen in waters. The composition of the water is reflective of the electrical conductivity possible, which will aid corrosion. In some cases calcareous deposits on metals, formed from hard waters can have a protective value. However, the temperature of the water generally controls the corrosive powers of the waters. As temperature rises corrosion will proceed more rapidly. Finally the actual flow force of the water will also be influential. Consider the force of the tidal flow of the Thames, there are two factors that would advance the corrosion rate. Firstly, the supply of oxygen is promoted hence, increasing the rate and secondly, the actual flow rate could prevent the adhesion of protective corrosion products.

Underground Corrosion

Underground the corrosion of metals can take place depending on three processes: (i) electrochemical action, (ii) sulphate reducing bacteria and (iii) stray currents, whereby one or more may be operative.

The nature of the soil is probably the most important factor in the electrochemical action process. Soils, such as sand and chalk because of their permeable nature are termed aerobic, hence a plentiful supply of oxygen, which therefore, means that the appearance of corrosion is more likely. Whereas , anaerobic soils are deficient of free oxygen, corrosion is likely to be much slower (unless sulphate-reducing bacteria are present).

It is probable that corrosion would be localised and intense when in soils of a intermediary nature, because air pockets would be present creating differential aeration currents. The attack of corrosion would occur where the soil presses on the metallic surface when oxygen is within the air pockets. Within the construction industry many air pockets get created artificially when soil is thrown back into a trench after pipe laying, concreting or holes dug to receive steelworks. During backfilling, an entirely different environment from that found in undisturbed soil is created in the soil only a few feet away, such differences may lead to the formation of a current.

Corrosion currents can also be set up by the chemical nature of the soil and in particular its different constituents [Burstein et al., 1994]. Therefore, the acidity or alkalinity of the soil will also affect corrosion. Ground made up of ashes and clinker used for steel pipes or stanchions have been buried requires attention, as the steel is highly likely to corrode at a very fast rate. This is because of the content of water-soluble matter in the soil as it yields electrolytes of low resistivity with the soil water. Together with this, the presence of any unburned carboniferous matter may also promote corrosion, as the wood would act as a cathode of a corrosion cell and being hydroscopic will retain moisture in contact with the metal.

Having mentioned anaerobic soils and their effect of a slower corrosion rate the picture is distorted by sulphate-reducing bacteria in such soils such as waterlogged clays that contain sulphates and organic matter, where they can be highly corrosive because of the organisms present enable sulphates to act as hydrogen acceptors and reductions to sulphides. The corrosion product is a mixture of rust and black iron sulphide.

Metal pipes and structures buried in the soil may act as conductors and pick up stray currents from sources such as power and telephone cables, thus giving serious risk of corrosion by stray currents. Part of the current from the main power line would stray to enter the buried metal and then leave it some distance away to rejoin the main power fine. Corrosion in steel would occur at the loss areas (anodes).

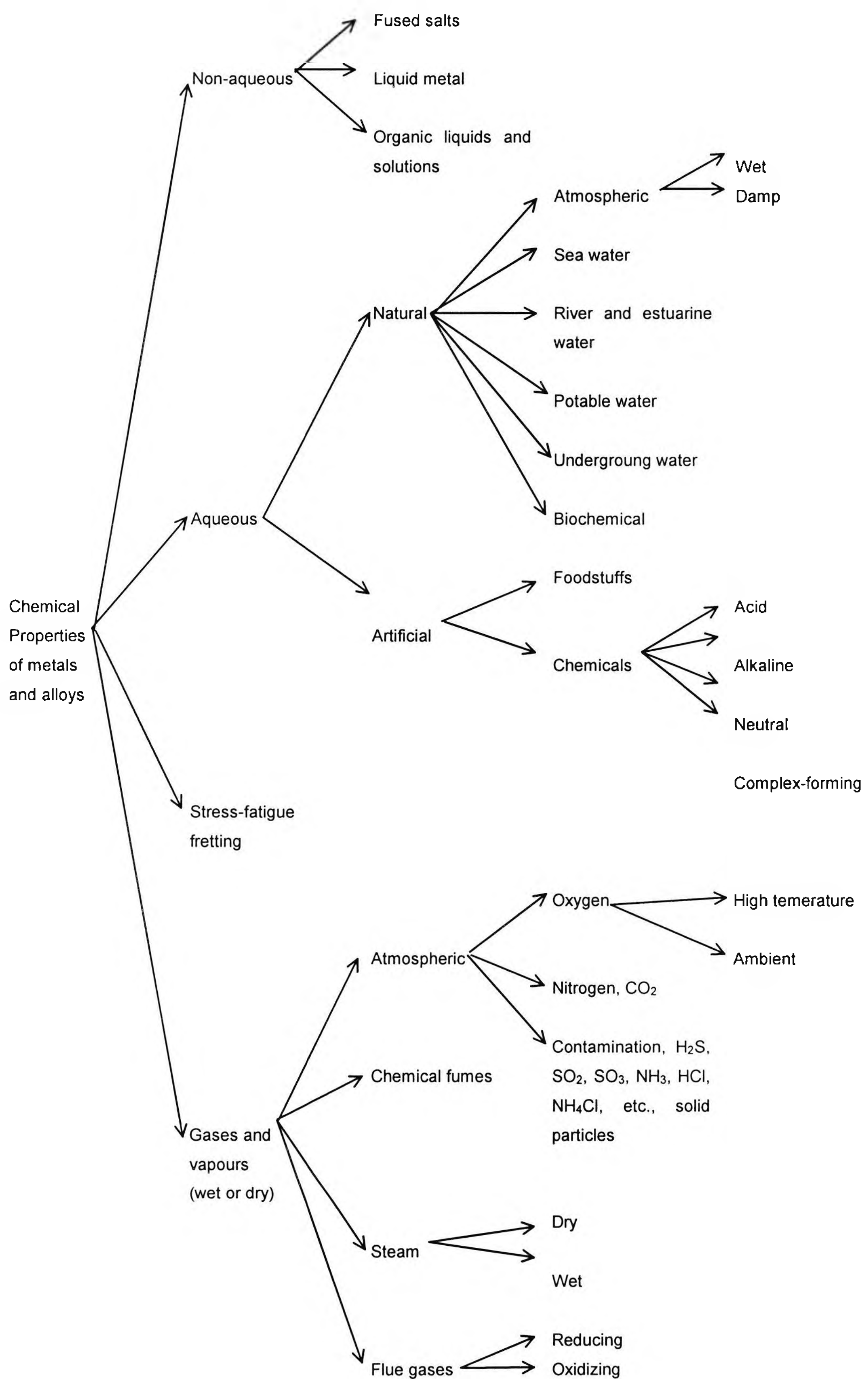

Chemical Properties of metals and alloys
Non-aqueous
Fused salts
Liquid metal
Organic liquids and solutions
Aqueous
Natural
Atmospheric
Wet
Damp
Sea water
River and estuarine water
Potable water
Undergroung water
Biochemical
Artificial
Foodstuffs
Chemicals
Acid
Alkaline
Neutral
Complex-forming
Stress-fatigue fretting
Gases and vapours (wet or dry)
Atmospheric
Oxygen
High temerature
Ambient
Nitrogen, CO_2
Contamination, H_2S, SO_2, SO_3, NH_3, HCl, NH_4Cl, etc., solid particles
Chemical fumes
Steam
Dry
Wet
Flue gases
Reducing
Oxidizing

Figure A.1 Different Environments Causing Corrosion

A.2 Mechanisms of Corrosion

Corrosion of steel is an electrochemical process, which can be compared with the operation of a battery. Anodic and cathodic reactions are produced of equal occurrence and with the magnitude as that of the corrosion rate. The corrosion intensity depends on the initial difference between the anodic and cathodic potentials, the polarization of the anodic and cathodic processes and the ohmic resistance between the anode and the cathode. The anodic and cathodic processes take place mainly in two steps: ionization and diffusion. In the case of the anodic region, the ionization overvoltage of the reaction Fe to Fe^{2+} + 2e-(electrons) and the further diffusion of the Fe^{2+} ions are the normal reactions. The formation of the corrosion product results from the interaction between anodic and cathodic products. Once the Fe^{2+} ions are formed they can be oxidised to Fe^{3+} or hydrated:

$$Fe^{2+} + 2H_2O \rightarrow Fe(OH)_2 + 2H^+ + 2e- \quad (A.1)$$

The ferrous hydroxide is a white product, but in oxygenated conditions this will rapidly oxidise to form, ferric hydroxide:

$$4Fe(OH)_2 + 0_2 + 2H_20 \rightarrow 4Fe(OH)_3 \quad (A.2)$$

As the ferric hydroxide is unstable, it subsequently loses water to form hydrated ferric oxide, Fe_20_3 (red rust):

$$Fe(OH)_3 \rightarrow FeO(OH) + H_20 \quad (A.3)$$

An important characteristic of most of the solid compounds formed by corrosion is that they occupy a larger volume than the metal destroyed in producing them. This aspect is particularly important when metals are connected or embedded, as the expansion accompanying corrosion can lead to the development of forces strong enough to cause breakage. For example where two steel plates are held together by a line of rivets, the failure of which is due to the rust, which forms between the plates. It acts as if a wedge has been driven between them.

Although the steel ionization overvoltage is low various factors can influence the anodic reaction:

(i) a low amount of electrolyte that would make hydration of the anodic products difficult, (ii) the shielding of the anodic sites by their products which also impedes the anodic reaction evolution and (iii) the presence of anodic depolarizers such as chlorides, that will favour the reaction progress.

In case of the cathodic sites, two main reactions are possible - hydrogen evolution or oxygen reduction. Other secondary reactions may occur temporarily, such as the reduction of Fe^{3+} to Fe^{2+}. Hydrogen evolution is the cathodic process but the diffusion of the solvated protons up to the cathode is so comparatively rapid that it does not significantly control the reaction rate. The same principle may be applied to hydrogen evolution due to the reaction of water molecules. Therefore, only the H_2 overvoltage could influence the rate. In case of steel it is established that it has a low H_2 overvoltage, although higher than that of the noble metals.

Oxygen reduction is different, however, because as a neutral gas molecule it has to diffuse through the electrolyte and cross the diffusion layer on the metal surface, before being ionized. Therefore, both steps of the process, diffusion and ionization may be relevant in the case of oxygen reduction.

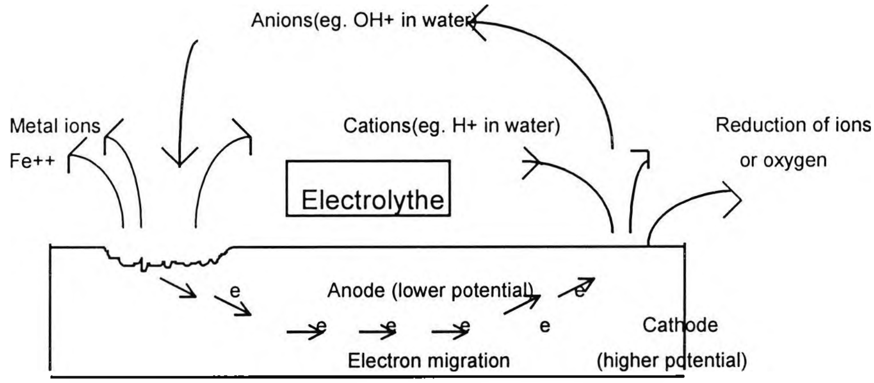


Figure A.2 Simple Corrosion Cell

The theory may be straightforward, but with corrosion there are so many variable factors involved that are likely to influence initiation, course, rate and the final result of corrosion.

A.3 Initiation and Rate of Attack

There are number of factors within the formation of steel that can lead to the appearance of corrosion. Firstly the non-uniformity of steel can increase the probability of corrosion because it will rarely follow the ideal metal lattice with many differences in atoms. Boundary conditions between various grains of the metal may exist giving rise to micro-cells, in which the boundary act as an anode. Areas of unequal stress or deformation give rise to different potentials and are important in the production of galvanic cells with a single metal, such as steel. In general the more stressed parts are anodic and corrode more readily. Variations in stress can be caused by many factors, such as strains or extemal stresses, for example corrosion can result from unequal stress or deformation at bends in steel or heads of rivets.

Attack will also arise where there has been a breakdown of the protective oxide film on the steel. The breaks in the film cause the underlying metal to be exposed, which will then become anodic and hence attacked. Within steel discontinuities of the mili scale are often responsible for intense localised corrosion of the underlying metal. Variations in physical conditions that give rise to the setting up of currents include temperature differences, stray currents and flow of water, which have mostly be covered, however, differences in temperature usually result in the warmer part becoming anodic, hence corrosive.

The basic conditions necessary for corrosion to take place may exist (an anode and cathode), but this does not give the engineer an indication of the rate of attack that may occur. It is perhaps self-evident that this rate of attack will be dependant on the strength of the electric current in any given galvanic cell. There are three factors that must be taken into account to predict the current strength. Firstly there is polarisation, which concems the shifts of potential between the anode and the cathode, however, this does not so much affect steel but mainly the noble metals. Secondly there is conductivity and composition. Conductivity of electrolytes varies widely. Good conductors not only increase the rate of corrosion but it also enables cathodes and anodes to take place in the corrosion process.

Therefore, with the cathodic protection provided by zinc on galvanised steel when the steel is exposed through breaks in the zinc coating is dependant not only on the area of steel exposed but also on the conductivity of the electrolyte. When a small area of steel is exposed, the potential of the exposed steel is polarised to more negative potential at which the ferrous ions can no longer leave the steel. Hence, the steel is cathodically protected, with oxygen reduction taking place on the iron cathode and an increase of corrosion of the zinc anode. When a large area of steel is exposed, protection is maintained only if the electrolyte has a high conductivity.

Finally the ph value of solutions also affect the corrosion rate, as it is a measure of the hydrogen ion concentration. However, this depends whether the metal is noble or its oxide is soluble in acid or both acid and alkali, thus not being totally relevant to steel. Having discovered all the aspects concerning the corrosion rate the actual types of corrosion need to be examined. Different types of corrosion yield different forms of corrosive attack.

A.4 Forms of Corrosion and other Defects

Firstly, 'uniform attack' should be discussed, as this would be most beneficial to the designer as uniform thinning would be considerably easier to allow for. Such attack usually takes place in acid solutions, or in strongly alkaline solutions. Surfaces, which have a uniform deposit, are more likely to have a reasonably uniform type of attack, while deposits of calcium carbonate will reduce considerably the rate of attack. For the designer and the engineer it is probably not wise to rely on uniform corrosion taking place as attack is more commonly localised [Evans, 1981].

A.4.1 Pitting

'Pitting' is one of the more dangerous forms of localised attack. When situated in conditions not suited to steel, for example a wet environment, steel is highly vulnerable to pitting, which may amount to approximately 0.3 mm per year or more. Pitting corrosion can cause a serious reduction in load carrying capacity and introduces a particular risk of fatigue failure. The process is thought to be associated at its inception with a small anode area and a large cathode area, and may be due to variations in the metal, in the surface film or in the film-solution interface. Breakdown of mill scale, a film laid down during manufacture, on steel is a common cause of intense localised attack.

A.4.2 Crevice Corrosion

Crevice corrosion is an inevitable consequence of design in all engineering components, where two sections of steel are bolted or riveted together, they are also potential initiators of corrosion and as such their location and protection must be given special attention in any assembly. Corrosion may commence from a variety of causes. Firstly crevices are not always easy to reach with paint unless the components are painted before assembly - even then scratches during assembly may break down protective measures at a crevice. Paintwork at the edges of a crevice is likely to be thinner and more easily damaged or worn away. Secondly, crevices always retain moisture for longer than flat surfaces, thereby allowing corrosive reactions longer to, attack the metal. Where the corrosion product is porous, moisture may be retained for even longer and as build up of rust occurs, high pressures may be set up in the crevice. This means the crevice may be opened up allowing further and deeper penetration of moisture and in extreme cases the rivets may be fractured.

Thirdly, conditions in a crevice are invariably favourable to differential aeration: the oxygen supply at the bottom of the crevice, whether completely immersed or merely moist, is scanty compared with that on the bulk surface of the steel and therefore, the bottom of the crevice becomes anodic. The result is a typically intense large cathode-small anode attack, which is all the more dangerous because it may proceed unseen for a considerable time if the crevice is very narrow. Crevices may also occur accidentally, as well as being a feature of design; overlying gaskets or washers, stones resting against immersed steelworks, deposited grit in a circulating water system can all give rise to rapid attack.

A.4.3 Stress Corrosion

The combination of corrosion and stress is of great importance since the two factors acting together may cause very much more damage than the sum of their effects acting separately. The visible result of stress corrosion is the spreading of cracks across the specimen roughly at right angles to the axis of stress. The cracks may follow transgranular or intergranular paths depending on the corrodant concerned, and the exact mechanism of crack propagation is still doubtful in many cases. Crack propagation may be principally mechanical with the corrodant merely serving to aid failure at the tip of the crack or it may be a matter of electrochemical corrosion with the stress acting to open the crack so formed and allow free passage of the corrodant to the crack tip. The presence of high-density regions of dislocations has considerable effect on crack propagation. They may act by causing sudden brittle failure over short distances as the crack grows or lead to an intensification of electrochemical action as the crack reaches them. The presence of impurities or a precipitated phase in the grain boundaries can render the steel susceptible to stress corrosion cracking. The effect is particularly marked if, as a result of precipitation, the grain boundaries become anodic to the main body of the grains. Steels may suffer from stress corrosion cracking particularly in concrete, if there is appreciable (>l%) chloride present, and also in hot aqueous nitrate solutions. Basically any form of grain-boundary precipitation or residual stress greatly increases the likelihood of attack.

A.4.4 Corrosion Fatigue

Alongside stress corrosion there is the other corrosion-associated defect of corrosion, fatigue. The mechanisms of corrosion fatigue appear to fairly simple in its essentials. When a metal is subjected to alternating cycling or a fluctuating stress, such as vibrations from cars on a steel bridge, e.g. Wandsworth bridge, it may develop surface cracks which gradually propagate throughout the material resulting in its failure. This is known as fatigue failure and the level of stress at which it occurs decreases with the number of cycles. Some metals demonstrate a limit to this effect, called the fatigue limit, others are not and are progressively weakened until failure occurs at the endurance limit. Even ordinary fatigue failure in air must comprise some element of corrosion, and inspection of such failures reveals local heating and oxidation. Under special circumstances however, the reaction of the environment with the stressed metal may so advance the development of cracks that the fatigue limit is absent and the time to failure much reduced. Fatigue failures generally result in a series of surface cracks at right angles to the line of principal stress.

The essential features of corrosion fatigue are therefore: (i) the coincidence of cyclic stressing and (ii) the presence of the reactivenvironment. Under these conditions the effect of the fatigue failure in creating cracks is then accelerated and propagated by increased chemical activity. Among the factors that cause fatigue failure to be exaggerated are atmospheric contaminants such as ammonia or sulphur dioxide and chlorides in estuarine and seawater.

A.4.5 Fretting Corrosion

Finally another corrosion associated problem is fretting corrosion and this occurs when closely fitted steel or other metals surfaces are subjected to slight oscillatory slip the surfaces often become pitted or acquire a quantity of oxidised debris. Although on the whole mechanical wear contributes to this effect, the more serious incidences occur in corrosive environments. Thus steel will fret by adhesive wear in an inert atmosphere, generating finely divided iron particles which generally lead to seizure. In air, however, such debris rapidly oxidises, and the process becomes one of abrasive wear. Fretting corrosion is generally the result of variations of stress, and it is often associated in the construction industry with some element of fatigue in vibrating environments at bolted or riveted joints.

A.4.6 Effects of Corrosion within the Construction Field

In terms of building, there are many principal effects of the corrosion process upon the much-used metal - steel. The first effect concerns general construction structural soundness. It is generally assumed that corrosion will impair the strength of the component. The extent to which a reduction in strength may be significant will depend on circumstances. In the case of uniform attack, the effects of corrosion on strength may be allowed for, as the corrosion will result in uniform reduction of the thickness. This is a different case for the various forms of localised attack, which on the whole tend to be more frequent. Other things being equal, localised reduction in strength can, of course, have more serious consequences than uniform reduction. When considering the effects of corrosion on the structural soundness of a metal component it is always important to remember that all components are required to be as strong enough to perform their primary functions. For example, steel is used for the major rail bridges over the Thames for its strength under such oscillating vibrations, other examples would be the use of steel for fixings associated with claddings.

Further to the above point, steel is used in construction in conjunction with other materials, therefore, a defect in steel can cause distortion or cracking of other building materials. The products of corrosion are far more voluminous than steel and corrosion is confined to the exposed surfaces of any metal. Consequently, the growth of corrosion products may cause distortion or cracking of other building materials in which the metal may either be embedded or with which it may be in contact. However, the failure of other building materials in this way may lead to more rapid attack of the metal due to the freer access of water or oxygen to the metal and consequently renewed damage on the associated building material. So the cycle may continue making the deleterious effects of corrosion progressively worse. The fact that the rate of destruction is often increased is also extremely significant.

The changes in appearance associated with corrosion are generally unsightly. The surfaces affected may be either the steel or some other building material adjacent but underneath a corroding metal, e.g. brown staining on concrete. Water flowing over the surfaces of the corroded steel transfers some of the corrosion products onto the adjacent material making an anaesthetically pleasing sight, which is almost unremovable or undisguisable. Finally the failure of the component may lead to entry of water into the building, as may occur with steel roof deckings, pipes, engineering services and equipment.

A.5 Resistance to Corrosion

Ferrous metals are used in construction not for their resistance to corrosion properties but because of their mechanical properties, ease of fabrication, relative cheapness and ease of extraction. In most environments ferrous metals have a low resistance to corrosion when compared with other metals [British Journal of NDT, 1988]. In summary this fact can be accounted for by the following factors: (i) the ease with which cathodic reactions can proceed on its surface, (ii) the readiness with which concentration cells are formed and (iii) the poor protection afforded by corrosion products.

There are two main groups of steel identified by many authors, namely mild steel and low alloy steel. Both are classified as low-carbon steels, that is, with a carbon content up to 0.25%: (i) low alloy steel: carbon content up to 0.2 and (ii) mild steel: carbon content up to 0.25%. The role of carbon in steel is important as far as the hardness and strength of the material, but with reference to corrosion the effect is very limited. However, the inclusion of small amounts of certain alloying elements such as copper, chromium and nickel, does increase the resistance of steels outdoors.

Copper introduction onto mild steel increases resistance against corrosion and such steels are generally known as copper-bearing steels. Low alloy steel generally has a carbon content not exceeding 0.2% to which small percentages of alloying elements up to, say 3.0% in all have been deliberately added.

Low-alloy steels and the copper bearing steels are more resistant to corrosion than ordinary mild steel, they are not necessarily immune to corrosion and are therefore, better considered as 'slow-rusting'. The relative performance of some mild, copper bearing and low-alloy steel in an industrial atmosphere and in pure mountain air shows that corrosion is rife in the first year and the beneficial effects of low-alloy additions are of greater practical value in the more corrosive atmosphere.

Another contributing factor to the increased resistance provided by the alloy steels is that the mill scale is more firmly held than on mild steel. The presence of mill scale on the metal surface does not affect the overall corrosion but does increase significantly the localisation of the attack, due to breaks in the mill scale. It has been established that in seawater the penetration of pitting after one year is four times greater with steel with mill scale than with steel without it.

The effect of temperature on corrosiveness of steel is that corrosion reactions do proceed more rapidly as the temperature rises, particularly in water. In moving waters there is a twofold increase initially for every rise of 10°C above atmospheric temperature. At higher temperatures this tendency is reversed, as oxygen becomes less soluble.

It has been mentioned earlier about the effect of water movement, whereby movement often increases the rate of corrosion by promoting the supply of oxygen to the metal surface, while high speeds may also prevent the adhesion of protective corrosion products or calcareous deposits. Additionally the scouring effect of sand and detritus may seriously aggravate corrosion in harbour and estuarine installations.

Finally it should also be noted the effect of cement mortar and concrete, where steel is embedded in or in contact with them is dependant, in addition to the presence of moisture, on a number of factors, which are as follows:

- The highly alkaline nature (p.h. of 12.5) of most cement mortars and concrete is normally sufficient to inhibit rusting,

- Carbonation of the surface layers which results in an increase in volume of one-sixth, may effectively seal the concrete if it is of good quality., in poor quality concrete such sealing does not occur. Thus with dense concrete and with proper depth of cover to the steel, sufficient uncarbonated material will remain uncarbonated to protect the steel for at least 50 years,
- The quality of concrete in which steel in embedded and the depth of cover are important if a crack, and the entry of water, is to be prevented. Entry of water will promote corrosion,
- The presence of chlorides from use of salt-containing aggregates or as additives to accelerate hardening will also promote corrosion.

APPENDIX B. GENETIC ALGORITHM THEORY

This Appendix is not intended as a presentation of the theory of GA, but rather as a guide to familiarity with these techniques and as a discussion forum on the mechanics of GA with some simple examples, to aid the reader. Numerous books exist, that go into detail of the completeness and correctness of GA and the interested reader is referred to these for further details [Goldberg, 1989a). Section B.1 describes the mechanics of GA and section B.2 concentrates on factors influencing effectiveness of GA.

B.1 Mechanisms of Genetic Algorithm

The Genetic Algorithm (GA) is a class of stochastic search algorithms, which borrow ideas from natural evolution and biogenetics. Possible solutions are formed as strings of parameters, with each parameter represented as a gene, which is constrained by minimum and maximum values. Each 'candidate solution' is then represented by a string of genes - 'a chromosome' - or a set of such strings, meaning chromosomes (depicted in Figure B.1). A randomly created number of these solutions form a set of candidate solutions (population) which undergoes a constant transformation, subjected to genetic operations such as reproduction, crossover or mutation (to be explained below) and being ranked for effectiveness (fitness to survive). New generations of chromosomes are formed by randomly selecting pairs of chromosomes - 'parents' - and swapping part of their genes to form a 'child' chromosome (crossover). The selection of parents favourable for crossover is biased to more effective (fit) subsequent parents. Occasionally, a randomly selected gene in a limited number of chromosomes (offsprings) is mutated, which helps to generate unexpected directions in the solution space and allows new genes, or building blocks, which do not appear in neither parent, to be created. The value of each candidate is referred to as its 'fitness', based on the adopted evaluation method and should be characterised by the following: (i) non-negative, (ii) better individuals are assigned larger fitness scores and (iii) the evaluation function should have such a form as to be executed quickly. The first characteristic is needed because of the selection process. Fitness values can be scaled to make negative values non-negative or all negative values can simply be set to zero. The second is needed to promote selection of fitter individuals for reproduction. The selection process will be shown to be a function of the individuals' fitnesses, thus the larger the fitness, i.e., the better the individual, the greater is it's probability of reproducing. Finally, the third is needed for pragmatic reasons, since the evaluation function will be called repeatedly. Along with the parameter coding, the evaluation of the individual members of the population is where the implementers of the GA can really impact the results obtained.

To improve the results of GA, fitness scaling is frequently used. Fitness scaling expands or contracts the range of fitness values to fit some predetermined range. There are numerous schemes for fitness scaling, one of the most common being linear scaling. Using this technique, the maximum fitness is defined to be n times the average fitness and all other fitnesses scaled accordingly. A typical value for n is 2.0. Fitness scaling has the dual advantage of preserving genetic diversity in both the early generations by preventing a small number of chromosomes from dominating the pool of surviving individuals, and in the later generations by preferentially selecting marginally better individuals.

The requirements for the evaluation function are not very different from the requirements for evaluation functions for other types of optimisation algorithms. Where GA radically departs from other algorithms, is how they generate the next points to be evaluated. Calculus based optimisation techniques use knowledge of the functions derivatives to "hill climb" towards an optimal result. GA uses the 'random' contributions of successful individuals in one generation to produce individuals for the next generation. The first step of this process is selecting those members to be used to produce the next generation. Some degree of randomness is allowed in the generation of offspring to allow the weaker building blocks to have a chance to survive in order to avoid premature convergence to false optima.

GA differ from traditional optimisation techniques in four fundamental ways [Goldberg, 1989a]: (i) they work with a coding of the parameters, not the parameters themselves; (ii) they use a population of samples, not a single sample; (iii) they use payoff information, not auxiliary information or derivatives; and (iv) they use probabilistic transition rules, not deterministic ones. The key to the successful implementation of a GA is the parameter coding. (The reason for this has to do with schemata theory, the underlying theory of GA that explains why they work. How they work is through fitness proportionate reproduction, which has been shown mathematically to be near optimal in some senses.)

The rationale behind GA is that candidate solutions, which display higher fitness, will have a better chance to pass on their genes to more members of the new population. Therefore, the more the program iterates the better the chance to generate a near-optimal solution. The measure of success is the convergence of the population, meaning that all the members of the population become identical. However, convergence should not be accepted without conscious assessment, as the population could very often converge on a sub-optimal solution.

gene1	gene2	gene3	....	gene (n-1)	gene (n)

Building Blocks = Genes (i = 1,...,n) = Chromosome

Figure B.1 Chromosome or Potential Solution Representation

Bit strings can be used to encode more than one number. They can incorporate any number of parameters that are necessary for the problem. In this, bit strings are analogous to chromosomes. A group of bits that defines a feature is a gene and the values taken on by the bits are alleles. Different GA may vary in the way they create the mating pool, pick parents, create offsprings, or display their population dynamics. Common parameters for GA are: population size (mating pool size), number of iterations (generations), and number of offsprings generated in each iteration.

The correct balance between exploitation and exploration is crucial to the success of a GA. Exploitation of the historical information of the population is reflected in generating new populations from the old, which leads to improvement in the fitness of the population over the iterations. Exploration of the search spaces, where the optimal solution may lie, is allowed through the introduction of randomness, which in turn prevents the population from immediate convergence to sub-optimal solutions.

Generally the problem of optimising a complex system selection has three parts [Roston, 1994]: (i) selection of means to represent a system in such a way which is amenable to computer manipulation and is able to incorporate all the required range of parameters, (ii) specifying a way to evaluate the performance of the system (means of expressing the desired results) and (iii) using a scheme which searches the object design space (optimises) to find values of parameters for systems that perform well for the required variety of tasks in a minimum time.

The GA process is summarised, using the following algorithm [Goldberg, 1989a]:

```
start
initialise  Population (t=0)                                  {random selection
                                                               from allowable space}
evaluate Individuals in Population (t)
        while termination conditions not satisfied {convergence}
        do {iterative loop}
        start
            t = t +1
            select      Parents (t) from Population (t-1)           {based on their
                                                                    fitness}
            recombine Individuals in Population (t)                 {using genetic
                                                                    operations}
            evaluate    Individuals in Population (t)
        end
end
```

Figure B.2 GA Process

To initialise the procedure, the population is created by randomly selecting members from the allowable space, based on the method of parameter coding used. Then, each of the members in the population is evaluated. Next, an iterative loop is entered in which some members of the population are selected for the next generation based on their fitness. These members are recombined to form the population of the next generation, and finally, the members of the new generation are evaluated. Each step of the algorithm is shown in Figure B.2.

The most common types of GA used are bit-string genetic algorithms. These GA represent the parameters by a binary string. Non-binary representations are possible, but tend to be more complex without yielding any benefit. Binary numbers can be used to represent arbitrarily large integral values. Operating on integral values can represent non-integral values. It is important to note that this method of representation is non-continuous, and it is possible that the optimal answer cannot be exactly expressed using this method - irrational numbers, for example. However, in practice, the required accuracy of the answer is known and a properly set up parameter representation will yield answers of sufficient accuracy.

B.2 Performance factors

Once a GA is designed and ready to run, the implementator is faced with the task of selecting appropriate GA control parameters. Proper choice of the control parameters is necessary to maximise the probability that the GA will produce good results. More about GA techniques can be found in [Goldberg 1989a].

Koza [1992] identifies twelve numerical (quantitative) parameters and six qualitative parameters for use with Genetic Programming. The author selects and describes seven numerical and all quantitative parameters, which are the relevant ones to GA design.

B.2.1 Numerical Control Parameters

B.2.1.1 Population Size

The first step in the genetic algorithm is to determine an appropriate population size (the number of individuals in the population) and then to create an initial population. Population size (npopsiz=N) can be determined by applying theoretical results [Goldberg, 1989b] or can be determined empirically. Typically, a larger population size will yield better results, although, more computation time is required. The composition of the initial population is determined randomly. GA generally do poorly with very small population sizes (N) because the population provides an insufficient sample for most hyperplanes [Grefenstette, 1992]. A large population discourages premature convergence to sub-optimal solutions, but at the same time requires more evaluations per generation, resulting in an unacceptable slow rate of convergence.

Population size (npopsiz) is assumed as 1000 individuals for both, as for large problems a hundred individuals as recommended by [Goldberg, et all 1992], is not enough. A crude population scaling law, based on [Goldberg et al., 1992] and [Carroll, 1996] guided the size of npopsiz:

$$\text{npopsiz} \approx \text{order}[(\frac{l}{k}) \times (2^{k}) \quad \text{for binary coding} \tag{B.1}$$

where, l equals to ***nchrome*** (number of binary bits in the chromosomes), k equals ***nchrome/nparam*** and ***nparam*** is number of parameters.

When the uniform crossover and niching is turned on, this scaling law is usually overkill and can be easily reduced by half. If micro-GA is used, the above law becomes void.

B.2.1.2 Number of Generations (Iterations)

For the number of test/select/reproduce cycles, see Figure 5.1. The efficacy of genetic algorithms can be assessed in many ways. Convergence is one of the most obvious ones, but even Holland, as early as in 1975, realised that convergence is not a useful performance measure, as there is always a danger that the optimum is not the global but the local one. It is impossible to find searching procedures for complex systems that converge to global optima. Such procedures for large finite spaces require already excessively long computing time. To avoid the search algorithm being entrapped in a local optimum, various methods are available, such as (i) improvements to the searching mechanisms, (ii) observing the speed with which the optimum is arrived at, (iii) analysing the efficiency of the fitness function with which it approaches the optimum or (iv) the analysis of the quality of the optimum solution at the intermediate stages.

As the problem under investigation is a complex one, and involves searching large spaces, the approach of monitoring the performance throughout and analysing the current optima is adopted. From the typical form of the evolution curve [Bullock et al., 1995], it is evident that the major improvements tend to occur during the early stages of search. A maximum number of iterations must be chosen wisely, in case the goal is too ambitious. The quality of solution can be improved with more searching, but it is up to the designer to decide how long to wait.

B.2.1.3 Number of Offsprings from Each Iteration

An option for the number of children per pair of parents has been added to the program.

B.2.1.4 Probability of Single Point and Uniform Crossover

Crossover is one of the most powerful genetic operators. The higher the crossover rate (C - random number indicating whether the crossover should be performed) the more quickly new structures are introduced into the population. However, if C is too high, high-performance structures are discarded faster than the selection can produce improvement. With C being too low, the exploration rate will be too slow and the search stagnates.

Choosing at random a single position in both parents performs traditional (single point) crossover and the parts after the crossover position are exchanged to form two new offsprings. Although, biological processes inspire one-point crossover, its algorithmic counterpart has drawbacks, as it cannot combine and protect certain combinations of features encoded in chromosomes. Therefore, different numbers of crossover points are experimented with, by GA practitioners, however, they still cannot link certain combinations of preferred features, so parameterised uniform crossover is introduced [Syswerda, 1989] and [Spears and De Jong, 1991]. Two offsprings are produced out of two parents with each bit position in both children being randomly decided which parent it originates from. An exchange happens at each bit position when the probability (pcross) test is passed. The success of the specific choice of the type of crossover depends, among other factors, on such ones as fitness function and type of encoding. Although, the software has an option for single point crossover, the uniform one is recommended and all the tests are carried out using the latter. Experiments indicate that a crossover probability of approximately 50-60% yields good results. However, using the crossover rate of 95% is not uncommon and it has been widely recognised that there are many fitness functions for which standard parameter settings are not optimal.

B.2.1.5 Probability of Reproduction

Probability of reproduction is 1.0, as we intend to reproduce at each generation (no special allowance needs to be made for this in the program).

B.2.1.6 Probability of Creep and Jump Mutation

Although the crossover is considered the major instrument of variation and innovation in GA, mutation's importance as the tool against permanent fixation at any particular locus is widely recognised. In a simple GA, mutation (second, the most important operator) is the occasional, with small probability, random alteration of the value of the string position and in binary coding it means changing a 1 to a 0 and vice versa [Goldberg, 1989a]. When used with other operators it ensures that premature loss of vital information is avoided. A low level of mutation rate (MR) serves to prevent any given bit position from remaining forever converged to a single value. A high level of mutation yields an essentially random search.

Initial determination to see if a pair of individuals selected for reproduction will be crossed, or an individual will undergo a mutation, is done probabilistically. Experiments indicate that an acceptable mutation probability is in the range of 0.1%-1.0%.

In this thesis, a traditional jump mutation on a binary string is implemented and aided with creep mutation or real number creep [Davies, 1991]. The idea behind the creep operator is that a chromosome that is reproducing is already in a fairly good position in relation to other members of the population. What is needed, is just a small browse around the current position to see if a movement nearer the optimum can be detected. The creep mutation moves along the chromosome, creeping up or down each parameter by an increment, by which the parameter array is increased. This is achieved by converting the binary encoding into a real number, creeping and converting back. It is a good practice to have the same number of creep and jump mutations and this happens when:

$$\text{pmutate} = \frac{1}{\text{npopsiz}} \tag{B.2}$$

$$\text{pcreep} = \frac{\text{nchrome}}{\text{nparam}} \times \text{pmutate} \tag{B.3}$$

where, ***pmutate*** is the jump mutation probability and ***pcreep*** is the probability of creep mutation. Creep mutation probability equals zero with micro-GA.

B.2.1.7 Probability of Decimation and Decimation Percentage

Decimation is the operation of removing some portion of the population from further consideration. This can be useful in the initial population has a large number of members that have poor fitness. To use the decimation operation, an initial population larger than the actual population is generated. Immediately after the evaluating the initial population, fitness proportionate selection is performed to select the members of the actual population. The author seeks improvement in performance using alternative techniques (elitism, niching and micro-GA).

B.2.2 Qualitative Control Parameters

B.2.2.1 Generating Initial Population

The initial population is randomly selected.

B.2.2.2 Selection Mechanisms

Genetic evolution is greatly influenced by not only the way the individuals are introduced into the population, but also the way the interesting individuals are kept inside the population. Selection methods influence preservation of these individuals and also correspond to nature and speed of convergence [Neri and Saitta, 1995], [Thierens and Goldberg, 1994]. There are a number of selection algorithms commonly used, with three basic approaches: (i) stochastic sampling, (ii) deterministic sampling and (iii) mixed sampling. In stochastic sampling selection phase determines the actual number of copies that each chromosome will receive, based on its survival probability and best known in this class are proportionate reproduction (weighted roulette wheel) and ranking selection. Deterministic sampling usually selects the best chromosomes from the selection space, the generational part or full replacement is another version of the deterministic approach and the classics here are: Genitor by Whitley [1989] or “steady state” selection by Syswerda [1989]. Mixed sampling contains both random and deterministic features simultaneously and a typical example in this group is tournament selection. The author uses in her GA driver the tournament selection (as micro-GA is used).

B.2.2.2.1 Weighted Roulette Wheel

The most basic selection algorithm is stochastic sampling with replacement. To visualise this scheme, imagine a weighted roulette wheel that is partitioned according to fitness of the individuals, as in Figure B3, below.

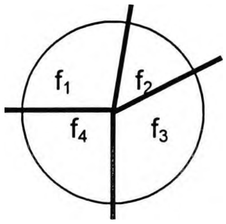


Figure B.3 Weighted Roulette Wheel

'Spinning' this wheel will yield, probabilistically, a higher percentage of those individuals with higher fitnesses and a lower percentage of those with lower fitnesses. The wheel is spun N times, where N=npopsiz. On each spin, the individual under the wheel's marker is selected to be in the pool of parents for the next generation. The problem with this method is that it is too random, and non-representative populations are frequently observed. A scheme that yields better results is called remainder stochastic sampling without replacement [Goldberg, 1989a]. The individuals comprising the next generation are found by first calculating the expected number of copies that the individuals in the current generation are expected to contribute. Then, the integral parts of the expected contributions are assigned and the rest of the succeeding generation is found probabilistically using the remaining fractional parts. Therefore, rather than spin the roulette wheel N times to select N parents, stochastic sampling spins the wheel once, but with N equally spaced pointers, which are used to select N parents.

Fitness proportionate selection early on, often puts too much emphasis on exploitation of highly fit strings at the expense of exploration of other regions of the search space.

B.2.2.2.2 Tournament Selection

The competition for producing the next generation is achieved through binary tournament selection with a shuffling technique for choosing random pairs for mating. Pairs of individuals are chosen randomly from a population and the better out of the two is selected with fixed probability [Goldberg and Deb, 1991]. In this implementation, each generation has the same size as the original one and if the best individual from the previous generation is not copied into the new one, a random member is replaced by it, unless the elitist option is switched on.

B.2.2.2.3 Ranking Selection

Baker [1985] introduced the notion of ranking selection to Genetic Algorithms. The population is sorted out from best to worst and ranked. The number of copies, which each individual should received, is assigned according to a non-increasing assignment function. Then, the proportionate selection is carried out according to that assignment. The expected value of each individual depends on its rank rather than on its absolute fitness. There is no need to scale fitnesses in this case, since absolute differences in fitnesses are obscured. Discarding of absolute fitness information has both advantages - avoiding premature convergence, for example, and disadvantages - it may be important to know that one individual is far fitter than its nearest competitor. Ranking avoids giving the greatest share of offspring to a small group of highly fit individuals, and therefore reduces the selection pressure when the fitness variance is high. Rank selection is similar to tournament selection in terms of selection pressure, but is computationally less efficient and less amenable to parallel implementation.

B.2.2.2.4 Genitor and "Steady State" Selection

Genitor selection [Whitley, 1989] works individual by individual, choosing an offspring for birth according to linear ranking, and choosing the currently worst individual for replacement. Only a few individuals are replaced in each generation, usually a small number of the least fit individuals by offspring resulting from crossover and mutation of the fittest individuals. Steady state selection is often used in evolving rule-based systems in which incremental learning is important and has been analysed by De Jong and Sarma [1993]. Informal testing and comparison of steady state and any of the previous three (generational) approaches, indicate that at least for some problems, 'steady state' GA find as good or better solutions in much less time. In the program one member is reproduced in each generation. This is done by selecting a member according to its fitness and making a copy. In order to insert a copy into the population, a randomly selected member is deleted. To compute the ideal values, each member in the population is increased in number according to its chance of being selected for reproduction, and decreased according to its chance of being deleted.

B.2.2.2.5 Application Method of Ranking Selection Methods

While applying micro-GA, the selection strategy had to be a deterministic one.

Since the population is so small, the law of averages did not hold well and the selection strategy is kept purely deterministic. In the tournament selection strategy, the strings are grouped randomly and adjacent pairs are made to compete for the final positions in the next generation. The tournament selection results in the same outcome when is used on local selection pools after [Sarma and De Jong, 1997] and this particular feature is necessary when using micro-GA after [Thierens, 1997].

If, however, use of the micro-GA is suspended, there is the alternative group of selection schemes based on ranking. The most promising [Whitley, 1989] is the Genitor algorithm. This approach suggests that allocating reproductive trials according to rank is superior to fitness proportionate reproduction, and provides the degree of control, which is not possible with proportionate reproduction. Ranking acts as a function transformation, that assigns a new fitness value to a genotype, based on its performance relative to other genotypes. A well-tested mechanism by Whitley [1989] suggests the following sequence of steps in order to implement Genitor:

1. Calculate the bias of the population:

$$\text{bias (selective pressure)} = \frac{\text{best fitness of the population}}{\text{average fitness}} \tag{9.1}$$

2. Sort out members of the population depending on fitness.

3. For bias up to and including 2.0, a linear function is used to allocate reproductive trials. The function suggested by Whitley [1989] has the following form:

$$\text{index} = (\text{population size}) \times \frac{\dfrac{\text{bias} - \sqrt{\text{bias}^2 - 4.0(\text{bias} - 1) \times \text{rand}}}{2.0}}{\text{bias} - 1} \tag{9.2}$$

where rand returns a random fraction between 0 and 1.

4. For selective pressures greater than 2.0 a non-linear allocation of trials is used. A selective pressure of X implies that X% of all reproductive opportunities go to the top ranked position in the population. X% of the remaining (100%-X%) are given the second ranked position etc. Any residual opportunities are evenly distributed.

B.2.2.3 Method for Selecting Second Parent

Same as the method for selecting the first parent: random selection.

B.2.2.4 Fitness Scaling

To improve the results of GA, fitness scaling is frequently used. Scaling method maps the real fitness value to expected values in order to make GA less susceptible to premature convergence. Fitness scaling expands or contracts the range of fitness values to fit some predetermined range. There are several mechanisms of fitness scaling, the most effective one is linear scaling. The maximum fitness is defined to be n (typically, 2.0) times the average fitness and all other fitnesses are scaled accordingly. Fitness scaling has the dual advantage of preserving genetic diversity in both: (i) the early generations by preventing a small number of chromosomes from dominating the pool of surviving individuals, and (ii) later generations by preferentially selecting marginally better individuals.

B.2.2.5 Over Selection

Over selection is a means to select a higher proportion of fitter individuals from the population. Koza [1992] uses this method for problems that have large populations. Over selection is implemented by first rank-ordering the population, dividing the population into two groups, the first group containing certain percent of the fittest individuals, and the second group containing the rest of the population. During selection, a higher percentage of the individuals are taken from the first group than the second. This is not implemented for GA since the selection method used does not have some of the problems associated with the "pure" fitness proportionate selection method of sampling used by Koza.

B.2.2.6 Elitism

In each generation, some small number of the best individuals in the population can be copied directly into the next generation (they are subject to mutation, which is typically a low probability). This will tend to make the maximum fitness monotonically increase with time, as the best individuals will persist. Elitist reproduction on number of the individuals is, therefore, added in order to ensure that the best member of the population produces offspring in the next generation. The elitist strategy fixes this potential source of loss by copying the best member of each generation into the succeeding generation. The elitist strategy may increase the speed of domination of a population by a super individual, but on balance it appears to improve genetic algorithm performance.

B.2.3 Values of Control Parameters

The values of the control parameters are given for each experiment or group of experiments. Although the preferable approach would be to enable the GA to modify its own parameters dynamically during the search, the number of evaluations which can be performed in a reasonable amount of time would not allow the GA enough evaluations to modify its search techniques to any significant degree [Roston, 1997].

APPENDIX C. STEEL SURFACE RESTORATION TOOL AND SYSTEMS

C1. Commercial Steel Surfaces Restoration Tools

Nelco Manufacturing Corporation [Nelco, 1993].

E10-10 APTDC (Hepa type filter) - surface preparation, two cell (blaster and dust collector) system with 250mm wide blast strip, power - electric (10HP 230v 3 phase motor for the blaster and 5HP 230v 3 phase motor for the dust collector), requires 50mm distance from the obstacles, self-propelled forward / reverse, 15m duct hose and 30m electrical cable, blast head's size - 1000mm L x 500mm H x 600mm W and weight 280 KGs, dust collector's size - 1400mm L x 1700mm H x 800mm W and weight 220 KGs, application - downwards only (suitable for metal decks).

AC7-4 Deck Blaster - two cell unit, powered by 4HP pneumatic motor, able to blast 25mm away from the obstacles, uses steel grit, 180mm wide blast strip, blaster's size - 500mm L x 600mm H x 200 W and weight - 66 KGs, application - downwards only.

JHJ-2000 "Hand Held" Vertical - 50mm wide blast strip, 0.5HP electric or pneumatic motor, horizontal, vertical and overhead hoppers included, weight - 5 KGs, application - perpendicular to the surface of unrestricted orientation.

EV7-2 Vertical - 180 mm blasting width, available with 2HP electric or 4HP pneumatic drive motor, remote control blast head, size of the blast head - 70 mm L x 25 mm W x 45 mm H and weight - 40 KGs, suitable for vertical or overhead application, easily mountable on any mobile and lifting vehicles.

EV-15-30 Vertical - designed to clean 45 angle of a ship hull, 40 mm blasting width, 30HP electric motor, able to blast vertically or horizontally, size - 70 mm L x 60 mm W x 180 mm H and weight 625 KGs, easily mountable on any mobile and lifting vehicles.

Trelawny - Pneumatic Tools [Trelawny, 1994]

Specialises in development of the pneumatic percussion tools. Delivers complete TVS (Tool and Vacuum Systems).

Vacuum Type 5310 - Cylindrical drum of height - 600 mm and weight - 8 KGs, with recovery capacity - 12 litres, electrically powered, HEPA filters.

Vacuum Type 7310 - Cylindrical drum of height 1000 mm and weight - 32 KGs, with recovery capacity - 57 litres, electrically powered, HEPA filters.

MCV (Multi Component Vacuum) - portable, 18 litres HEPA vacuum, 210 litres drum adapter capable of running 3 tools simultaneously.

Accessible tools include:

Needle Scalers (weight - 1.5 KGs), perpendicular orientation to any surface.

PPT Peening Prep Tools - width of the cleaned strip - 50 or 100 mm, 1.1HP motor, weight - 4 KGs, perpendicular orientation to any surface.

SF 11 Shrouded Floor Scaler - width - 300 mm, 25 mm strip around obstacles, weight - 38 KGs, perpendicular and downward orientation to the surface required.

SRA Shrouded Right Angle Grinder - 1.2HP power source, 125 mm wide cleaned strip, weight - 3.2 KGs in weight, perpendicular orientation to the surface - essential.

3M - Abrasive Systems Division - [3M, 1998]

Heavy Duty Roto Peen - leaves a bare metal surface with 3-5 mill anchor pattern, does not require complete containment, 2" or 4" widths.

Scotch - Brite Coating Removal Discs - dia. 2" to 8", max. operating speed 4500 to 18000 RPM, discs can be used on right angles, mini angles and right angle die grinders.

De Kleijn B.V. - [De Kleijn B.V., 1994].

DKV 2005 Vacuum and Abrasive Recovery System - air powered vacuum system for the removal of dust and abrasives. Developed to work with blast vessels. vacuum Filter Unit - height 220 cm, pressure - 7.5 bar, empty weight - 130 KGs, air requirement - 5 m^3 /min., diameter vacuum hose - 7.5 cm. Dust separator - vacuum hose up to 30 m long, height - 69 cm, weight - 27 KGs, for use in combination with 150 and 200 litre blast vessels. Container - height - 160 cm, weight 98 KGs.

Dust Filtering Unit DKO 35000 - mobile filtering unit used on mobile, covered grit blast jobs. Capacity 35000 m^3 /h., dust emission - 99.9%, max. dust release - 3 milligram /nm^3, noise level 79 DBA, statistical pressure - 1200 Pascal.

Flow Europe (UK) - A Flow International Company - Flow UK, [1998]. manufacturer of the ultrahigh-pressure waterjet equipment providing thermal spray coating removal and precision cleaning of metal surfaces.

Ultrahigh-Pressure Waterjet Pump (Husky E-150) - works at pressures up to 2800 bar and flow rates up to 24 lpm, electric powered - 150 horsepowered, mobile, weight 2040 KGs, water filtration to 10 micron absolute.

Ultrahigh-Pressure Waterjet Pump (Husky S-200) - works at pressures up to 2800 bar and flow rates up to 24 lpm, 205 horsepower model 3306DIT engine, 6 cylinder, turbo-charged, 4-cycle direct-injected, speed 2100 RPM, weight (without fuel tank) - 2770 KGs, mobile.

Ultrahigh-Pressure Hand A-3000 - lightweight, portable, multi-purpose tool, rotating a variety of multiple-orifice tips at the rotation speed of up to 3000 RPM, tool delivers up to 20 lpm of ultrahigh pressure water. It is rated for operating pressures of up to 3100 bar. Because it is pneumatic, it can be powered by the on-board air compressor of the Husky pump. Weight - 5 KGs, operating pressure up to 3100 bar, motor horsepower - 0.5 kW.

Ultrahigh-Pressure Hand Tool Jetlance 5062 - hand-held, area cleaning and material removal tool, tips with multiple orifices at various angles, rotating at up to 1500 RPM deliver up to 27 lpm of high pressure water. Operates pressures of up to 3000 bar, weight 11 KGs, length 54 cm.

Ultrahigh-Pressure Hand Tool Jetwand 5060 - hand-held, lightweight tool for cutting and removing tough coatings from surfaces. Operates at pressures up to 3000 bar, max. flow rate 26.5 lpm, total length 124.5 cm, weight 8.4 KGs.

C2. Existing Automated and Partially Automated Systems

Several maintenance companies worldwide as well as other academic institutions assembled more or less successfull systems to solve restoration problems on large steel structures. These systems are being reviewed below in order to assess their overall applicability.

ALPHA 100 by Ipec Inc. is a mobile unit with two blasting outlets, continuous grit classification and an air drying system coupled with vacuum system. Recommended applications include surface preparation on water tanks, power plants, bridge overpasses, etc. [Ipec, 1998].

The Aqua Blast 2500 Plus System consists of water diesel-driven package mounted on a four-wheel trailer and is designed for single gun operation with twin intensifiers. The system is manually operated and does not provide effluent containment [Aqua Blast, 1998].

The **Auto-Blaster**, manufactured by D&S Services, Inc. of Kentwood, Michigan, is designed for the abrasive blast cleaning of particular type of bridges (beam type with main I sections along the edges), as the work platform is suspended from the flanges. Eight blast nozzles are operating from the platform in three modes:

manual, remote and automatic control.

The **Laser-Vac**, developed by Valley Systems Inc. utilises the water jet and is operating from the mobile platform controlled by motors connected to cables, fixed to the roof of the structure. Horizontal movement is controlled by a rack and pinion tracker system that is attached to two horizontal cables fixed to the platform.

The **LTC Robotic Unit** operates using a blast nozzle, held by a manipulator, high automation level allowed, as the location and size of obstacles can be pre-programmed. Nozzle is positioned within a vacuum head [Cignatta, 1993].

Nelco System is designed to blast clean storage tanks and ship's hulls using vertical, remote control blast head, off hydraulic powered crane [Nelco, 1993]. The cleaning productivity is given as 280 square meters per hour.

Sandroid Systems Inc. has developed a system capable of blast cleaning, inspection and painting of steel surfaces. All processes are performed with 35 m reach articulated arm fixed to the base vehicle. The supporting vehicles can carry the blast pot, compressors, vacuum system and other equipment, as required for a particular application [Sandroid, 1993]. The precision control of the end effector and the containment for the blast debris are still to be resolved. Paint removing productivity is given as 150 square meters per hour.

The **Ship ARMS Depainting System** by United Technologies is developed to depaint large ships. The system consists of a high-pressure, high flow water pump, a teleoperated transporter with 5 DOF (degrees of freedom) telescoping arm, a 6 DOF manipulator with specialised end effector and a precision, computer designed 6 inch wide waterjet nozzle in a frame for precise application. The system also removes the process effluent through vacuum recovery shroud and ports, leaving a dry and rust-free surface. Cleaning and capture productivity is given as 15 square meters per hour [United Technologies, 1994].

DKS 7116 Vacuum Blast Machine by De Kleijn B.V. [De Kleijn B.V., 1994] - the used abrasives, paint and corrosion particles are vacuumed back into the machine, where the re-usable grit is separated from the paint/corrosion dust. Cleaning capacity of the flat steel surface rusted to grade B - 4-6 m^2 /h., cleaned to SA 2.5. Cleans weld tracks to SA 2.5 - SA 3, 5.5 cm wide 1-1.5 m /min. Height - 200 cm, hose length - up to 30 m, empty weight - 320 KGs, blast media capacity - 40 litres, nozzle diameter - 6.3 mm.

Blastman B10 and Blastman B20 by Rautaruukki [Rautaruukki, 1998] - cleaning capacity 100-150 m^2 /h. for finish SA 2 and 70-100 m^2 /h. for surface finish SA 2.5 and rusting degree A. Both systems include the same main components: frame, dust-proof telescopic boom, driving machinery hydraulic unit, energy transfer system and electric and control switchgear cabinets and the same accessories: automation, PC for supervision and rotating mechanism/conveyor. As the configuration and size are different, B10 only moves longitudinally, in range of 0-10m. with speed up to 0.2m /s, while B20 also moves transversally with identical speed up to 10m. in range. Also nozzle movement is therefore different - for B10 and B20 - telescopic movement 150-300 cm, swinging motion 180°, nozzle rotating 360° and nozzle rocking 270° and additionally B20 - rotation of the boom 340° and stretching of the boom - 180°. The technical data for both is as follows: hydraulic unit consists of: electric motor (11 kW), pump delivery 60 lpm, operational pressure 100 bar, tank capacity 200 l, weight - 800 KGs, switchboard is on the unit and connection to power - 20 kW. Height - 150 cm. Control switch cabinet - enclosure class IP54, weight - 450 KGs, connected power - 3 kW, height - 210 cm. Diameter of the nozzle - 16-19 mm, demand at the nozzle 16-24 m^3 /min., pressure 7-8 bar. Both systems use similar blasting materials (ferro-metallic and ceramic grains, organic materials, etc. and can additionally be used in high pressure washing and spray painting.

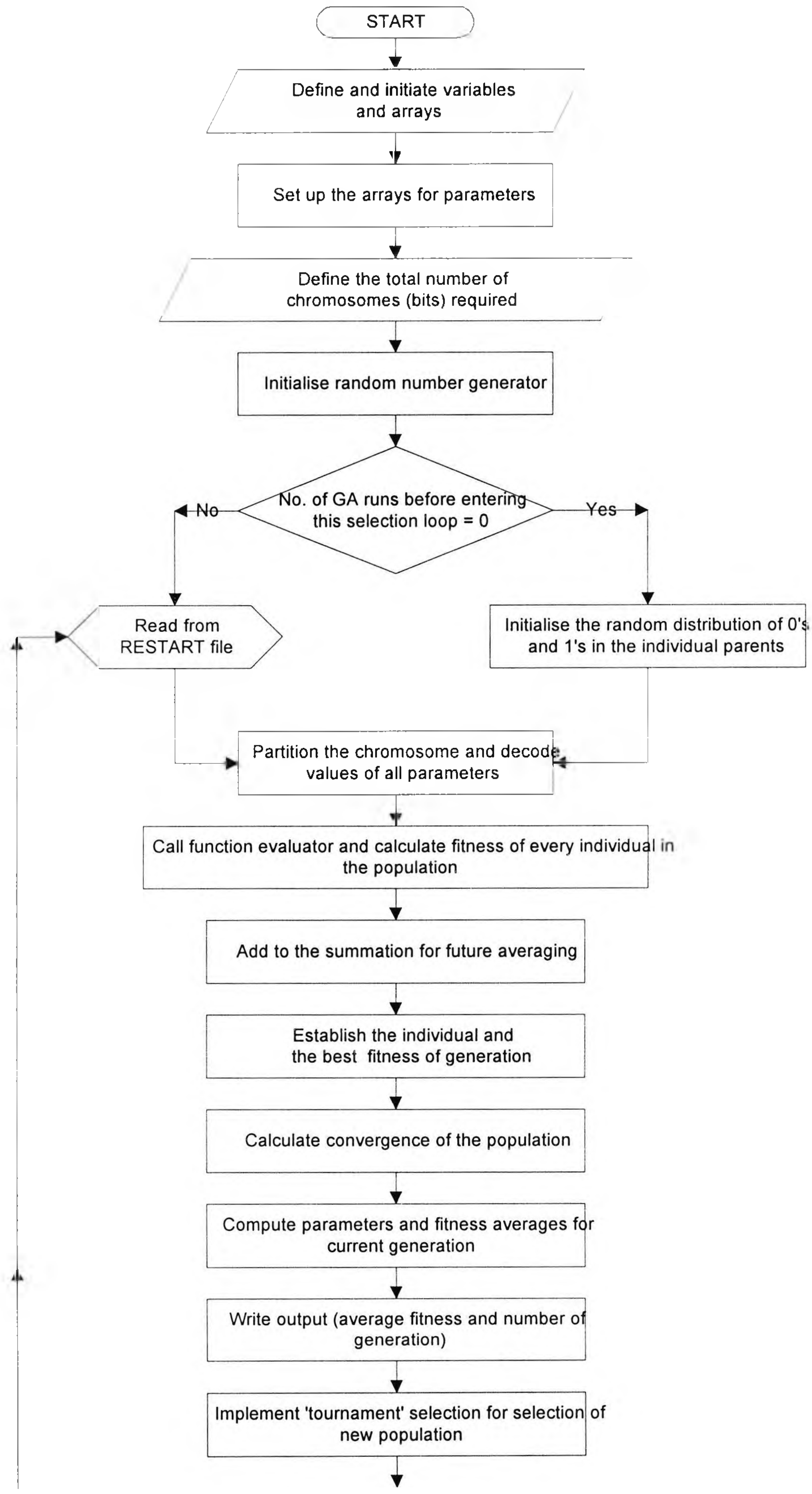
Blastman B20C by Rautaruukki [Rautaruukki, 1998] - more powerful than B20 and containing control cabin with air conditioning on a mobile bridge, but essentially similar. Used in shipyards. Robot movement range - 0-40m longitudinally and 0-20 m laterally. The technical data is as follows: hydraulic unit consists of: electric motor (15 kW), pump delivery 63 lpm, tank capacity 250l, weight - 800 KGs, switchboard is on the unit and connection to power - 25 kW. Height - 180 cm. Other characteristics similar to B20.

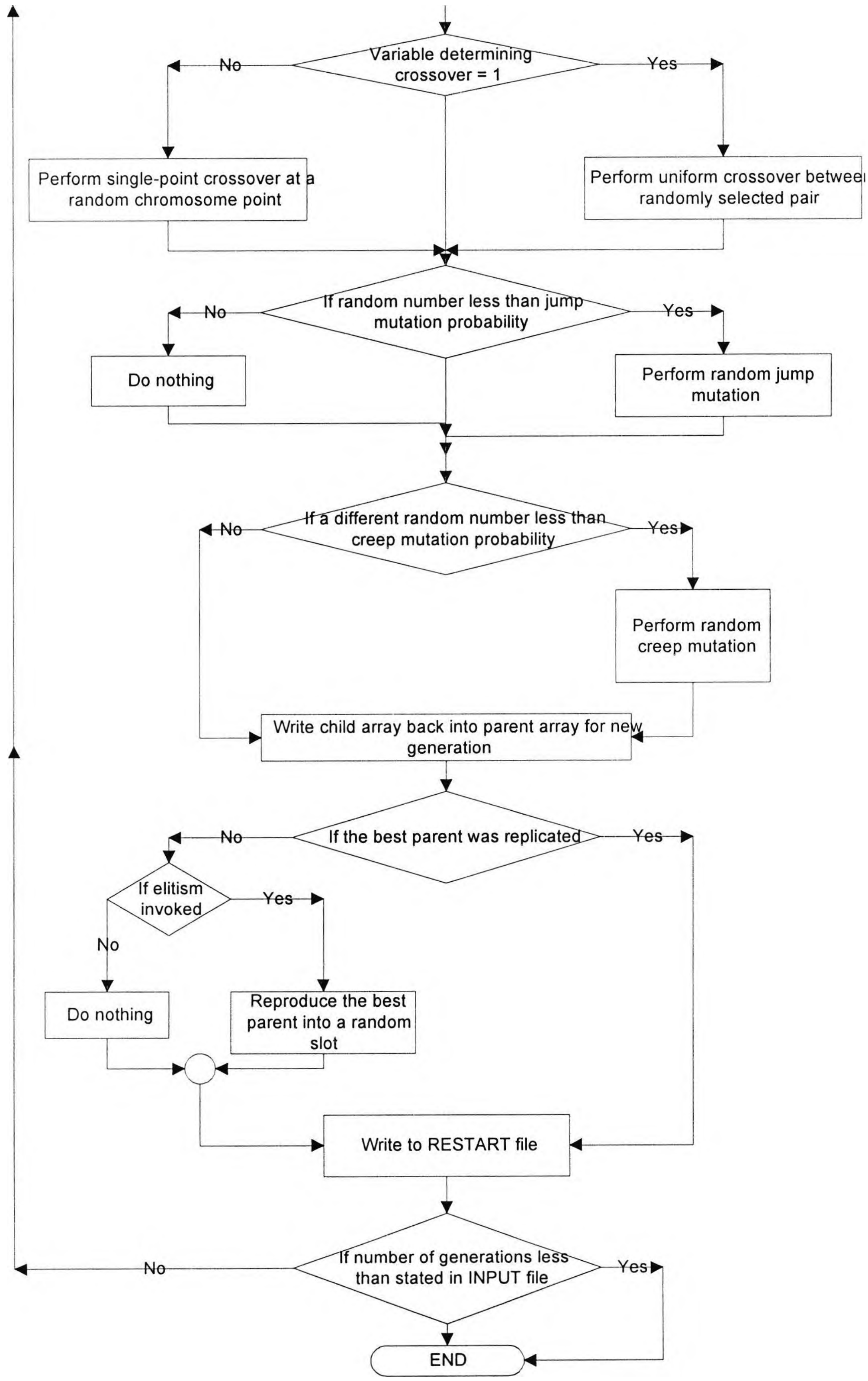
Blastman BE20 by Rautaruukki [Rautaruukki, 1998] - is a step up in precision of delivery, the system is more powerful than B20C, with horizontal speed of the robot up to 0.5m /s. The nozzle movement is more restricted - telescopic boom can move between 150-300 cm, swinging motion 120°, lifting motion 70°, nozzle rotation 360° and nozzle rocking 270°. The technical data is as follows: hydraulic unit consists of: electric motor (18.5 kW), pump delivery 110 lpm, weight - 900 KGs. Other characteristics similar to B20.

Blastman BR20 (2 units) by Rautaruukki [Rautaruukki, 1998] - an improved version of the BE20, with all characteristics identical, except: electric motor in hydraulic unit - 2x30 kW, pump delivery 2x160 lpm, operational pressure 120 bar, tank capacity 800l, weight - 2000 KGs, switchboard is on the unit and connection to power - 70 kW, double weight of control switch cabinet - 700 KGs. Double cleaning capacity. Heavy mass production wagon cleaning.

Robotic Bridge Paint Removal System by (RBPR) by Construction Automation and Robotics Laboratory (CARL) at the North Carolina State University, Raleigh,USA, utilises a bridge maintenance crane as the handling device for an articulated platform, two sliding tables, a robot arm, a dust control mechanism, a vision system, sensors and a sand blasting and vacuuming system [Moon and Bernold, 1995]. The third section of the crane booms is replaced with a new one for retrofitting. The actuated platform with the two sliding tables, are built for positioning the robotic sand blast and dust control mechanism. The vision system uses a frame grabber, a monitor and a camera. Ultrasonic transducers are used as distance sensors. To date, the system has undergone preliminary field testing with positive results.

APPENDIX D. ALGORITHM


START
Define and initiate variables and arrays
Set up the arrays for parameters
Define the total number of chromosomes (bits) required
Initialise random number generator
No. of GA runs before entering this selection loop = 0
No
Yes
Read from RESTART file
Initialise the random distribution of 0's and 1's in the individual parents
Partition the chromosome and decode values of all parameters
Call function evaluator and calculate fitness of every individual in the population
Add to the summation for future averaging
Establish the individual and the best fitness of generation
Calculate convergence of the population
Compute parameters and fitness averages for current generation
Write output (average fitness and number of generation)
Implement 'tournament' selection for selection of new population


Variable determining crossover = 1
No
Yes
Perform single-point crossover at a random chromosome point
Perform uniform crossover betwee randomly selected pair
If random number less than jump mutation probability
No
Yes
Do nothing
Perform random jump mutation
If a different random number less than creep mutation probability
No
Yes
Perform random creep mutation
Write child array back into parent array for new generation
If the best parent was replicated
No
Yes
If elitism invoked
Yes
No
Do nothing
Reproduce the best parent into a random slot
Write to RESTART file
If number of generations less than stated in INPUT file
No
Yes
END

APPENDIX E. FORTRAN AND GRASP SOFTWARE

E.1 FORTRAN Subroutines

code = Codes floating point value to binary string.

crosovr = Performs crossover (single-point or uniform).

decode = Decodes binary string to floating point value.

evalout = Evaluates the fitness of each individual and outputs generational information to the 'ga.out' file.

func = The function which is being evaluated.

gamicro = Implements the micro-GA technique.

input = Inputs information from the 'ga.inp' file.

initial = Program initialization and inputs information from the 'ga.restart' file.

mutate = Performs mutation (jump and/or creep).

newgen = Writes child array back into parent array for new generation; also checks to see if best individual was replicated (elitism).

niche = Performs niching (sharing) on population.

possibl = Checks to see if decoded binary string falls within specified range of parmin and parmax.

ran3 = The random number generator.

restart = Writes the 'ga.restart' file.

select = A subroutine of 'selectn'.

selectn = Performs selection; tournament selection is the only option in this version of the code.

shuffle = Shuffles the population randomly for selection.

E.2 File "Params.f"

```
parameter (indmax=1500,nchrmax=60,nparmax=60)
indmax = maximum # of individuals, i.e. max population size
nchrmax = maximum # of chromosomes (binary bits) per individual
nparmax = maximum # of parameters which the chromosomes make up
```

E.3 FORTRAN File "GA.f"

E.3.1 Main Body of the Program for Stage One

```
  implicit double precision (a-h,o-z)
 save
 include 'parmrb1.f'
     dimension parent(indmax,nparmax),child(indmax,nparmax)
      dimension fitness(indmax),nposibl(nparmax),nichflg(nparmax)
      dimension iparent(indmax,nchrmax),ichild(indmax,nchrmax)
      dimension g0(nparmax),g1(nparmax),ig2(nparmax)
      dimension paramsm(nparmax),paramav(nparmax),ibest(nchrmax)
      dimension parmax(nparmax),parmin(nparmax),pardel(nparmax)
      dimension path(mxcoor,mxpt),collis(mxcoor,mxcpt)
      dimension tool(indmax,mxcoor,mxpt),elbow(indmax,mxcoor,mxpt)
      dimension ll(mxpt),devv(indmax)
      dimension par(mxpt),ptx(mxpt),pty(mxpt),ptz(mxpt),den(mxpt)
      dimension distance(mxpt),dist(mxpt)
      dimension disttl(indmax)

      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga3   / parent,iparent
      common / ga4   / fitness
      common / ga5   / g0,g1,ig2
      common / ga6   / parmax,parmin,pardel,nposibl
      common / ga7   / child,ichild
      common / ga8   / collis
      common / ga9   / tool
      common / ga10  / coor,pt
      common / ga11  / path
      common / ga12  / cpt
      common / ga13  / ll,devv
```

```
          common / ga14  / elbow
          common / ga15  / par,ptx,pty,ptz,den
          common / ga16  / distance, dist
          common / ga17  / paramsm
          common / ga18  / paramav
          common / ga19  /distl
          common / ga20  / nichflg
          common /inputga/ pcross,pmutate,pcreep,maxgen,idum,irestrt,
         +             itourny,ielite,icreep,iunifrm,iniche,
         +             iskip,iend,nchild,microga,kountmx
        Call the input subroutine.
          call input
        Perform necessary initialization and read the ga.restart file.
          call initial(istart,npossum,ig2sum)
      $$$$$ Main generational processing loop. $$$$$
      kount=0
   do 20 i=istart,maxgen+istart-1
   write (6,1111) i
   write (24,1111) i
Evaluate the population, assign fitness, establish the best individual, and write output information.
           call evalout(iskip,iend,ibest)
           if(npopsiz.eq.1 .or. iskip.ne.0) then
             close(24)
             stop
           endif
Implement niching
            if (iniche.ne.0) call niche
Enter selection, crossover and mutation loop.
           ncross=0
           ipick=npopsiz
           do 45 j=1,npopsiz,nchild
Perform selection.
           call selectn(ipick,j,mate1,mate2)
Now perform crossover between the randomly selected pair.
           call crosovr(ncross,j,mate1,mate2)
      45     continue
           write(6,1225) ncross
           write(24,1225) ncross
Now perform random mutation.If running micro-GA skip mutation.
           if (microga.eq.0) call mutate
Write child array back into parent array for new generation.  Check to see if the best parent was replicated.
           call newgen(ielite,npossum,ig2sum,ibest)
Implement micro-GA if enabled
           if (microga.ne.0) call gamicro(i,npossum,ig2sum,ibest)
Write to restart file.
           call restart(i,istart,kount)
      20   continue
      $$$$$ End of main generational processing loop. $$$$$
      CLOSE (24)
1050 format(1x,' #      Binary Code',8x,'Param1 Param2 Param3',
     +        ' Param4 Param5 Param6 Param7 Param8 Param9 Param10',
     +        ' Param11 Param12 Param13 Param14 Param15 Param16',
     +        ' Param17 Param18 Param19 Param20 Param21',
     +        ' Param22 Param23 Param24 Param25 Param26 Param27',
     +        ' Param28 Param29 Param30 Param31 Param32 Param33',
     +        ' Param34 Param35 Param36 Param37 Fitness')
      1111 format(//'################  Generation',i5,'  ################')
      1225 format(/'  Number of Crossovers      =',i9)
         stop
         end
```

E.3.2 Subroutines

E.3.2.1.Subroutine input

This subroutine inputs information from the ga.inp file.

```
         implicit double precision (a-h,o-z)
         save
         include 'parmrb1.f'
         dimension nposibl(nparmax),nichflg(nparmax)
         dimension parmax(nparmax),parmin(nparmax),pardel(nparmax)
         dimension path(mxcoor,mxpt),collis(mxcoor,mxcpt)
         common / ga1   / npopsiz,nowrite
         common / ga2   / nparam,nchrome
         common / ga6   / parmax,parmin,pardel,nposibl
         common / ga8   / collis
```

```
      common / ga10  / coor,pt
      common / ga11  / path
      common / ga12  / cpt
      common / ga20  / nichflg
      common /inputga/ pcross,pmutate,pcreep,maxgen,idum,irestrt,
     +                itourny,ielite,icreep,iunifrm,iniche,
     +                iskip,iend,nchild,microga,kountmx
      namelist / garbap / irestrt,npopsiz,pmutate,maxgen,idum,pcross,
     +                 itourny,ielite,icreep,pcreep,iunifrm,iniche,
     +                 iskip,iend,nchild,nparam,parmin,parmax,nposibl,
     +                 nowrite,path,collis,coor,pt,cpt,microga,
     +                 nichflg,kountmx
      kountmx=50
      irestrt=0
      itourny=0
      ielite=0
      iunifrm=0
      iskip=0
      iend=0
      nchild=1
      do 2 i=1,nparam
      nichflg(i)=1
    2  continue
      microga=0
      OPEN (UNIT=24, FILE='garbap.out', STATUS='UNKNOWN')
      rewind 24
      OPEN (UNIT=26, FILE='garb1p.out', STATUS='UNKNOWN')
      rewind 26
      OPEN (UNIT=23, FILE='garbap1.inp', STATUS='OLD')
      READ (23, NML = garbap)
      CLOSE (23)
      itourny=1
      if (itourny.eq.0) nchild=2
c  Check for array sizing errors.
      if (npopsiz.gt.indmax) then
        write(6,1600) npopsiz
        write(24,1600) npopsiz
        close(24)
        stop
      endif
      if (nparam.gt.nparmax) then
        write(6,1700) nparam
        write(24,1700) nparam
        close(24)
        stop
      endif
c  If using the microga option, reset some input variables
      if (microga.ne.0) then
        pmutate=0.0
        pcreep=0.0
        itourny=1
        ielite=1
        iniche=0
        nchild=1
        if (iunifrm.eq.0) then
          pcross=1.0
        else
          pcross=0.5
        endif
      endif
      if (coor.gt.mxcoor) then
        write(6,1900) coor
        write(24,1900) coor
        close(24)
        stop
      endif
      if (pt.gt.mxpt) then
        write(6,1950) pt
        write(24,1950) pt
        close(24)
        stop
      endif
      if (cpt.gt.mxcpt) then
        write(6,1999) cpt
```

```
          write(24,1999) cpt
          close(24)
          stop
        endif
 1600 format(1x,'ERROR: npopsiz > indmax.  Set indmax = ',i6)
 1700 format(1x,'ERROR: nparam > nparmax.  Set nparmax =',i6)
 1900 format(1x,'ERROR: coor>mxcoor.     Set mxcoor  =',i6)
 1950 format(1x,'ERROR: pt>mxpt.         Set mxpt    =',i6)
 1999 format(1x,'ERROR: cpt>mxcpt.       Set mxcpt   =',i6)
      return
      end
```

E.3.2.2Subroutine initial(istart,npossum,ig2sum)

This subroutine sets up the program by generating the g0, g1 and ig2 arrays, and counting the number of chromosomes required for the specified input. The subroutine also initializes the random number generator, parent and iparent arrays (reads the ga.restart file).

```
      implicit double precision (a-h,o-z)
      save
      include 'parmrb1.f'
      dimension parent(indmax,nparmax),iparent(indmax,nchrmax)
      dimension nposibl(nparmax)
      dimension g0(nparmax),g1(nparmax),ig2(nparmax)
      dimension parmax(nparmax),parmin(nparmax),pardel(nparmax)
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga3   / parent,iparent
      common / ga5   / g0,g1,ig2
      common / ga6   / parmax,parmin,pardel,nposibl
      common /inputga/ pcross,pmutate,pcreep,maxgen,idum,irestrt,
     +            itourny,ielite,icreep,iunifrm,iniche,
     +            iskip,iend,nchild,microga,kountmx
      do 3 i=1,nparam
        g0(i)=parmin(i)
        pardel(i)=parmax(i)-parmin(i)
        g1(i)=pardel(i)/dble(nposibl(i)-1)
 3    continue
      do 6 i=1,nparam
        do 7 j=1,40
          n2j=2**j
          if (n2j.ge.nposibl(i)) then
            ig2(i)=j
            goto 8
          endif
          if (j.ge.40) then
            write(6,2000)
            write(24,2000)
            close(24)
            stop
          endif
 7      continue
 8      continue
 6    continue
```

Count the total number of chromosomes (bits) required

```
      nchrome=0
      npossum=0
      ig2sum=0
      do 9 i=1,nparam
        nchrome=nchrome+ig2(i)
        npossum=npossum+nposibl(i)
        ig2sum=ig2sum+(2**ig2(i))
 9    continue
      if (nchrome.gt.nchrmax) then
        write(6,1800) nchrome
        write(24,1800) nchrome
        close(24)
        stop
      endif
      if (npossum.lt.ig2sum.and.microga.ne.0) then
        write(6,2100)
        write(24,2100)
      endif
```

Initialize random number generator

```
      call ran3(idum,rand)
      IF(irestrt.eq.0) THEN
```

Initialize the random distribution of parameters in the individual parents when irestrt=0.

```
          istart=1
          do 10 i=1,npopsiz
            do 15 j=1,nchrome
              call ran3(1,rand)
              iparent(i,j)=1
              IF(rand.lt.0.5) iparent (i,j)=0
 15         continue
 10       continue
          IF(npossum.lt.ig2sum) call possibl(parent,iparent)
          ELSE
If irestrt.ne.0, read from restart file.
          OPEN (UNIT=25, FILE='garbap.restart', STATUS='OLD')
          rewind 25
          read(25,*) istart,npopsiz
          do 1 j=1,npopsiz
            read(25,*) k,(iparent(j,l),l=1,nchrome)
 1        continue
          CLOSE (25)
        ENDIF
        IF(irestrt.ne.0) call ran3(idum-istart,rand)
 1800 format(1x,'ERROR: nchrome > nchrmax.  Set nchrmax = ',i6)
 2000 format(1x,'ERROR: You have a parameter with a number of '/
     +       1x,'   possibilities > 2**40!  If you really desire this,'/
     +       1x,'   change the DO loop 7 statement and recompile.')
 2100 format(1x,'WARNING: for some cases, a considerable performance'/
     +       1x,'   reduction has been observed when running a non-'/
     +       1x,'   optimal number of bits with the micro-GA.'/
     +       1x,'   If possible, use values for nposibl of 2**n,'/
     +       1x,'   e.g. 2, 4, 8, 16, 32, 64, etc.  See ReadMe file.')
      return
    end
```

E.3.2.3 Subroutine niche

Implement "niching" through Goldberg's multidimensional phenotypic sharing scheme with a triangular sharing function. To find the multidimensional distance from the best individual, normalize all parameter differences.

```
        implicit double precision (a-h,o-z)
        save
        include 'parmrb1.f'
        dimension parent(indmax,nparmax),iparent(indmax,nchrmax)
        dimension fitness(indmax),nposibl(nparmax),nichflg(nparmax)
        dimension parmax(nparmax),parmin(nparmax),pardel(nparmax)
        common / ga1   / npopsiz,nowrite
        common / ga2   / nparam,nchrome
        common / ga3   / parent,iparent
        common / ga4   / fitness
        common / ga6   / parmax,nposibl,parmin,pardel
        common / ga20  / nichflg
        alpha=1.0
        sigshar=0.1
        nniche=0.0
        do 33 jj=1,nparam
          nniche=nniche+nichflg(jj)
 33     continue
        if (nniche.eq.0.0) then
          write(6,1900)
          write(24,1900)
          close(24)
          stop
        endif
        do 34 ii=1,npopsiz
          sumshar=0.0
          do 35 j=1,npopsiz
            del2=0.0
            do 36 k=1,nparam
            if (nichflg(k).ne.0) then
            del2=del2+((parent(j,k)-parent(ii,k))/pardel(k))**2.0
              endif
 36         continue
            del=(dsqrt(del2))/dble(nniche)
            if (del.lt.sigshar) then
            share=1.0-((del/sigshar)**alpha)
            else
            share=0.0
            endif
```

```
          sumshar=sumshar+share/dble(npopsiz)
   35     continue
        if (sumshar.ne.0.0) fitness(ii)=fitness(ii)/sumshar
   34   continue
 1900 format(1x,'ERROR: iniche=1 and all values in nichflg array = 0'/
     +       1x,'       Do you want to niche or not?')
      return
      end
```

E.3.2.4 Subroutine selectn(ipick,j,mate1,mate2)

Subroutine for selection operator. Presently, tournament selection is the only option available.

```
      implicit double precision (a-h,o-z)
      save
      include 'parmrb1.f'
      dimension parent(indmax,nparmax),child(indmax,nparmax)
      dimension fitness(indmax)
      dimension iparent(indmax,nchrmax),ichild(indmax,nchrmax)
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga3   / parent,iparent
      common / ga4   / fitness
      common / ga7   / child,ichild
      common /inputga/ pcross,pmutate,pcreep,maxgen,idum,irestrt,
     +              itourny,ielite,icreep,iunifrm,iniche,
     +              iskip,iend,nchild,microga,kountmx
```

If tournament selection is chosen (i.e. itourny=1), then implement "tournament" selection for selection of new population.

```
      if(itourny.eq.1) then
        call select(mate1,ipick)
        call select(mate2,ipick)
        write(3,*) mate1,mate2,fitness(mate1),fitness(mate2)
        do 46 n=1,nchrome
          ichild(j,n)=iparent(mate1,n)
          if(nchild.eq.2) ichild(j+1,n)=iparent(mate2,n)
   46   continue
      endif
      return
      end
```

E.3.2.5 Subroutine crosovr(ncross,j,mate1,mate2)

Subroutine for crossover between the randomly selected pair.

```
      implicit double precision (a-h,o-z)
      save
      include 'parmrb1.f'
      dimension parent(indmax,nparmax),child(indmax,nparmax)
      dimension iparent(indmax,nchrmax),ichild(indmax,nchrmax)
      common / ga2   / nparam,nchrome
      common / ga3   / parent,iparent
      common / ga7   / child,ichild
      common /inputga/ pcross,pmutate,pcreep,maxgen,idum,irestrt,
     +              itourny,ielite,icreep,iunifrm,iniche,
     +              iskip,iend,nchild,microga,kountmx
      if (iunifrm.eq.0) then
```

Single-point crossover at a random chromosome point.

```
        call ran3(1,rand)
        if(rand.gt.pcross) goto 69
        ncross=ncross+1
        call ran3(1,rand)
        icross=2+dint(dble(nchrome-1)*rand)
        do 50 n=icross,nchrome
          ichild(j,n)=iparent(mate2,n)
          if(nchild.eq.2) ichild(j+1,n)=iparent(mate1,n)
   50   continue
      else
```

Perform uniform crossover between the randomly selected pair.

```
        do 60 n=1,nchrome
          call ran3(1,rand)
          if(rand.le.pcross) then
            ncross=ncross+1
            ichild(j,n)=iparent(mate2,n)
            if(nchild.eq.2) ichild(j+1,n)=iparent(mate1,n)
          endif
   60   continue
      endif
   69 continue
```

```
      return
      end
```

E.3.2.6 Subroutine mutate

```
      implicit double precision (a-h,o-z)
      save
      include 'parmrb1.f'
      dimension nposibl(nparmax)
      dimension child(indmax,nparmax),ichild(indmax,nchrmax)
      dimension g0(nparmax),g1(nparmax),ig2(nparmax)
      dimension parmax(nparmax),parmin(nparmax),pardel(nparmax)
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga5   / g0,g1,ig2
      common / ga6   / parmax,parmin,pardel,nposibl
      common / ga7   / child,ichild
      common /inputga/ pcross,pmutate,pcreep,maxgen,idum,irestrt,
     +                itourny,ielite,icreep,iunifrm,iniche,
     +                iskip,iend,nchild,microga,kountmx
c  This subroutine performs mutations on the children generation.
c  Perform random jump mutation if a random number is less than pmutate.
c  Perform random creep mutation if a different random number is less than pcreep.
      nmutate=0
      ncreep=0
      do 70 j=1,npopsiz
        do 75 k=1,nchrome
c  Jump mutation
          call ran3(1,rand)
          if (rand.le.pmutate) then
            nmutate=nmutate+1
            if(ichild(j,k).eq.0) then
              ichild(j,k)=1
            else
              ichild(j,k)=0
            endif
            if (nowrite.eq.0) write(6,1300) j,k
            if (nowrite.eq.0) write(24,1300) j,k
          endif
 75     continue
c  Creep mutation (one discrete position away).
        if (icreep.ne.0) then
          do 76 k=1,nparam
            call ran3(1,rand)
            if(rand.le.pcreep) then
              call decode(j,child,ichild)
              ncreep=ncreep+1
              creep=1.0
              call ran3(1,rand)
              if (rand.lt.0.5) creep=-1.0
              child(j,k)=child(j,k)+g1(k)*creep
              if (child(j,k).gt.parmax(k)) then
                child(j,k)=parmax(k)-1.0*g1(k)
              elseif (child(j,k).lt.parmin(k)) then
                child(j,k)=parmin(k)+1.0*g1(k)
              endif
              call code(j,k,child,ichild)
              if (nowrite.eq.0) write(6,1350) j,k
              if (nowrite.eq.0) write(24,1350) j,k
            endif
 76       continue
        endif
 70   continue
      write(6,1250) nmutate,ncreep
      write(24,1250) nmutate,ncreep
 1250 format(/'  Number of Jump Mutations  =',i5/
     +        '  Number of Creep Mutations =',i5)
 1300 format('*** Jump mutation performed on individual  ',i4,
     +       ', chromosome ',i3,' ***')
 1350 format('*** Creep mutation performed on individual ',i4,
     +       ', parameter  ',i3,' ***')
      return
      end
```

E.3.2.7 Subroutine newgen(ielite,npossum,ig2sum,ibest)

Write child array back into parent array for new generation. Check to see if the best parent was replicated; if not, and

```
if ielite=1, then reproduce the best parent into a random slot.
      implicit double precision (a-h,o-z)
      save
      include 'parmrb1.f'
      dimension parent(indmax,nparmax),child(indmax,nparmax)
      dimension iparent(indmax,nchrmax),ichild(indmax,nchrmax)
      dimension ibest(nchrmax)
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga3   / parent,iparent
      common / ga7   / child,ichild
      kelite=0
      do 94 j=1,npopsiz
        jelite=0
        do 95 n=1,nchrome
          iparent(j,n)=ichild(j,n)
          if (iparent(j,n).eq.ibest(n)) jelite=jelite+1
          if (jelite.eq.nchrome) kelite=1
 95     continue
 94   continue
      if (ielite.ne.0 .and. kelite.eq.0) then
        call ran3(1,rand)
        irand=1+dint(dble(npopsiz)*rand)
        do 96 n=1,nchrome
          iparent(irand,n)=ibest(n)
 96     continue
        write(24,1260) irand
      endif
 1260 format('  Elitist Reproduction on Individual ',i4)
      return
      end
```

E.3.2.8 Subroutine gamicro(i,npossum,ig2sum,ibest)

```
Micro-GA implementation subroutine
      implicit double precision (a-h,o-z)
      save
      include 'parmrb1.f'
      dimension parent(indmax,nparmax),iparent(indmax,nchrmax)
      dimension ibest(nchrmax)
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga3   / parent,iparent
First, check for convergence of micro population.
If converged, start a new generation with best individual and fill the remainder of the population with new randomly generated parents.
Count number of different bits from best member in micro-population
      icount=0
      do 81 j=1,npopsiz
        do 82 n=1,nchrome
          if(iparent(j,n).ne.ibest(n)) icount=icount+1
 82     continue
 81   continue
If icount less than 5% of number of bits, then consider population to be converged.  Restart with best individual and random others.
      diffrac=dble(icount)/dble((npopsiz-1)*nchrome)
      if (diffrac.lt.0.05) then
      do 87 n=1,nchrome
        iparent(1,n)=ibest(n)
 87   continue
      do 88 j=2,npopsiz
        do 89 n=1,nchrome
          call ran3(1,rand)
          iparent(j,n)=1
          if(rand.lt.0.5) iparent(j,n)=0
 89     continue
 88   continue
      if (npossum.lt.ig2sum) call possibl(parent,iparent)
      write(6,1375) i
      write(24,1375) i
      endif
 1375 format(//'%%%%%%%  Restart micro-population at generation',
     +       i5,' %%%%%%%')
      return
      end
```

E.3.2.9 Subroutine select(mate,ipick)

This routine selects the better of two possible parents for mating.

```
      implicit double precision (a-h,o-z)
          save
          include 'parmrb1.f'
          common / ga1   / npopsiz,nowrite
          common / ga2   / nparam,nchrome
          common / ga3   / parent,iparent
          common / ga4   / fitness
          dimension parent(indmax,nparmax),iparent(indmax,nchrmax)
          dimension fitness(indmax)
          if(ipick+1.gt.npopsiz) call shuffle(ipick)
          ifirst=ipick
          isecond=ipick+1
          ipick=ipick+2
          if(fitness(ifirst).gt.fitness(isecond)) then
            mate=ifirst
          else
            mate=isecond
          endif
          write(3,*)'select',ifirst,isecond,fitness(ifirst),fitness(isecond)
          return
          end
```

E.3.2.10 Subroutine shuffle(ipick)

This routine shuffles the parent array and its corresponding fitness

```
          implicit double precision (a-h,o-z)
          save
          include 'parmrb1.f'
          common / ga1   / npopsiz,nowrite
          common / ga2   / nparam,nchrome
          common / ga3   / parent,iparent
          common / ga4   / fitness
          dimension parent(indmax,nparmax),iparent(indmax,nchrmax)
          dimension fitness(indmax)
          ipick=1
          do 10 j=1,npopsiz-1
            call ran3(1,rand)
            iother=j+1+dint(dble(npopsiz-j)*rand)
            do 20 n=1,nchrome
              itemp=iparent(iother,n)
              iparent(iother,n)=iparent(j,n)
              iparent(j,n)=itemp
      20     continue
            temp=fitness(iother)
            fitness(iother)=fitness(j)
            fitness(j)=temp
      10   continue
          return
          end
```

E.3.2.11 Subroutine decode(i,array,iarray)

This routine decodes a binary string to a real number.

```
          implicit double precision (a-h,o-z)
          save
          include 'parmrb1.f'
          common / ga2   / nparam,nchrome
          common / ga5   / g0,g1,ig2
          dimension array(indmax,nparmax),iarray(indmax,nchrmax)
          dimension g0(nparmax),g1(nparmax),ig2(nparmax)
          l=1
          do 10 k=1,nparam
            iparam=0
            m=l
            do 20 j=m,m+ig2(k)-1
              l=l+1
              iparam=iparam+iarray(i,j)*(2**(m+ig2(k)-1-j))
      20     continue
            array(i,k)=g0(k)+g1(k)*dble(iparam)
      10   continue
          return
          end
```

E.3.2.12 Subroutine code(j,k,array,iarray)

This routine codes a parameter into a binary string.

```
      implicit double precision (a-h,o-z)
      save
      include 'parmrb1.f'
      common / ga2   / nparam,nchrome
      common / ga5   / g0,g1,ig2
      dimension array(indmax,nparmax),iarray(indmax,nchrmax)
      dimension g0(nparmax),g1(nparmax),ig2(nparmax)
```
First, establish the beginning location of the parameter string of interest.
```
      istart=1
      do 10 i=1,k-1
        istart=istart+ig2(i)
 10   continue
```
Find the equivalent coded parameter value, and back out the binary string by factors of two.
```
      m=ig2(k)-1
      if (g1(k).eq.0.0) return
      iparam=nint((array(j,k)-g0(k))/g1(k))
      do 20 i=istart,istart+ig2(k)-1
        iarray(j,i)=0
        if ((iparam+1).gt.(2**m)) then
          iarray(j,i)=1
          iparam=iparam-2**m
        endif
        m=m-1
 20   continue
     write(3,*)array(j,k),iparam,(iarray(j,i),i=istart,istart+ig2(k)-1)
      return
      end
```
E.3.2.13 Subroutine possibl(array,iarray)

This subroutine determines whether or not all parameters are within the specified range of possibility. If not, the parameter is randomly reassigned within the range. This subroutine is only necessary when the number of possibilities per parameter is not optimized to be 2**n, i.e. if npossum < ig2sum.
```
      implicit double precision (a-h,o-z)
      save
      include 'params1.f'
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga5   / g0,g1,ig2
      common / ga6   / parmax,parmin,pardel,nposibl
      dimension array(indmax,nparmax),iarray(indmax,nchrmax)
      dimension g0(nparmax),g1(nparmax),ig2(nparmax),nposibl(nparmax)
      dimension parmax(nparmax),parmin(nparmax),pardel(nparmax)
      do 10 i=1,npopsiz
        call decode(i,array,iarray)
        do 20 j=1,nparam
          n2ig2j=2**ig2(j)
        if(nposibl(j).ne.n2ig2j .and. array(i,j).gt.parmax(j)) then
            call ran3(1,rand)
            irand=dint(dble(nposibl(j))*rand)
            array(i,j)=g0(j)+dble(irand)*g1(j)
            call code(i,j,array,iarray)
            if (nowrite.eq.0) write(6,1000) i,j
            if (nowrite.eq.0) write(24,1000) i,j
          endif
 20     continue
 10   continue
 1000 format('*** Parameter adjustment to individual     ',i4,
     +       ', parameter  ',i3,' ***')
      return
      end
```
E.3.2.14 Subroutine restart(i,istart,kount)

This subroutine writes restart information to the ga.restart file.
```
      implicit double precision (a-h,o-z)
      save
      include 'parmrb1.f'
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga3   / parent,iparent
      dimension parent(indmax,nparmax),iparent(indmax,nchrmax)
      common /inputga/ pcross,pmutate,pcreep,maxgen,idum,irestrt,
     +              itourny,ielite,icreep,iunifrm,iniche,
     +              iskip,iend,nchild,microga,kountmx
      kount=kount+1
      if(i.eq.maxgen+istart-1 .or. kount.eq.kountmx) then
```

```
        OPEN (UNIT=25, FILE='garbap.restart', STATUS='OLD')
        rewind 25
        write(25,*) i+1,npopsiz
        do 80 j=1,npopsiz
          write(25,1500) j,(iparent(j,l),l=1,nchrome)
80      continue
        CLOSE (25)
        kount=0
      endif
1500 format(i6,3x,222i2)
      return
      end
```

E.3.2.15 Subroutine ran3(idum,rand)

Returns a uniform random deviate between 0.0 and 1.0. Set idum to any negative value to initialize or reinitialize the sequence.
This function is taken from W.H. Press', "Numerical Recipes" p. 199.

```
      implicit double precision (a-h,m,o-z)
      save
      implicit real*4(m)
      parameter (mbig=4000000.,mseed=1618033.,mz=0.,fac=1./mbig)
      parameter (mbig=1000000000,mseed=161803398,mz=0,fac=1./mbig)
```

According to Knuth, any large mbig, and any smaller (but still large) mseed can be substituted for the above values.

```
      dimension ma(55)
      data iff /0/
      if (idum.lt.0 .or. iff.eq.0) then
        iff=1
        mj=mseed-dble(iabs(idum))
        mj=dmod(mj,mbig)
        ma(55)=mj
        mk=1
        do 11 i=1,54
          ii=mod(21*i,55)
          ma(ii)=mk
          mk=mj-mk
          if(mk.lt.mz) mk=mk+mbig
          mj=ma(ii)
11      continue
        do 13 k=1,4
          do 12 i=1,55
            ma(i)=ma(i)-ma(1+mod(i+30,55))
            if(ma(i).lt.mz) ma(i)=ma(i)+mbig
12        continue
13      continue
        inext=0
        inextp=31
        idum=1
      endif
      inext=inext+1
      if(inext.eq.56) inext=1
      inextp=inextp+1
      if(inextp.eq.56) inextp=1
      mj=ma(inext)-ma(inextp)
      if(mj.lt.mz) mj=mj+mbig
      ma(inext)=mj
      rand=mj*fac
      return
      end
```

E.3.2.16 Subroutine func(j,funcval)

```
      implicit double precision (a-h,o-z)
      save
      include 'parmrb1.f'
      dimension disttl(indmax)
      common / ga19  / disttl
      common /inputga/pcross,pmutate,pcreep,maxgen,idum,irestrt,
     +          itourny,ielite,icreep,iunifrm,iskip,iend,
     +          nchild,iniche, microga,kountmx
      funcval=1/disttl(j)
      return
      end
```

E.3.2.17 Subroutine evalout(iskip,iend,ibest) for RRP with niching and micro-GA for stage one

This subroutine evaluates the population, assigns fitness, establishes the best individual, and outputs information.

```
      implicit double precision (a-h,o-z)
      save
      include 'parmrb1.f'
      dimension parent(indmax,nparmax),iparent(indmax,nchrmax)
      dimension fitness(indmax)
      dimension paramsm(nparmax),paramav(nparmax),ibest(nchrmax)
      dimension tool(indmax,mxcoor,mxpt),elbow(indmax,mxcoor,mxpt)
      dimension ll(mxpt),devv(indmax)
      dimension path(mxcoor,mxpt),collis(mxcoor,mxcpt)
      dimension par(mxpt),ptx(mxpt),pty(mxpt),ptz(mxpt),den(mxpt)
      dimension distance(mxpt),dist(mxpt)
      dimension disttl(indmax)
      common / ga1  / npopsiz,nowrite
      common / ga2  / nparam,nchrome
      common / ga3  / parent,iparent
      common / ga4  / fitness
      common / ga8  / collis
      common / ga9  / tool
      common / ga10 / coor,pt
      common / ga11 / path
      common / ga12 / cpt
      common / ga13 / ll,devv
      common / ga14 / elbow
      common / ga15 / par,ptx,pty,ptz,den
      common / ga16 / distance,dist
      common / ga17 / paramsm
      common / ga19 / disttl
      fitsum=0.0
      best=0.0
      do 29 n=1,nparam
        paramsm(n)=0.0
   29 continue
      jstart=1
      jend=npopsiz
      if(iskip.ne.0) jstart=iskip
      if(iend.ne.0) jend=iend
      do 30 j=jstart,jend
        call decode(j,parent,iparent)

        if(iskip.ne.0 .and. iend.ne.0 .and. iskip.eq.iend)
     +  write(6,1075) j,(iparent(j,k),k=1,nchrome),
     +              (parent(j,kk),kk=1,nparam),0.0
        distsum=0.0
        sumdist=0.0
Ensuring small joint angle changes between adjacent points
      do 159 m=1,pt-1
      ppenalt=0.0
      IF (ABS(parent(j,3*m+3)-parent(j,3*m)).gt.89.or.
     +ABS(parent(j,3*m+2)-parent(j,3*m-1)).gt.89.or.
     +parent(j,3*m-2).eq.0.or.parent(j,3*m).eq.0.0.or.
     +parent(j,3*m).eq.90.0.or.parent(j,3*m).eq.180.0.or.
     +parent(j,3*m).eq.270.0.or.parent(j,3*m).eq.360.0)
     +THEN
      ll(m)=1000000
      ppenalt=ppenalt+ll(m)
      ENDIF
  159 continue
     write(24,*) 'ppenalt', ppenalt

      do 160 m=1,pt
      penalt=0.0
Calculating the tool and elbow matrices from decoded parent array
            parm=parent(j,3*m-2)*2000
            psi1=parent(j,3*m-1)*2*3.14159265/360
            psi2=parent(j,3*m)*2*3.14159265/360

        tool(j,1,m)=cos(psi1)*sin(psi2)*parm
        tool(j,2,m)=sin(psi1)*sin(psi2)*parm
        tool(j,3,m)=cos(psi2)*parm

      elbow(j,1,m)=0.0
      elbow(j,2,m)=0.0
      elbow(j,3,m)=0.0
```

```
Checking collision for specific scaled down collision envelope
Assign penalty
             do 240 n=1,cpt/2-1
           IF (collis(1,n).eq.collis(1,n+1).and.
          +collis(1,n+1).eq.collis(1,n+cpt/2).and.
          +collis(1,n+cpt/2).eq.collis(1,n+cpt/2+1)) THEN

Collision plane x=const.
Calculate the intersection coordinates (ptx,pty,ptz)
           den(m)=tool(j,1,m)-elbow(j,1,m)
           IF(den(m).eq.0.0) THEN
           GOTO 111
           ELSE
           par(m)=(collis(1,n)-elbow(j,1,m))/den(m)
           ptx(m)=collis(1,n)
           pty(m)=tool(j,2,m)+(tool(j,2,m)-elbow(j,2,m))*par(m)
           ptz(m)=tool(j,3,m)+(tool(j,3,m)-elbow(j,3,m))*par(m)

Check if coordinates belong to either arm or collision plane
           IF (ptx(m).le.tool(j,1,m).and.ptx(m).ge.elbow(j,1,m).and.
          +pty(m).le.tool(j,2,m).and.pty(m).ge.elbow(j,2,m).and.
          +ptz(m).le.tool(j,3,m).and.ptz(m).ge.elbow(j,3,m).and.
          +ptx(m).le.collis(1,n+1).and.ptx(m).ge.collis(1,n).and.
          +ptx(m).le.collis(1,n+cpt/2).and.ptx(m).ge.collis(1,n+cpt/2+1).and.
          +pty(m).le.collis(2,n+1).and.pty(m).ge.collis(2,n).and.
          +pty(m).le.collis(2,n+cpt/2).and.pty(m).ge.collis(2,n+cpt/2+1).and.
          +ptz(m).le.collis(3,n+1).and.ptz(m).ge.collis(3,n).and.
          +ptz(m).le.collis(3,n+cpt/2).and.ptz(m).ge.
          +collis(3,n+cpt/2+1)) THEN
           GOTO 111
           ENDIF
           ENDIF

           ELSE IF (collis(2,n).eq.collis(2,n+1).and.
          +collis(2,n+1).eq.collis(2,n+cpt/2).and.
          +collis(2,n+cpt/2).eq.collis(2,n+cpt/2+1)) THEN

Collision plane y=const.
Calculate the intersection coordinates (ptx,pty,ptz)
           den(m)=tool(j,2,m)-elbow(j,2,m)
           IF(den(m).eq.0.0) THEN
           GOTO 111
           ELSE
           par(m)=(collis(2,n)-elbow(j,2,m))/den(m)
           ptx(m)=tool(j,1,m)+(tool(j,1,m)-elbow(j,1,m))*par(m)
           pty(m)=collis(2,n)
           ptz(m)=tool(j,3,m)+(tool(j,3,m)-elbow(j,3,m))*par(m)

Check if coordinates belong to either arm or collision plane
           IF (ptx(m).le.tool(j,1,m).and.ptx(m).ge.elbow(j,1,m).and.
          +pty(m).le.tool(j,2,m).and.pty(m).ge.elbow(j,2,m).and.
          +ptz(m).le.tool(j,3,m).and.ptz(m).ge.elbow(j,3,m).and.
          +ptx(m).le.collis(1,n+1).and.ptx(m).ge.collis(1,n).and.
          +ptx(m).le.collis(1,n+cpt/2).and.ptx(m).ge.collis(1,n+cpt/2+1).and.
          +pty(m).le.collis(2,n+1).and.pty(m).ge.collis(2,n).and.
          +pty(m).le.collis(2,n+cpt/2).and.pty(m).ge.collis(2,n+cpt/2+1).and.
          +ptz(m).le.collis(3,n+1).and.ptz(m).ge.collis(3,n).and.
          +ptz(m).le.collis(3,n+cpt/2).and.ptz(m).ge.
          +collis(3,n+cpt/2+1)) THEN
           GOTO 111
           ENDIF
           ENDIF

           ELSE IF (collis(3,n).eq.collis(3,n+1).and.
          +collis(3,n+1).eq.collis(3,n+cpt/2).and.
          +collis(3,n+cpt/2).eq.collis(3,n+cpt/2+1)) THEN
Collision plane z=const.

Calculate the intersection coordinates (ptx,pty,ptz)
           den(m)=tool(j,3,m)-elbow(j,3,m)

           IF (den(m).eq.0.0) THEN
           GOTO 111
           ELSE
```

```
          par(m)=(collis(3,n)-elbow(j,3,m))/den(m)
          ptx(m)=tool(j,1,m)+(tool(j,1,m)-elbow(j,1,m))*par(m)
          pty(m)=tool(j,2,m)+(tool(j,2,m)-elbow(j,2,m))*par(m)
          ptz(m)=collis(3,n)

Check if coordinates belong to either arm or collision plane
          IF (ptx(m).le.tool(j,1,m).and.ptx(m).ge.elbow(j,1,m).and.
         +pty(m).le.tool(j,2,m).and.pty(m).ge.elbow(j,2,m).and.
         +ptz(m).le.tool(j,3,m).and.ptz(m).ge.elbow(j,3,m).and.
         +ptx(m).le.collis(1,n+1).and.ptx(m).ge.collis(1,n).and.
         +ptx(m).le.collis(1,n+cpt/2).and.ptx(m).ge.collis(1,n+cpt/2+1).and.
         +pty(m).le.collis(2,n+1).and.pty(m).ge.collis(2,n).and.
         +pty(m).le.collis(2,n+cpt/2).and.pty(m).ge.collis(2,n+cpt/2+1).and.
         +ptz(m).le.collis(3,n+1).and.ptz(m).ge.collis(3,n).and.
         +ptz(m).le.collis(3,n+cpt/2).and.ptz(m).ge.
         +collis(3,n+cpt/2+1)) THEN
          GOTO 111
          ENDIF
          ENDIF
          ENDIF
      240 continue
          GOTO 112
          111  ll(m)=1000000
          penalt=penalt+ll(m)
      112  dist(m)=sqrt((tool(j,1,m)-path(1,m))**2+(tool(j,2,m)
         +-path(2,m))**2+(tool(j,3,m)-path(3,m))**2)
          sumdist=sumdist+dist(m)
          distmin=dist(m)
          distance(m)=distmin+penalt
          distsum=distsum+distance(m)
      160 continue
          ddistsum=distsum+ppenalt
            avg=sumdist/pt
            dev=0.0
     do 170 m=1,pt
            dev=dev+(dist(m)-avg)**2
      170     continue
            devv(j)=sqrt(dev/pt)
            disttl(j)=ddistsum

Call function evaluator, write out individual and fitness, and add to the summation for later averaging.
             call func(j,funcval)
            fitness(j)=funcval
            fitsum=fitsum+fitness(j)
               do 222 n=1,nparam
                paramsm(n)=paramsm(n)+parent(j,n)
      222      continue
Check to see if fitness of individual j is the best fitness.
             if (fitness(j).gt.best) then
                best=fitness(j)
                jbest=j
                do 24 k=1,nchrome
                  ibest(k)=iparent(j,k)
      24          continue
             endif
      30   continue
          write(24,1075) jbest,(parent(jbest,kk),kk=1,nparam),best
          write(24,1250) devv(jbest)

Compute parameter and fitness averages.
          fbar=fitsum/dble(npopsiz)
          do 23 n=1,nparam
            paramav(n)=paramsm(n)/dble(npopsiz)
      23   continue
Write output information
           if (npopsiz.eq.1) then
             write(24,1075) 1,(parent(1,k),k=1,nparam),fitness(1)
             write(24,*) ' Average Values:'
             write(24,1275) (parent(1,k),k=1,nparam),fbar
           else
             write(24,1275) (paramav(k),k=1,nparam),fbar
           endif
          write(6,1100) fbar
           write(24,1100) fbar
```

```
      write(6,1200) best
       write(24,1200) best
      write(26,1340) best
 1075 format(i3,1x,f7.2,3x,3(1x,f7.2),3x,3(1x,f7.2),
     +3x,3(1x,f7.2),3x,3(1x,f7.2),3x,3(1x,f7.2),3x,
     +3(1x,f7.2),3x,3(1x,f7.2),3x,3(1x,f7.2),
     +3x,3(1x,f7.2),3x,3(1x,f7.2),3x,3(1x,f7.2),3x,
     +3(1x,f7.2),1x,f24.18)
 1100 format(1x,'Average Fitness Value of Generation=',f24.18)
 1200 format(1x,'Maximum Fitness Value              =',f24.18)
 1250 format(1x,'Standard Deviation=',f20.16)
 1275 format(/' Average Values:',10x,37(2x,f7.2),1x,f24.18/)
 1340 format(f24.18)
      return
      end
```

E.3.2.18 Subroutine evalout(iskip,iend,ibest) for RRR with niching and micro-GA for stage one

This subroutine evaluates the population, assigns fitness, establishes the best individual, and outputs information.

```
      implicit double precision (a-h,o-z)
      save
      include 'parmrb1.f'
      dimension parent(indmax,nparmax),iparent(indmax,nchrmax)
      dimension fitness(indmax)
      dimension paramsm(nparmax),paramav(nparmax),ibest(nchrmax)
      dimension tool(indmax,mxcoor,mxpt),elbow(indmax,mxcoor,mxpt)
      dimension ll(mxpt),devv(indmax)
      dimension path(mxcoor,mxpt),collis(mxcoor,mxcpt)
      dimension par(mxpt),ptx(mxpt),pty(mxpt),ptz(mxpt),den(mxpt)
      dimension distance(mxpt),dist(mxpt)
      dimension disttl(indmax)
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga3   / parent,iparent
      common / ga4   / fitness
      common / ga8   / collis
      common / ga9   / tool
      common / ga10  / coor,pt
      common / ga11  / path
      common / ga12  / cpt
      common / ga13  / ll,devv
      common / ga14  / elbow
      common / ga15  / par,ptx,pty,ptz,den
      common / ga16  / distance,dist
      common / ga17  / paramsm
      common / ga19  / disttl
      fitsum=0.0
      best=0.0
      do 29 n=1,nparam
        paramsm(n)=0.0
 29   continue

      jstart=1
      jend=npopsiz
      if(iskip.ne.0) jstart=iskip
      if(iend.ne.0) jend=iend
      do 30 j=jstart,jend
        call decode(j,parent,iparent)

        if(iskip.ne.0 .and. iend.ne.0 .and. iskip.eq.iend)
     +  write(6,1075) j,(iparent(j,k),k=1,nchrome),
     +               (parent(j,kk),kk=1,nparam),0.0
        distsum=0.0
        sumdist=0.0

c Ensuring small joint angle changes between adjacent points
      do 159 m=1,pt-1
      ppenalt=0.0
      IF(ABS(parent(j,3*m+2)-parent(j,3*m-1)).gt.89.or.
     +ABS(parent(j,3*m+3)-parent(j,3*m)).gt.89.or.
     +ABS(parent(j,3*m+4)-parent(j,3*m+1)).gt.89) THEN
      ll(m)=1000000
      ppenalt=ppenalt+ll(m)
      ENDIF
```

```
159 continue

      do 160 m=1,pt
      penalt=0.0
From deg into rad + calculating the tool and elbow matrices from decoded parent array
          psi1=parent(j,3*m-1)*3.14159265/180
          psi2=parent(j,3*m)*3.14159265/180
          psi3=parent(j,3*m+1)*3.14159265/180

Avoid zero tool position
      IF(((2000*(1-parent(j,1)))*cos(psi2+psi3)).eq.
     +-(2000*parent(j,1)*cos(psi2))) GOTO 111
      tool(j,1,m)=cos(psi1)*((2000*(1-parent(j,1)))*cos(psi2+psi3)
     ++2000*parent(j,1)*cos(psi2))
      tool(j,2,m)=sin(psi1)*((2000*(1-parent(j,1)))*cos(psi2+psi3)
     ++2000*parent(j,1)*cos(psi2))
      tool(j,3,m)=-2000*parent(j,1)*sin(psi2)-(2000*(1-parent(j,1)))*
     +sin(psi2+psi3)

      elbow(j,1,m)=2000*parent(j,1)*cos(psi1)*cos(psi2)
      elbow(j,2,m)=2000*parent(j,1)*cos(psi2)*sin(psi1)
      elbow(j,3,m)=2000*parent(j,1)*sin(psi2)
Assign penalty
Checking collision for specific scaled down collision envelope
      do 240 n=1,cpt/2-1
      IF (collis(1,n).eq.collis(1,n+1).and.
     +collis(1,n+1).eq.collis(1,n+cpt/2).and.
     +collis(1,n+cpt/2).eq.collis(1,n+cpt/2+1)) THEN

Collision plane is x=const.
Calculate the intersection coordinates (ptx,pty,ptz)
      den(m)=tool(j,1,m)-elbow(j,1,m)
      IF(den(m).eq.0.0) THEN
      GOTO 111
      ELSE
      par(m)=(collis(1,n)-elbow(j,1,m))/den(m)
      ptx(m)=collis(1,n)
      pty(m)=tool(j,2,m)+(tool(j,2,m)-elbow(j,2,m))*par(m)
      ptz(m)=tool(j,3,m)+(tool(j,3,m)-elbow(j,3,m))*par(m)

Check if coordinates belong to either arm or collision plane
      IF (ptx(m).le.tool(j,1,m).and.ptx(m).ge.elbow(j,1,m).and.
     +pty(m).le.tool(j,2,m).and.pty(m).ge.elbow(j,2,m).and.
     +ptz(m).le.tool(j,3,m).and.ptz(m).ge.elbow(j,3,m).and.
     +ptx(m).le.collis(1,n+1).and.ptx(m).ge.collis(1,n).and.
     +ptx(m).le.collis(1,n+cpt/2).and.ptx(m).ge.collis(1,n+cpt/2+1).and.
     +pty(m).le.collis(2,n+1).and.pty(m).ge.collis(2,n).and.
     +pty(m).le.collis(2,n+cpt/2).and.pty(m).ge.collis(2,n+cpt/2+1).and.
     +ptz(m).le.collis(3,n+1).and.ptz(m).ge.collis(3,n).and.
     +ptz(m).le.collis(3,n+cpt/2).and.ptz(m).ge.
     +collis(3,n+cpt/2+1)) THEN
      GOTO 111
      ENDIF
      ENDIF
      ELSE IF (collis(2,n).eq.collis(2,n+1).and.
     +collis(2,n+1).eq.collis(2,n+cpt/2).and.
     +collis(2,n+cpt/2).eq.collis(2,n+cpt/2+1)) THEN

Collision plane is y=const.
Calculate the intersection coordinates (ptx,pty,ptz)

      den(m)=tool(j,2,m)-elbow(j,2,m)
      IF(den(m).eq.0.0) THEN
      GOTO 111
      ELSE
      par(m)=(collis(2,n)-elbow(j,2,m))/den(m)
      ptx(m)=tool(j,1,m)+(tool(j,1,m)-elbow(j,1,m))*par(m)
      pty(m)=collis(2,n)
      ptz(m)=tool(j,3,m)+(tool(j,3,m)-elbow(j,3,m))*par(m)

Check if coordinates belong to either arm or collision plane
      IF (ptx(m).le.tool(j,1,m).and.ptx(m).ge.elbow(j,1,m).and.
     +pty(m).le.tool(j,2,m).and.pty(m).ge.elbow(j,2,m).and.
     +ptz(m).le.tool(j,3,m).and.ptz(m).ge.elbow(j,3,m).and.
```

```
          +ptx(m).le.collis(1,n+1).and.ptx(m).ge.collis(1,n).and.
          +ptx(m).le.collis(1,n+cpt/2).and.ptx(m).ge.collis(1,n+cpt/2+1).and.
          +pty(m).le.collis(2,n+1).and.pty(m).ge.collis(2,n).and.
          +pty(m).le.collis(2,n+cpt/2).and.pty(m).ge.collis(2,n+cpt/2+1).and.
          +ptz(m).le.collis(3,n+1).and.ptz(m).ge.collis(3,n).and.
          +ptz(m).le.collis(3,n+cpt/2).and.ptz(m).ge.
          +collis(3,n+cpt/2+1)) THEN
           GOTO 111
           ENDIF
           ENDIF
           ELSE IF (collis(3,n).eq.collis(3,n+1).and.
          +collis(3,n+1).eq.collis(3,n+cpt/2).and.
          +collis(3,n+cpt/2).eq.collis(3,n+cpt/2+1)) THEN
```

Collision plane is z=const.
Calculate the intersection coordinates (ptx,pty,ptz)

```
           den(m)=tool(j,3,m)-elbow(j,3,m)
           IF (den(m).eq.0.0) THEN
           GOTO 111
           ELSE
           par(m)=(collis(3,n)-elbow(j,3,m))/den(m)
           ptx(m)=tool(j,1,m)+(tool(j,1,m)-elbow(j,1,m))*par(m)
           pty(m)=tool(j,2,m)+(tool(j,2,m)-elbow(j,2,m))*par(m)
           ptz(m)=collis(3,n)
```

Check if coordinates belong to either arm or collision plane

```
           IF (ptx(m).le.tool(j,1,m).and.ptx(m).ge.elbow(j,1,m).and.
          +pty(m).le.tool(j,2,m).and.pty(m).ge.elbow(j,2,m).and.
          +ptz(m).le.tool(j,3,m).and.ptz(m).ge.elbow(j,3,m).and.
          +ptx(m).le.collis(1,n+1).and.ptx(m).ge.collis(1,n).and.
          +ptx(m).le.collis(1,n+cpt/2).and.ptx(m).ge.collis(1,n+cpt/2+1).and.
          +pty(m).le.collis(2,n+1).and.pty(m).ge.collis(2,n).and.
          +pty(m).le.collis(2,n+cpt/2).and.pty(m).ge.collis(2,n+cpt/2+1).and.
          +ptz(m).le.collis(3,n+1).and.ptz(m).ge.collis(3,n).and.
          +ptz(m).le.collis(3,n+cpt/2).and.ptz(m).ge.
          +collis(3,n+cpt/2+1)) THEN
           GOTO 111
           ENDIF
           ENDIF
           ENDIF
       240  continue
             GOTO 112
        111  ll(m)=1000000
           penalt=penalt+ll(m)
       112  dist(m)=sqrt((tool(j,1,m)-path(1,m))**2+(tool(j,2,m)
          +-path(2,m))**2+(tool(j,3,m)-path(3,m))**2)
           sumdist=sumdist+dist(m)
           distmin=dist(m)
          distance(m)=distmin+penalt
           distsum=distsum+distance(m)
        160  continue
           ddistsum=distsum+ppenalt
           avg=sumdist/pt
             dev=0.0
           do 170 m=1,pt
             dev=dev+(dist(m)-avg)**2
        170    continue
             devv(j)=sqrt(dev/pt)
             disttl(j)=distsum
```

Call function evaluator, write out individual and fitness, and add to the summation for later averaging.

```
             call func(j,funcval)
            fitness(j)=funcval
            fitsum=fitsum+fitness(j)
               do 222 n=1,nparam
                paramsm(n)=paramsm(n)+parent(j,n)
        222     continue
```

Check to see if fitness of individual j is the best fitness.

```
              if (fitness(j).gt.best) then
                best=fitness(j)
                jbest=j
                do 24 k=1,nchrome
                  ibest(k)=iparent(j,k)
        24        continue
```

```
        endif
   30   continue
        write(24,1075) jbest,(parent(jbest,kk),kk=1,nparam),best
       write(24,1250) devv(jbest)

c  Compute parameter and fitness averages.
       fbar=fitsum/dble(npopsiz)
       do 23 n=1,nparam
         paramav(n)=paramsm(n)/dble(npopsiz)
   23   continue
c  Write output information
        if (npopsiz.eq.1) then
          write(24,1075) 1,(parent(1,k),k=1,nparam),fitness(1)
          write(24,*) ' Average Values:'
          write(24,1275) (parent(1,k),k=1,nparam),fbar
        else
          write(24,1275) (paramav(k),k=1,nparam),fbar
        endif
       write(6,1100) fbar
        write(24,1100) fbar
       write(6,1200) best
        write(24,1200) best
       write(26,1340) best
 1075 format(i3,1x,f7.2,3x,3(1x,f7.2),3x,3(1x,f7.2),
     +3x,3(1x,f7.2),3x,3(1x,f7.2),3x,3(1x,f7.2),3x,
     +3(1x,f7.2),3x,3(1x,f7.2),3x,3(1x,f7.2),
     +3x,3(1x,f7.2),3x,3(1x,f7.2),3x,3(1x,f7.2),3x,
     +3(1x,f7.2),1x,f24.18)
  1100 format(1x,'Average Fitness Value of Generation=',f24.18)
  1200 format(1x,'Maximum Fitness Value            =',f24.18)
  1250 format(1x,'Standard Deviation=',f20.16)
  1275 format(/' Average Values:',10x,37(2x,f7.2),1x,f24.18/)
  1340 format(f24.18)
       return
       end
```

E.3.3 Main Body of the Program for Stage Two

```
      implicit double precision (a-h,o-z)
    save
    include 'parmrb2.f'
      dimension parent(indmax,nparmax),child(indmax,nparmax)
       dimension fitness(indmax),nposibl(nparmax)
       dimension iparent(indmax,nchrmax),ichild(indmax,nchrmax)
       dimension g0(nparmax),g1(nparmax),ig2(nparmax)
       dimension paramsm(nparmax),paramav(nparmax),ibest(nchrmax)
       dimension parmax(nparmax),parmin(nparmax),pardel(nparmax)
       dimension rotat(indmax,mxcoor,mxpt)
       dimension dext(indmax,mxpt),sumttl(mxpt)
       dimension t(mxpt),t1(mxpt),t2(mxpt)
       dimension partfit1(indmax),partfit2(indmax)
       common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga3   / parent,iparent
      common / ga4   / fitness
      common / ga5   / g0,g1,ig2
      common / ga6   / parmax,parmin,pardel,nposibl
      common / ga7   / child,ichild
       common / ga8   / dext,sumttl
       common / ga9   / rotat
       common / ga10  / pt
       common / ga11  / t,t1,t2
       common / ga12  / coor
       common / ga13  / partfit1, partfit2
       common / ga14  / paramsm
       common / ga15  / paramav
       common /inputga/ pcross,pmutate,pcreep,maxgen,idum,irestrt,

     +            itourny,ielite,icreep,iunifrm,

     +            iskip,iend,nchild,kountmx

       call input
c  Perform necessary initialization and read the ga.restart file.
    call initial(istart,npossum,ig2sum)
```

$$$$$ Main generational processing loop. $$$$$

As for stage One

$$$$$ End of main generational processing loop. $$$$$

```
        CLOSE (24)
 1050 format(1x,' #     Binary Code',8x,'Param1 Param2 Param3',
     +        ' Param4 Param5 Param6 Param7 Param8 Param9 Param10',
     +        ' Param11 Param12 Param13 Param14 Param15 Param16',
     +        ' Param17 Param18 Param19 Param20 Param21 Param22',
     +        ' Param23 Param24 Param25 Param26 Param27',
     +        ' Param28 Param29 Param30 Param31 Param32 Param33',
     +        ' Param34 Param35 Param36 Param37 Param38 Param39',
     +        ' Param40 Param41 Param42 Param43 Param44 Param45',
     +        ' Param46 Param47 Param48 Param49 Param50 Param51',
     +        ' Param52 Param53 Param54 Param55 Param56 Fitness')
 1111 format(//'################# Generation',i5,' #################')
 1225 format(/'  Number of Crossovers      =',i5)
 1300 format(i5)
      stop
      end
```

E.3.3.1 Subroutine evalout(iskip,iend,ibest)for RRP

For stage two with niching and micro-GA for RRP

This subroutine evaluates the population, assigns fitness, establishes the best individual, and outputs information.

```
      subroutine evalout(iskip,iend,ibest)
c
c  This subroutine evaluates the population, assigns fitness,
c  establishes the best individual, and outputs information.
      implicit double precision (a-h,o-z)
      save
c
      include 'parmrb2.f'
      dimension parent(indmax,nparmax),iparent(indmax,nchrmax)
      dimension fitness(indmax)
      dimension paramsm(nparmax),paramav(nparmax),ibest(nchrmax)
      dimension rotat(indmax,mxcoor,mxpt)
      dimension dext(indmax,mxpt)
      dimension sumttl(mxpt)
      dimension t(mxpt),t1(mxpt),t2(mxpt),ll(mxpt)
      dimension partfit1(indmax),partfit2(indmax)
c
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga3   / parent,iparent
      common / ga4   / fitness
      common / ga8   / dext,sumttl
      common / ga9   / rotat
      common / ga10  / pt
      common / ga11  / t1,t2,t,ll
      common / ga12  / coor
      common / ga13  / partfit1,partfit2
      common / ga14  / paramsm

c
      fitsum=0.0
      best=0.0

      do 29 n=1,nparam
        paramsm(n)=0.0
 29   continue

      jstart=1
      jend=npopsiz
      if(iskip.ne.0) jstart=iskip
      if(iend.ne.0) jend=iend
      do 30 j=jstart,jend
        call decode(j,parent,iparent)
        if(iskip.ne.0 .and. iend.ne.0 .and. iskip.eq.iend)
     +  write(6,*) j,(iparent(j,k),k=1,nchrome),
     +              (parent(j,kk),kk=1,nparam),0.0
c
      dexter=0.0
        penalt=0.0
        timesum=0.0
      sumt=0.0
```

```
c Calculating the rotational part of the total transformation matrix from decoded parent array
c and introducing dexterity conditions and calculating the dexterity
      do 160 m=1,pt
      IF(ABS(parent(j,18*m+3)-parent(j,18*m-15)).gt.89.or.
     +ABS(parent(j,18*m+4)-parent(j,18*m-14)).gt.89.or.
     +ABS(parent(j,18*m+5)-parent(j,18*m-13)).gt.89.or.
     +ABS(parent(j,18*m+6)-parent(j,18*m-12)).gt.89.or.
     +ABS(parent(j,18*m+7)-parent(j,18*m-11)).gt.89.) THEN
          ll(m)=10000
          ELSE
          ll(m)=0.0
          ENDIF
          penalt=penalt+ll(m)

          reach=parent(j,18*m-16)*2000
          psi1=parent(j,18*m-15)*2*3.14159265/360
          psi2=parent(j,18*m-14)*2*3.14159265/360
          psi4=parent(j,18*m-13)*2*3.14159265/360
          psi5=parent(j,18*m-12)*2*3.14159265/360
          psi6=parent(j,18*m-11)*2*3.14159265/360

c RRP is the preffered main config.

     IF (parent(j,1).eq.1)  THEN
c RPR wrist
     IF(parent(j,18*m-12).eq.0.or.
     +parent(j,18*m-12).eq.180) THEN
          ll(m)=10000
          ELSE
          ll(m)=0.0
          ENDIF
          penalt=penalt+ll(m)
      rotat(j,3,1)=-sin(psi2)*cos(psi4)*cos(psi5)*cos(psi6)
     ++sin(psi2)*sin(psi4)*sin(psi6)-cos(psi2)*sin(psi5)
     +*cos(psi6)
      rotat(j,3,2)=sin(psi2)*cos(psi4)*cos(psi5)*sin(psi6)
     ++sin(psi2)*sin(psi4)*cos(psi6)+cos(psi2)*sin(psi5)
     +*sin(psi6)
      rotat(j,1,3)=cos(psi1)*cos(psi2)*cos(psi4)*sin(psi5)
     ++cos(psi1)*sin(psi2)*cos(psi5)-sin(psi1)*sin(psi4)*sin(psi5)
      rotat(j,2,3)=sin(psi1)*cos(psi2)*cos(psi4)*sin(psi5)
     ++sin(psi1)*sin(psi2)*cos(psi5)+cos(psi1)*sin(psi4)*sin(psi5)
      rotat(j,3,3)=-sin(psi2)*cos(psi4)*sin(psi5)
     ++cos(psi2)*cos(psi5)
      rotat(j,2,1)=sin(psi1)*cos(psi2)*cos(psi4)*cos(psi5)*cos(psi6)
     +-sin(psi1)*cos(psi2)*sin(psi4)*sin(psi6)+cos(psi1)*sin(psi4)
     +*cos(psi5)*cos(psi6)+cos(psi1)*cos(psi4)*sin(psi6)-sin(psi1)
     +*sin(psi2)*sin(psi5)*cos(psi6)
      rotat(j,1,2)=-cos(psi1)*cos(psi2)*cos(psi4)*cos(psi5)*sin(psi6)
     +-cos(psi1)*cos(psi2)*sin(psi4)*cos(psi6)+sin(psi1)*sin(psi4)
     +*cos(psi5)*sin(psi6)-sin(psi1)*cos(psi4)*cos(psi6)+cos(psi1)
     +*sin(psi2)*sin(psi5)*sin(psi6)

c RPY wrist
      ELSE
     IF(parent(j,18*m-12).eq.90.or.
     +parent(j,18*m-12).eq.270) THEN
          ll(m)=10000
          ELSE
          ll(m)=0.0
          ENDIF
          penalt=penalt+ll(m)
      rotat(j,3,1)= sin(psi2)*cos(psi4)*cos(psi5)
     ++cos(psi2+psi3)*sin(psi4)*cos(psi5)
      rotat(j,3,2)=sin(psi2)*cos(psi4)*sin(psi5)*sin(psi6)
     +-sin(psi2)*sin(psi4)*cos(psi6)+cos(psi2)*cos(psi5)*sin(psi6)
      rotat(j,1,3)=-cos(psi1)*cos(psi2)*cos(psi4)*sin(psi5)
     +*cos(psi6)-cos(psi1)*cos(psi2)*sin(psi4)*sin(psi6)+sin(psi1)
     +*sin(psi4)*sin(psi5)*cos(psi6)-sin(psi1)*cos(psi4)
     +*sin(psi6)+cos(psi1)*sin(psi2)*cos(psi5)*cos(psi6)
      rotat(j,2,3)=-sin(psi1)*cos(psi2)*cos(psi4)*sin(psi5)
     +*cos(psi6)-sin(psi1)*cos(psi2)*sin(psi4)*sin(psi6)-cos(psi1)
     +*sin(psi4)*sin(psi5)*cos(psi6)+cos(psi1)*sin(psi4)
     +*sin(psi6)+sin(psi1)*sin(psi2)*cos(psi5)*cos(psi6)
```

```
 rotat(j,3,3)=sin(psi2)*cos(psi4)*sin(psi5)*cos(psi6)
++sin(psi2)*sin(psi4)*sin(psi6)+cos(psi2)*cos(psi5)
+*cos(psi6)
 rotat(j,2,1)=-sin(psi1)*cos(psi2)*cos(psi4)*cos(psi5)
+-sin(psi1)*sin(psi2)*sin(psi5)-cos(psi1)
+*sin(psi4)*cos(psi5)
 rotat(j,2,2)=-sin(psi1)*cos(psi2)*cos(psi4)*sin(psi5)
+*sin(psi6)+sin(psi1)*cos(psi2)*sin(psi4)*cos(psi6)-cos(psi1)
+*cos(psi6)+sin(psi1)*sin(psi2)*cos(psi5)*sin(psi6)

 ENDIF

 IF (m.ge.4.and.m.le.9) THEN
 dext(j,m)=rotat(j,2,1)**2+rotat(j,2,2)**2
++rotat(j,1,3)**2+rotat(j,3,3)**2
++(1-abs(rotat(j,2,3)))**2
 ELSE
 dext(j,m)=rotat(j,3,1)**2+rotat(j,3,2)**2
++rotat(j,1,3)**2+rotat(j,2,3)**2
++(1-abs(rotat(j,3,3)))**2
 ENDIF
 dexter=dexter+dext(j,m)+penalt

c Calculating the time to reach each point along the path

 IF (parent(j,18*m-10).eq.0.0) then
   t(m)=0.0
 else if (parent(j,18*m-10).ne.0.0.and.
+parent(j,18*m-4).eq.0.0) then
   t(m)=reach/parent(j,18*m-10)
 else
   t1(m)=parent(j,18*m-10)/parent(j,18*m-4)
   anacc1=t1(m)**2*parent(j,18*m-4)*2
 if ((reach-anacc1).gt.0.0) then
   t2(m)=(reach-anacc1)/parent(j,18*m-10)
   t(m)=2*t1(m)+t2(m)
 else
   t(m)=sqrt(reach/parent(j,18*m-4))
 endif
 endif
   sumt=t(m)

 IF (parent(j,18*m-9).eq.0.0) then
   t(m)=0.0
  else if (parent(j,18*m-9).ne.0.0.and.
+parent(j,18*m-3).eq.0.0) then
  t(m)=psi1/parent(j,18*m-9)
 else
 t1(m)=parent(j,18*m-9)/parent(j,18*m-3)
   anacc2=t1(m)**2*parent(j,18*m-3)*2
 if ((psi1-anacc2).gt.0.0) then
   t2(m)=(psi1-anacc2)/parent(j,18*m-9)
   t(m)=2*t1(m)+t2(m)
 else
   t(m)=sqrt(psi1/parent(j,18*m-3))
 endif
 endif
   if (t(m).gt.sumt) then
   sumt=t(m)
 endif

 IF (parent(j,18*m-8).eq.0.0) then
   t(m)=0.0
  else if (parent(j,18*m-8).ne.0.0.and.
+parent(j,18*m-2).eq.0.0) then
 t(m)=psi2/parent(j,18*m-8)
 else
   t1(m)=parent(j,18*m-8)/parent(j,18*m-2)
   anacc3=t1(m)**2*parent(j,18*m-2)*2
 if ((psi2-anacc3).gt.0.0) then
   t2(m)=(psi2-anacc3)/parent(j,18*m-8)
   t(m)=2*t1(m)+t2(m)
 else
   t(m)=sqrt(psi2/parent(j,18*m-2))
```

```
      endif
      endif
      if (t(m).gt.sumt) then
        sumt=t(m)
        endif

      IF (parent(j,18*m-7).eq.0.0) then
         t(m)=0.0
      else if (parent(j,18*m-7).ne.0.0.and.
     +parent(j,18*m-1).eq.0.0) then
      t(m)=psi3/parent(j,18*m-7)
      else
        t1(m)=parent(j,18*m-7)/parent(j,18*m-1)
        anacc4=t1(m)**2*parent(j,18*m-1)*2
      if ((psi3-anacc4).gt.0.0) then
        t2(m)=(psi3-anacc4)/parent(j,18*m-7)
        t(m)=2*t1(m)+t2(m)
      else
        t(m)=sqrt(psi3/parent(j,18*m-1))
      endif
      endif
      if (t(m).gt.sumt) then
        sumt=t(m)
        endif

      IF (parent(j,18*m-6).eq.0.0) then
         t(m)=0.0
      else if (parent(j,18*m-6).ne.0.0.and.
     +parent(j,18*m).eq.0.0) then
      t(m)=psi4/parent(j,18*m-6)
      else
        t1(m)=parent(j,18*m-6)/parent(j,18*m)
        anacc5=t1(m)**2*parent(j,18*m)*2
      if ((psi4-anacc5).gt.0.0) then
        t2(m)=(psi4-anacc5)/parent(j,18*m-6)
        t(m)=2*t1(m)+t2(m)
      else
        t(m)=sqrt(psi4/parent(j,18*m))
      endif
      endif
      if (t(m).gt.sumt) then
        sumt=t(m)
        endif

      IF (parent(j,18*m-5).eq.0.0) then
         t(m)=0.0
      else if (parent(j,18*m-5).ne.0.0.and.
     +parent(j,18*m+1).eq.0.0) then
      t(m)=psi5/parent(j,18*m-5)
      else
        t1(m)=parent(j,18*m-5)/parent(j,18*m+1)
        anacc6=t1(m)**2*parent(j,18*m+1)*2
      if ((psi5-anacc6).gt.0.0) then
        t2(m)=(psi5-anacc6)/parent(j,18*m-5)
        t(m)=2*t1(m)+t2(m)
      else
        t(m)=sqrt(psi5/parent(j,18*m+1))
      endif
      endif
      if (t(m).gt.sumt) then
        sumt=t(m)
        endif
c         write(24,*) m,sumt
        sumttl(m)=sumt
        timesum=timesum+sumttl(m)+penalt
 160  continue
c         write(24,*) dexter,timesum
      partfit1(j)=dexter
      partfit2(j)=timesum

c  Call function evaluator, write out individual and fitness, and add
c  to the summation for later averaging.
        call func(j,funcval)
        fitness(j)=funcval
```

```
c     write(24,*) j,(parent(j,kk),kk=1,nparam),fitness(j)
      fitsum=fitsum+fitness(j)

      do 222 n=1,nparam
        paramsm(n)=paramsm(n)+parent(j,n)
 222    continue
c
c Check to see if fitness of individual j is the best fitness.
      if (fitness(j).gt.best) then
        best=fitness(j)
        jbest=j
        do 24 k=1,nchrome
          ibest(k)=iparent(j,k)
 24       continue
      endif
 30   continue
c
      write(24,*) jbest,(parent(jbest,kk),kk=1,nparam),best

c Compute parameter and fitness averages.
      fbar=fitsum/dble(npopsiz)

      do 23 n=1,nparam
        paramav(n)=paramsm(n)/dble(npopsiz)
 23   continue
c
c Write output information
c     if (npopsiz.eq.1) then
c       write(24,*) 1,(iparent(1,k),k=1,nchrome),
c    +              (parent(1,k),k=1,nparam),fitness(1)
c       write(24,*) ' Average Values:'
c       write(24,1275) (parent(1,k),k=1,nparam),fbar
c     else
c       write(24,1275) (paramav(k),k=1,nparam),fbar
c     endif
      write(6,1100)  fbar
      write(24,1100) fbar
      write(26,1340) best
      write(6,1200)  best
      write(24,1200) best

c 1075 format(i4,3x,18(1x,f7.2),3x,18(1x,f7.2),3x,18(1x,f7.2),
c    +3x,18(1x,f7.2),3x,18(1x,f7.2),3x,18(1x,f7.2),3x,18(1x,f7.2),
c    +3x,18(1x,f7.2),3x,18(1x,f7.2),3x,18(1x,f7.2),3x,18(1x,f7.2),
c    +3x,18(1x,f7.2),3x,f24.18)
 1100 format(1x,'Average Fitness Value of Generation=',f24.18)
 1200 format(1x,'Maximum Fitness Value              =',f24.18)
 1275 format(' Average Values:',10x,217(2x,f7.2),1x,f24.18)
 1340 format(f24.18)
      return
      end
```

E.3.3.2 Subroutine evalout(iskip,iend,ibest) for stage two with niching and micro-GA for RRR

```
c This subroutine evaluates the population, assigns fitness,
c establishes the best individual, and outputs information.
      implicit double precision (a-h,o-z)
      save
c
      include 'parmrb2.f'
      dimension parent(indmax,nparmax),iparent(indmax,nchrmax)
      dimension fitness(indmax)
      dimension paramsm(nparmax),paramav(nparmax),ibest(nchrmax)
      dimension rotat(indmax,mxcoor,mxpt)
      dimension dext(indmax,mxpt),sumttl(mxpt)
      dimension t(mxpt),t1(mxpt),t2(mxpt), ll(mxpt)
      dimension partfit1(indmax),partfit2(indmax)
c
      common / ga1   / npopsiz,nowrite
      common / ga2   / nparam,nchrome
      common / ga3   / parent,iparent
      common / ga4   / fitness
      common / ga8   / dext, sumttl
      common / ga9   / rotat
```

```
      common / ga10  / pt
      common / ga11  / t1,t2,t,ll
      common / ga12  / coor
      common / ga13  / partfit1,partfit2
      common / ga14  / paramsm

c
      fitsum=0.0
      best=0.0

      do 29 n=1,nparam
        paramsm(n)=0.0
 29   continue

      jstart=1
      jend=npopsiz
      if(iskip.ne.0) jstart=iskip
      if(iend.ne.0) jend=iend
      do 30 j=jstart,jend
        call decode(j,parent,iparent)
        if(iskip.ne.0 .and. iend.ne.0 .and. iskip.eq.iend)
     +  write(6,*) j,(iparent(j,k),k=1,nchrome),
     +                (parent(j,kk),kk=1,nparam),0.0
c
       dexter=0.0
       penalt=0.0
       timesum=0.0
       sumt=0.0
c Calculating the rotational part of the total transformation matrix from decoded parent array
c and introducing dexterity conditions and calculating the dexterity
      do 160 m=1,pt

      IF(ABS(parent(j,18*m+4)-parent(j,18*m-14)).gt.89.or.
     +ABS(parent(j,18*m+5)-parent(j,18*m-13)).gt.89.or.
     +ABS(parent(j,18*m+6)-parent(j,18*m-12)).gt.89.or.
     +ABS(parent(j,18*m+7)-parent(j,18*m-11)).gt.89.or.
     +ABS(parent(j,18*m+8)-parent(j,18*m-10)).gt.89.) THEN
          ll(m)=10000
          ELSE
          ll(m)=0.0
          ENDIF
          penalt=penalt+ll(m)

          psi1=parent(j,18*m-15)*2*3.14159265/360
          psi2=parent(j,18*m-14)*2*3.14159265/360
          psi3=parent(j,18*m-13)*2*3.14159265/360
          psi4=parent(j,18*m-12)*2*3.14159265/360
          psi5=parent(j,18*m-11)*2*3.14159265/360
          psi6=parent(j,18*m-10)*2*3.14159265/360
c RPR wrist
      IF (parent(j,2).eq.1) THEN
      IF (parent(j,18*m-11).eq.0.or.
     +parent(j,18*m-11).eq.180) THEN
       ll(m)=10000
       ELSE
       ll(m)=0.0
       ENDIF
       penalt=penalt+ll(m)

c and RRR preffered config.
        rotat(j,3,1)=-sin(psi2+psi3)*cos(psi4)*cos(psi5)*cos(psi6)
     ++sin(psi2+psi3)*sin(psi4)*sin(psi6)-cos(psi2+psi3)*sin(psi5)
     +*cos(psi6)
        rotat(j,3,2)=sin(psi2+psi3)*cos(psi4)*cos(psi5)*sin(psi6)
     ++sin(psi2+psi3)*sin(psi4)*cos(psi6)+cos(psi2+psi3)*sin(psi5)
     +*sin(psi6)
        rotat(j,1,3)=-cos(psi1)*cos(psi2+psi3)*cos(psi4)*sin(psi5)
     +-cos(psi1)*sin(psi2+psi3)*cos(psi5)-sin(psi1)*sin(psi4)*sin(psi5)
        rotat(j,2,3)=-sin(psi1)*cos(psi2+psi3)*cos(psi4)*sin(psi5)
     +-sin(psi1)*sin(psi2+psi3)*cos(psi5)-cos(psi1)*sin(psi4)*sin(psi5)
        rotat(j,3,3)=sin(psi2+psi3)*cos(psi4)*sin(psi5)
     +-cos(psi2+psi3)*cos(psi5)
c RPY wrist
      ELSE
```

```
      IF(parent(j,18*m-11).eq.90.or.
     +parent(j,18*m-11).eq.270) THEN
           ll(m)=10000
           ELSE
           ll(m)=0.0
           ENDIF
           penalt=penalt+ll(m)
        rotat(j,3,1)= sin(psi2+psi3)*cos(psi4)*cos(psi5)
     ++cos(psi2+psi3)*sin(psi4)*cos(psi5)
        rotat(j,3,2)=sin(psi2+psi3)*cos(psi4)*sin(psi5)*sin(psi6)
     +-sin(psi2+psi3)*sin(psi4)*cos(psi6)+cos(psi2+psi3)*sin(psi4)
     +*sin(psi5)*sin(psi6)+cos(psi2+psi3)*cos(psi4)*cos(psi6)
        rotat(j,1,3)=cos(psi2)*cos(psi2+psi3)*cos(psi4)*sin(psi5)
     +*cos(psi6)
     ++cos(psi2)*cos(psi2+psi3)*sin(psi4)*sin(psi6)-cos(psi1)
     +*cos(psi2+psi3)*sin(psi4)
     +*sin(psi5)*cos(psi6)+cos(psi1)*cos(psi2+psi3)*cos(psi4)
     +*sin(psi6)+sin(psi1)*cos(psi5)*cos(psi6)
        rotat(j,2,3)=sin(psi1)*cos(psi2+psi3)*cos(psi4)*sin(psi5)
     +*cos(psi6)
     ++sin(psi1)*cos(psi2+psi3)*sin(psi4)*sin(psi6)-sin(psi2)
     +*sin(psi2+psi3)*sin(psi4)
     +*sin(psi5)*cos(psi6)+sin(psi2)*sin(psi2+psi3)*cos(psi4)
     +*sin(psi6)-cos(psi1)*cos(psi5)*cos(psi6)
        rotat(j,3,3)=sin(psi2+psi3)*cos(psi4)*sin(psi5)*cos(psi6)
     ++sin(psi2+psi3)*sin(psi4)*sin(psi6)+cos(psi2+psi3)*cos(psi4)
     +*sin(psi5)*cos(psi6)-cos(psi2+psi3)*cos(psi4)*sin(psi6)
      ENDIF

      IF (m.ge.4.and.m.le.9) THEN
        dext(j,m)=rotat(j,2,1)**2+rotat(j,2,2)**2

     ++rotat(j,1,3)**2+rotat(j,3,3)**2+(1-abs(rotat(j,2,3)))**2
            ELSE
        dext(j,m)=rotat(j,3,1)**2+rotat(j,3,2)**2
     ++rotat(j,1,3)**2+rotat(j,2,3)**2+(1-abs(rotat(j,3,3)))**2
            ENDIF
      dexter=dexter+dext(j,m)+penalt

c Calculating the time to reach each point along the path


      IF (parent(j,18*m-9).eq.0.0) then
         t(m)=0.0
      else if (parent(j,18*m-9).ne.0.0.and.
     +parent(j,18*m-3).eq.0.0) then
        t(m)=psi1/parent(j,18*m-9)
      else
        t1(m)=parent(j,18*m-9)/parent(j,18*m-3)
        anacc1=t1(m)**2*parent(j,18*m-3)*2
      if ((psi1-anacc1).gt.0.0) then
        t2(m)=(psi1-anacc1)/parent(j,18*m-9)
        t(m)=2*t1(m)+t2(m)
      else
        t(m)=sqrt(psi1/parent(j,18*m-3))
      endif
      endif
        sumt=sumt+t(m)

      IF (parent(j,18*m-8).eq.0.0) then
         t(m)=0.0
       else if (parent(j,18*m-8).ne.0.0.and.
     +parent(j,18*m-2).eq.0.0) then
       t(m)=psi2/parent(j,18*m-8)
      else
      t1(m)=parent(j,18*m-8)/parent(j,18*m-2)
        anacc2=t1(m)**2*parent(j,18*m-2)*2
      if ((psi2-anacc2).gt.0.0) then
        t2(m)=(psi2-anacc2)/parent(j,18*m-8)
        t(m)=2*t1(m)+t2(m)
      else
        t(m)=sqrt(psi2/parent(j,18*m-2))
      endif
      endif
```

```
   if (t(m).gt.sumt) then
   sumt=t(m)
 endif

 IF (parent(j,18*m-7).eq.0.0) then
    t(m)=0.0
  else if (parent(j,18*m-7).ne.0.0.and.
+parent(j,18*m-1).eq.0.0) then
 t(m)=psi3/parent(j,18*m-7)
 else
   t1(m)=parent(j,18*m-7)/parent(j,18*m-1)
   anacc3=t1(m)**2*parent(j,18*m-1)*2
 if ((psi3-anacc3).gt.0.0) then
   t2(m)=(psi3-anacc3)/parent(j,18*m-7)
   t(m)=2*t1(m)+t2(m)
 else
   t(m)=sqrt(psi3/parent(j,18*m-1))
 endif
 endif
 if (t(m).gt.sumt) then
   sumt=t(m)
   endif

 IF (parent(j,18*m-6).eq.0.0) then
    t(m)=0.0
 else if (parent(j,18*m-6).ne.0.0.and.
+parent(j,18*m).eq.0.0) then
 t(m)=psi4/parent(j,18*m-6)
 else
   t1(m)=parent(j,18*m-6)/parent(j,18*m)
   anacc4=t1(m)**2*parent(j,18*m)*2
 if ((psi4-anacc4).gt.0.0) then
   t2(m)=(psi4-anacc4)/parent(j,18*m-6)
   t(m)=2*t1(m)+t2(m)
 else
   t(m)=sqrt(psi4/parent(j,18*m))
 endif
 endif
 if (t(m).gt.sumt) then
   sumt=t(m)
   endif

 IF (parent(j,18*m-5).eq.0.0) then
    t(m)=0.0
 else if (parent(j,18*m-5).ne.0.0.and.
+parent(j,18*m+1).eq.0.0) then
 t(m)=psi5/parent(j,18*m-5)
 else
   t1(m)=parent(j,18*m-5)/parent(j,18*m+1)
   anacc5=t1(m)**2*parent(j,18*m+1)*2
 if ((psi5-anacc5).gt.0.0) then
   t2(m)=(psi5-anacc5)/parent(j,18*m-5)
   t(m)=2*t1(m)+t2(m)
 else
   t(m)=sqrt(psi5/parent(j,18*m+1))
 endif
 endif
 if (t(m).gt.sumt) then
   sumt=t(m)
   endif

 IF (parent(j,18*m-4).eq.0.0) then
    t(m)=0.0
 else if (parent(j,18*m-4).ne.0.0.and.
+parent(j,18*m+2).eq.0.0) then
 t(m)=psi6/parent(j,18*m-4)
 else
   t1(m)=parent(j,18*m-4)/parent(j,18*m+2)
   anacc6=t1(m)**2*parent(j,18*m+2)*2
 if ((psi6-anacc6).gt.0.0) then
   t2(m)=(psi6-anacc6)/parent(j,18*m-4)
   t(m)=2*t1(m)+t2(m)
 else
   t(m)=sqrt(psi6/parent(j,18*m+2))
```

```
      endif
      endif
      if (t(m).gt.sumt) then
        sumt=t(m)
        endif

        sumttl(m)=sumt
        timesum=timesum+sumttl(m)+penalt
 160  continue

      partfit1(j)=dexter
      partfit2(j)=timesum

c  Call function evaluator, write out individual and fitness, and add
c  to the summation for later averaging.
        call func(j,funcval)
        fitness(j)=funcval
c       write(24,*) j,(parent(j,kk),kk=1,nparam),fitness(j)
        fitsum=fitsum+fitness(j)

        do 222 n=1,nparam
          paramsm(n)=paramsm(n)+parent(j,n)
 222      continue
c
c  Check to see if fitness of individual j is the best fitness.
        if (fitness(j).gt.best) then
          best=fitness(j)
          jbest=j
          do 24 k=1,nchrome
            ibest(k)=iparent(j,k)
 24         continue
        endif
 30   continue
c
      write(24,*) jbest,(parent(jbest,kk),kk=1,nparam),best

c  Compute parameter and fitness averages.
      fbar=fitsum/dble(npopsiz)
c
      do 23 n=1,nparam
        paramav(n)=paramsm(n)/dble(npopsiz)
 23   continue
c
c  Write output information
c     if (npopsiz.eq.1) then
c        write(24,*) 1,(iparent(1,k),k=1,nchrome),
c    +               (parent(1,k),k=1,nparam),fitness(1)
c       write(24,*) ' Average Values:'
c       write(24,1275) (parent(1,k),k=1,nparam),fbar
c     else
c       write(24,1275) (paramav(k),k=1,nparam),fbar
c     endif
      write(6,1100)  fbar
      write(24,1100) fbar
      write(26,*) best
      write(6,1200)  best
      write(24,1200) best

c 1075 format(i3,1x,56(1x,f6.2),1x,f8.6)
 1100 format(1x,'Average Fitness Value of Generation=',f20.14)
 1200 format(1x,'Maximum Fitness Value             =',f20.14)
 1275 format(' Average Values:',10x,56(2x,f6.2),1x,f20.14)
c 1340 format(f20.14)
      return
      end
```

E.3.3.3 Subroutine func(j,funcval) for stage two

```
       implicit double precision (a-h,o-z)
  save
  include 'parmrb2.f'
  dimension partfit1(indmax),partfit2(indmax)
       common / ga13  / partfit1, partfit2
       common /inputga/pcross,pmutate,pcreep,maxgen,idum,irestrt,
```

```
     +          itourny,ielite,icreep,iunifrm,iskip,iend,
     +          nchild, kountmx

      funcval=1/partfit1(j) + 1/partfit2(j)
      return
     end
```

E.3.4 Data Input files

E.3.4.1 Input for RRP for stage one

```
$garbap
 irestrt=0,
 npopsiz=100,
 nparam=36,
 pmutate=0.01,
 maxgen=2000,
 idum=-10000,
 pcross=0.5,
 itourny=1,
 ielite=1,
 icreep=1,
 pcreep=0.06,
 iunifrm=1,
iniche=1,
microga=1,
nchild=1,
iskip=0, iend=0,
nowrite=1,
kountmx=50,
 parmin=0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0.0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
 0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
parmax=1.0,360.0,360.0,1.0,360.0,360.0,1.0,360.0,360.0,1.0,360.0,360.0,1.0,360.0,360.0,1.0,360.0,360.0,1.0,36
0.0,360.0,1.0,360.0,360.0,1.0,360.0,360.0,1.0,
 360.0,360.0,1.0,360.0,360.0,1.0,360.0,360.0,
nposibl=64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
64,64,64,
 coor=3,
 pt=12,
 cpt=20,
 path=1000,-1000,200,   0,-1000,200,   -1000,-1000,200,
-1000,-700,513,   0,-700,513,   1000,-700,513,
1000,-700,887,   0,-700,887,   -1000,-700,887,
-1000,0,948,   0,0,948,   1000,0,948,
 collis=1000,-1150,500,1000,-850,500,1000,-850,526,1000,-1000,526,1000,-
1000,1248,1000,1000,1248,1000,1000,526,1000,850,526,1000,850,500,1000,1150,500,-1000,-1150,500,-1000,-
850,500,-1000,-850,526,-1000,-1000,526,-1000,-1000,1248,-1000,1000,1248,-1000,1000,526,-1000,850,526,-
1000,850,500,-1000,1150,500,
 $end
```

E.3.4.2 Input file for RRR for stage one

```
$garbar
irestrt=0,
microga=1,
 npopsiz=100,
 nparam=37,
 pmutate=0.01,
 maxgen=2000,
 idum=-10000,
 pcross=0.5,
 itourny=1,
 ielite=1,
 icreep=1,
 pcreep=0.06,
 iunifrm=1,
 iniche=1,
 nchild=1,
 iskip=0, iend=0,
 nowrite=1,
 kountmx=50,
parmin=0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
parmax=1.0,360.0,360.0,360.0,360.0,360.0,360.0,360.0,360.0,360.0,360.0,360.0,360.0,360.0,360.0,360.0,360.0,360.0
,360.0,360.0,360.0,360.0,360.0,360.0,360.0,360.0,360.0,360.0,
360.0,360.0,360.0,360.0,360.0,360.0,360.0,360.0,360.0,
nposibl=64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
64,64,
```

```
nichflg=1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,11,11,1,1,1,1,1,1,1,11,1,1,
    coor=3,
    pt=12,
    cpt=20,
    path=1000,-1000,200,0,-1000,200,-1000,-1000,200,-1000,-700,513,0,-700,513,1000,-700,513,1000,-700,887,0,-
700,887,-1000,-700,887,-1000,0,948,0,0,948,1000,0,948,
    collis=1000,-1150,500,1000,-850,500,1000,-850,526,1000,-1000,526,1000,-
1000,1248,1000,1000,1248,1000,1000,526,1000,850,526,1000,850,500,1000,1150,500,-1000,-1150,500,-1000,-
850,500,-1000,-850,526,-1000,-1000,526,-1000,-1000,1248,-1000,1000,1248,-1000,1000,526,-1000,850,526,-
1000,850,500,-1000,1150,500,
    $end
```

E.3.4.3 Input for RRR for stage two

```
$garbbr
irestrt=0,
microga=1,
npopsiz=200,
nparam=272,
pmutate=0.01,
maxgen=2000,
idum=-10000,
pcross=0.9,
itourny=1,
ielite=1,
icreep=1,
pcreep=0.06,
iunifrm=1,
iniche=1,
nchild=1,
iskip=0, iend=0,
nowrite=1,
kountmx=50,
parmin= 0.47,0.0,
313.0,39.0,273.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
102.0,85.0,142.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
45.0,136.0,96.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
176.0,357.0,273.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
268.0,33.0,228.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
148.0,256.0,262.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
148.0,182.0,90.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
73.0,308.0,222.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
228.0,296.0,56.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
5.0,279.0,256.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
148.0,0.0,222.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
10.0,348.0,279.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
325.0,302.0,102.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
90.0,273.0,216.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
33.0,273.0,228.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
parmax= 0.50,1.0,
316.0,42.0,276.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
105.0,88.0,145.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
48.0,139.0,99.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
179.0,357.0,276.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
271.0,36.0,231.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
151.0,259.0,265.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
151.0,185.0,93.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
76.0,311.0,225.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
231.0,299.0,59.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
8.0,282.0,259.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
151.0,3.0,225.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
13.0,351.0,282.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
328.0,305.0,105.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
93.0,276.0,219.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
36.0,276.0,231.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
nposibl=4,2,
4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
```

```
4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
nichflg=1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
coor=3,
pt=15,
$end
```

E.3.4.4 Input for RRP for stage two

```
$garbbp
 irestrt=0,
 microga=1,
 npopsiz=200,
 nparam=217,
 pmutate=0.01,
 maxgen=2000,
 idum=-10000,
 pcross=0.9,
 itourny=1,
 ielite=1,
 icreep=1,
 pcreep=0.06,
 iniche=1,
 iunifrm=1,
 nchild=1,
 iskip=0, iend=0,
 nowrite=1,
 kountmx=50,
 parmin= 0.0,
 0.75,136.0,279.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
 0.47,268.0,79.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
 0.74,39.0,279.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
 0.64,39.0,290.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
 0.42,90.0,302.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
 0.61,319.0,62.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
 0.77,325.0,56.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
 0.48,279.0,45.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
 0.74,33.0,308.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
 0.66,170.0,39.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
 0.43,210.0,353.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
 0.69,182.0,313.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
 parmax= 1.0,
 0.78,139.0,282.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
 0.50,271.0,82.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
 0.77,42.0,282.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
 0.67,42.0,293.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
 0.45,93.0,305.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
 0.64,322.0,65.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
 0.80,328.0,59.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
 0.51,282.0,48.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
 0.77,36.0,311.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
 0.69,173.0,42.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
 0.46,213.0,356.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
 0.72,185.0,316.0,360.0,360.0,360.0,1.5,1.5,1.5,1.5,1.5,1.5,3.0,3.0,3.0,3.0,3.0,3.0,
 nposibl=2,
 4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
 4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
```

```
4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
4,4,4,64,64,64,64,64,64,64,64,64,64,64,64,64,64,64,
nichflg=1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
coor=3,
pt=12,
$end
```

E.4 GRASP File for the Selected Unit Workcell

```
polyprism %edge1 height 5000.0 axis y
0 0
1000 0
1000 75
525 75
525 2100
1000 2100
1000 2175
0 2175
0 2100
475 2100
475 75
0 75;

polyprism %girder height 1500 axis x
0 0
0 300
26 300
26 142
748 142
748 300
774 300
774 0
748 0
748 142
26 142
26 0;
copy %girder %g1 %g2 ;
set %bridge = %edge1
%girder (shift x 525 y 200 z 1220)
%g1 (shift x 525 y 2200 z 1220)
%g2 (shift x 525 y 4200 z 1220)
;

copy targ targ_C31 targ_C32 targ_C33 targ_C34 targ_C35 targ_C36 targ_C37 targ_C38 ;
copy targ targ_C39 targ_C310 targ_C311 targ_C312 targ_C313 targ_C314 targ_C315;
copy targ targ_C316 targ_C317 targ_C318 targ_C319 targ_C320;
copy targ targ_P31 targ_P32 targ_P33 targ_P34 targ_P35 targ_P36 targ_arm ;
copy targ targ_P37 targ_P38 targ_P39 targ_P310 targ_P311 targ_P312 ;

to %bridge add targ_C31 (shift x 2000 y 200 z 1220 );
to %bridge add targ_C32 (shift x 2000 y  500 z 1220 );
to %bridge add targ_C33 (shift x 2000 y 500 z 1246);
to %bridge add targ_C34 (shift x 2000 y 342 z 1246);
to %bridge add targ_C35 (shift x 2000 y 342 z 1968);
to %bridge add targ_C36 (shift x 2000 y 2342 z 1968);
```

```
to %bridge add targ_C37 (shift x 2000 y 2342 z 1246);
to %bridge add targ_C38 (shift x 2000 y 2200 z 1246);
to %bridge add targ_C39 (shift x 2000 y 2200 z 1220);
to %bridge add targ_C310 (shift x 2000 y 2500 z 1220);
to %bridge add targ_C311 (shift x 540 y 200 z 1220);
to %bridge add targ_C312 (shift x 540 y 500 z 1220);
to %bridge add targ_C313 (shift x 540 y 500 z 1246);
to %bridge add targ_C314 (shift x 540 y 342 z 1246);
to %bridge add targ_C315 (shift x 540 y 342 z 1968);
to %bridge add targ_C316 (shift x 540 y 2342 z 1968);
to %bridge add targ_C317 (shift x 540 y 2342 z 1246 );
to %bridge add targ_C318 (shift x 540 y 2200 z 1246);
to %bridge add targ_C319 (shift x 540 y 2200 z 1220);
to %bridge add targ_C320 (shift x 540 y 2500 z 1220);

polyprism %collis height 1460 axis x
1220 200
1220 500
1246 500
1246 342
1968 342
1968 2342
1246 2342
1246 2200
1220 2200
1220 2500
1220 2200
1246 2200
1246 2342
1968 2342
1968 342
1246 342
1246 500
1220 500
1220 200
;
set collision = %collis (shift x 540);

to %bridge add targ_P31 (shift x 825 y 350 z 950);
to %bridge add targ_P32 (shift x 1375 y 350 z 950);
to %bridge add targ_P33 (shift x 1925 y 350 z 950);
to %bridge add targ_P34 (shift x 825 y 800 z 1265);
to %bridge add targ_P35 (shift x 1375 y 800 z 1265);
to %bridge add targ_P36 (shift x 1925 y 800 z 1265);
to %bridge add targ_P37 (shift x 825 y 600 z 1600);
to %bridge add targ_P38 (shift x 1375 y 600 z 1600);
to %bridge add targ_P39 (shift x 1925 y 600 z 1600);
to %bridge add targ_P310 (shift x 825 y 1350 z 1750);
to %bridge add targ_P311 (shift x 1375 y 1350 z 1750);
to %bridge add targ_P312 (shift x 1925 y 1350 z 1750);

to %bridge add targ_arm (shift x 1255 y 1350 z 750);

Note - set turn_bridge = %bridge (rotate z 180)
collision (rotate z 180);
stop
```

E.5 GRASP File for RRP Configuration with Four-path Track

```
robot rrp new type phd1
joint 1 revolute z
joint 2 revolute y
joint 3 prismatic z
joint 4 (shift z 1500) revolute z
joint 5 (shift z 45) revolute y
joint 6 (shift z 25) revolute z
TAP (shift x 600.0)
minimum 165 302 0    62  120 57
maximum 222 350 2000 354 354 360
initial 0 0 0 0 0 0
park 0 0 0 0 0 0
velocity 90 90 90 90 90 90
acceleration 180 180 180 180 180 180
maximum_linear_speed 800
```

```
;
cylinder base length 150 diameter 200 tolerance 4;
to rrp add base (shift z -250);
cylinder %cyl1 length 80.0 diameter 200.0 tolerance 4;
set cyl = %cyl1 (rotate x -90 shift y -25 );
to rrp_j2 add cyl;
cuboid arm2 60 60 2000;
to rrp_j3 add arm2 (shift x -30 y -30 z -500);

cylinder rot1 length 20 diameter 80 tolerance 4;
to rrp_j4 add rot1;
cylinder rot2 length 50 diameter 50 tolerance 4;
to rrp_j5 add rot2 (rotate x -90 shift y -25);
cylinder rot3 length 20 diameter 50 tolerance 4;
to rrp_j6 add rot3;

revsolid %noz angle 360 increment 30
0 0
20 0
10 200
0 200;
set G_TCP=;
set gripper = %noz
G_TCP (shift z 200);
tool gripper TCP G_TCP;
mount gripper (shift x -600 z 20) on rrp;

note - cuboid %fr1 50.0 1200.0 50.0;
note - cuboid %fr2 50.0 1200.0 50.0;
note - cuboid %fr3 1100.0 50.0 50.0;
note - cuboid %fr4 1100.0 50.0 50.0;
note - cuboid %deck 1200.0 1200.0 30.0;
note - set frame = %fr1
note - %fr2 (shift x 1150.0)
note - %fr3 (shift x 50.0)
note - %fr4 (shift x 50.0 y 1150.0)
note - %deck (shift z 50.0);

note - cylinder %wh1 length 50.0 diameter 200.0 tolerance 2;
note - cuboid %ver1 50.0 30.0 300.0;
note - set %wheel = %wh1 (rotate x 90.0)
note - %ver1 (shift x -25.0 z -50.0);
note - copy %wheel %wheel2 %wheel3 %wheel4;
note - to frame add %wheel (shift x 200.0 y -30.0)
note - %wheel2 (shift x 1000.0 y -30.0)
note - %wheel3 (shift x 200.0 y 1230.0 rotate z 180.0)
note - %wheel4 (shift x 1000.0 y 1230.0 rotate z 180.0);
note - to workplace add frame (shift x -600.0 y -600.0 z -130.0);
stop
```

E.6 GRASP File for RRR Configuration with Five-path Track

```
robot rrr new type phd2
joint 1 revolute z
joint 2 revolute y
joint 3 ( shift z 1350) revolute y
joint 4 (shift z 850) revolute z
joint 5 (shift z 45) revolute y
joint 6 (shift z 25) revolute z
TAP (shift x 600.0)
minimum 177 6   97  62  120 57
maximum 260 360 274 354 354 360
initial 0 0 0 0 0 0
park 0 0 0 0 0 0
velocity 90 90 90 90 90 90
acceleration 180 180 180 180 180 180
;
cylinder base length 150 diameter 200 tolerance 4;
to rrr add base (shift z -250);
cylinder %cyl1 length 80.0 diameter 200.0 tolerance 4;
set cyl = %cyl1 (rotate x -90 shift y -25 );
to rrr_j1 add cyl;
cuboid L1 60 60 1200;
to rrr_j2 add L1 (shift x-30 y -30 z 100);
```

```
cylinder %cyl2 length 60 diameter 100 tolerance 4;
cuboid %L2 60 60 800;
set arm2 = %cyl2 ( rotate x -90 shift y -30 )
%L2 (shift x -30 y -30 z 50);
to rrr_j3 add arm2 ;

cylinder rot1 length 20 diameter 80 tolerance 4;
to rrr_j4 add rot1;
cylinder rot2 length 50 diameter 50 tolerance 4;
to rrr_j5 add rot2 (rotate x -90 shift y -25);
cylinder rot3 length 20 diameter 50 tolerance 4;
to rrr_j6 add rot3;

revsolid %noz angle 360 increment 30
0 0
20 0
10 200
0 200;
set G_TCP=;
set gripper = %noz
G_TCP (shift z 200);
tool gripper TCP G_TCP;
mount gripper (shift x -600 z 20) on rrr;

note - cuboid %fr1 50.0 1200.0 50.0;
note - cuboid %fr2 50.0 1200.0 50.0;
note - cuboid %fr3 1100.0 50.0 50.0;
note - cuboid %fr4 1100.0 50.0 50.0;
note - cuboid %deck 1200.0 1200.0 30.0;
note - set frame = %fr1
note - %fr2 (shift x 1150.0)
note - %fr3 (shift x 50.0)
note - %fr4 (shift x 50.0 y 1150.0)
note - %deck (shift z 50.0);

cylinder %wh1 length 50.0 diameter 200.0 tolerance 2;
cuboid %ver1 50.0 30.0 300.0;
set %wheel = %wh1 (rotate x 90.0)
%ver1 (shift x -25.0 z -50.0);
copy %wheel %wheel2 %wheel3 %wheel4;
to frame add %wheel (shift x 200.0 y -30.0)
%wheel2 (shift x 1000.0 y -30.0)
%wheel3 (shift x 200.0 y 1230.0 rotate z 180.0)
%wheel4 (shift x 1000.0 y 1230.0 rotate z 180.0);
to workplace add frame (shift x -600.0 y -600.0 z -130.0);
to WORKPLACE add TARG
 rrr
;
Refobject_list
rrr
WORKPLACE
;
path PTP70 ptpt percentage velocity 70.0000 acceleration 60.000 ;
path SLP50 straight speed 40.0000 acceleration 15.000 ;
track TR
park path PTP70,
position WORKPLACE (shift X 1000 y -1000 z 200) path PTP70 ,
position WORKPLACE (shift X 0.0 y -1000 z 200) path SLP50 ,
position WORKPLACE (shift X -1000 y -1000 z 200) path SLP50 ,
position WORKPLACE (shift X -1000 y -700 z 513) path PTP70 ,
position WORKPLACE (shift X 0 y -700 z 513) path SLP50 ,
position WORKPLACE (shift X 1000 y -700 z 513) path SLP50 ,
position WORKPLACE (shift X 1000 y -700 z 887) path PTP70 ,
position WORKPLACE (shift X 0 y -700 z 887) path SLP50 ,
position WORKPLACE (shift X -1000 y -700 z 887) path SLP50 ,
position WORKPLACE (shift X -1000 y 0 z 948) path PTP70 ,
position WORKPLACE (shift X 0 y 0  z 948) path SLP50 ,
position WORKPLACE (shift X 1000 y 0 z 948) path SLP50 ,
position WORKPLACE (shift X -1000 y 0 z 200) path PTP70 ,
position WORKPLACE (shift X 0 y 0  z 200) path SLP50 ,
position WORKPLACE (shift X 1000 y 0 z 200) path SLP50 ,
park path PTP70 ,
;
stop
```

APPENDIX F. ACRONYMS

3-D	Three Dimension
ACD	Acoustic Crack Detector
AF	Automated Facility
AE	Acoustic Emission
CARL	Construction Automation and Robotics Laboratory
CP	Continuous Path
CPM	Controlled Path Motion
CT	Computer Tomography
DOF	Degree(s) of Freedom
ED	Engineering Design
EPR	Electrochemical Potential and Resistance Measurement
GA	Genetic Algorithm
GRASP	Robot Simulation Software by BYG
MCD	Magnetic Crack Detector
MPIR	Multi-purpose Interior Robot
NDT	Non-destructive Testing
PTP	Point to Point Path
QFD	Quality Function Deployment
RBPR	Robotic Bridge Paint Removal System
ROBCAD	Robot Simulation Software
RPR	Roll-pitch-roll Wrist
RPY	Roll-pitch-yaw Wrist
RRP	Spherical Configuration (revolute-revolute-prismatic)
RRR	Articulated Configuration (revolute-revolute-revolute)
RT	Radiographic Testing
SMF	Single Figure of Merit
SGA	Simple Genetic Algorithm
TBD	Task Based Design
UT	Ultrasonic Testing

Appendix G - Tables and Figures from External Papers Used for Verification

REFERENCES

3M - Abrasive Systems Division (1998) - Published Product Literature, 3M Center Building 223-6N-01, St. Paul, MN 55144-1000. http://www.vikingtapes.co.uk

AEA Technology plc, (1998). ARTISAN - Published Product Literature - Harwell Laboratory, Remote Handling and Robotics Department, Oxfordshire OX11 0RA, UK. http://www.aeat.co.uk

Addleston, L. and Rice, S. (1991). Performance of Materials in Buildings, Butterworth Heinemann.

Ahuactzin, J.M., Talbi, E-G., Bessiere, P. and Mazer, E. (1991). "Using Genetic Algorithms for Robot Motion Planning", Geometric Reasoning for Perception and Action Workshop, Germany: Springer-Verlag: 84-93.

Angeline, P.J., Saunders, G.M. and Pollack, J.B. (1994). "An Evolutionary Algorithm that Constructs Recurrent Neural Networks", IEEE Transactions on Neural Networks, 5(1): 54-64.

Aqua Blast, (1998). "Aqua Blast - 2500 plus System", Published Product Literature, Aqua Blast Ltd., South Denes Road, Norfolk NR30 3PR, UK. http://aquablast.com

Arabas, J., Michalewicz, Z. and Mulawka, J. (1994). "A Genetic Algorithm with Varying Population Size", Proceedings of the First IEEE Conference on Evolutionary Computation, Florida, USA , 1: 73-78.

Ashiru, I., Czarnecki, C. and Routen, T. (1996). "Characteristics of a Genetic Based Approach to Path Planning for Mobile Robots", Journal of Network and Computer Applications, 19(2): 149-169.

Baba, N. and Kubota, N. (1994). "Collision Avoidance Planning of a Robot Manipulator by Using Genetic Algorithm", The First IEEE Conference on Evolutionary Computation, IEEE, NY, USA, 2:714-719.

Bach, T. (1996). Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms, Oxford University Press, UK.

Bai, Y. and Bernold, L.E. (1996). "Development of a Robotic Bridge Painting System," to be published in the Proceedings of ASCE Speciality Conference "Robotics for Challenging Environments", New York, USA: ASCE.

Baker, J.E. (1985). "Adaptive Selection Methods for Genetic Algorithms", The First International Conference on Genetic Algorithms and Their Applications, Erlbaum:100-111.

Barbosa, H.J.C. (1997). "A Co-evolutionary Genetic Algorithm for a Game Approach to Structural Optimization", Proceedings of the Seventh International Conference on Genetic Algorithms, East Lansing, MI, USA: 545-552.

Bartling, U. and Muehlenbein, H. (1997). "Optimization of Large Scale Parcel Distribution Systems by the Breeder Genetic Algorithm (BGA)", Proceedings of the Seventh International Conference on Genetic Algorithms, East Lansing, MI, USA: 473-480.

Bauer, R.J. (1994). Genetic Algorithms and Investment Strategies, John Wiley & Sons, Inc., NY, USA.

Beeche Systems Corporation (1992). Modular Space Frame Platforms and Mecano Systems, Scotia, N.Y., USA.

Belew, R.K. and Booker, L.B. editors (1991). Proceedings of the Fourth International Conference on Genetic Algorithms, San Mateo, CA, USA: Morgan Kaufmann Publishers.

Bennett, P.J. (1993). Abrasive Air Blast Cleaning, Good Painting Guide, USA: The Steel Structures Painting Council (SSPC).

Bernold, L.E. and Moon, S. (1995). Robotic Bridge Paint Removal, Field Testing and Evaluation of Promising Technologies, Project Report to Federal Highway Administration, North Carolina State University, USA.

Bevan, N.R. and Luk, B.L. (1992). "A Sensor Based Autonomous Climbing Vehicle for Unstructured Environments", Proceedings of the 10th ISARC, Houston, Texas, U.S.A, Elsevier: 205-212.

Birchon, D. (1991). Non-destructive Testing, Oxford University Press.

Bramlette, M.F. and Cusic, R.A. (1990). "A comparative evaluation of search methods applied to parametric design of aircraft". Third International Conference on Genetic Algorithms, Los Altos, CA, USA: Morgan Kaufmann Publishers.

British Construction Steelwork Association, (1968). Proceedings of the Conference on Steel Bridges, BCSA, London, UK.

British Defence Standards (MoD UK) NES 755 (1987). Abrasive Blasting of Steel, UK: Aug.

BSI (British Standards Institution), BS 5400: (1978). Steel Concrete and Composite Bridges; Pt 1, London, UK.

BSI, BS 2830: (1973). Suspended Safety Chairs and Cradles for Use in the Construction Industry, UK: Nov.

BSI, BS 4555: (1970). Specification for High Efficiency Dust Respirators, UK.

BSI, BS 6037: (1990). Permanent Suspended Access System, UK.

BSI, BS 2569: (1965). Spraying of Metal Protective Coatings on Iron and Steel, UK.

BSI, BS EN 28028: (1993). Rubber and Plastic Hose Assemblies for Airless Paint Spraying, UK.

BSI, BS 7079: (1992).Preparation of Steel Substrates before Application of Paints and Related Products, UK.

BSI, BS 5121: (1975). Specification for Rubber Sandblast Hose, UK.

BSI, BS EN 60529: (1992). Specification for Degrees of Protection Provided by Enclosures. (IP code), UK.

British Steel Corporation, (1992). Bridged in Steel Series, BSC, UK.

Brown, C.W. (1995). Design of Steel Bridges for Durability, Steel Construction Institute Technical Rep. No.154:1-4, British Constructional Steelwork Association Ltd., UK.

Brown, D.R. and Kuo-Yen Hwang, K.Y. (1993). "Solving Fixed Configuration Problems with Genetic Search", Research in Engineering Design, Springer-Verlag, 5:80-87.

Buckley, M. (1988). "Multicriteria Evaluation: Measures, Manipulation, and Meaning", Environment and Planning B, 15(1):55-64.

Bullock, G.N., Denham, M.J., Parmee I.C. and Wade J.G. (1995). "Developments in the Use of the Genetic Algorithm in Engineering Design", Design Studies, 16(4): 507-524.

Bungey, J.H. and Millard, S.G. (1993) "Radar Inspection of Structures", Proceedings of The Institution of Civil Engineers - Structures and Buildings, London, UK.

Burstein, T., Shreir, L.L. and Jarman, R.A. (1994). Corrosion, 3rd Ed., Butterworth-Heinemann.

Cagan, J. and Agogino, A.M. (1987). "Innovative Design of Mechanical Structures from First Principles. AI EDAM, 1(3):169-189.

Carroll, D.L. (1996). The General Genetic Algorithm (GA) Driver, E-mail: carroll@uxh.cso.uiuc.edu

CEN (European Committee for Standardisation) PREN Z71, (1989). Respiratory Protective Devices.

CEN PREN 1093-4, (1993). Safety of Machinery - Evaluation of the Emission of Airborne Hazardous Substances.

Chamberlain, D.A. (1992). "A remotely Operated Building Inspection Cell", Automation in Construction, Elsevier Science Publishers B.V., Netherlands, 1(1):71-83.

Chen, I.M. and Burdick, J.W. (1995). "Determining Task Optimal Modular Robot Assembly Configurations", IEEE International Conference on Robotics and Automation: 132-137.

Chen, Ch. and Tseng, Ch.S. (1996). "The Path and Location Planning of Workpieces by Genetic Algorithms", Journal of Intelligent Manufacturing, 7(1): 69-76.

Cignatta, J.V. (1993). "Use of Robotics for Replacing Blasters at Industrial Painting Projects", Proceedings of SSPC International Conference, USA: 74-103.

Collie, A.A. and Billingsley, J. (1990). "Design and Performance of the Portsmouth Climbing Robot", Proceedings of the 7th ISARC, Bristol, U.K: 16-23.

Craig, J.J. (1989). Introduction to Robotics, Mechanics and Control. Addison-Wesley Publishing Company, Menlo Park, Californiia, USA.

Cusack, M. and Fogarty, T. (1993). "Autonomous Learning in Vertical Environments", Proceedings of the 10th ISARC, Houston, Texas, U.S.A, Elsevier: 196-203.

Davidor, Y. (1991). Genetic Algorithms and Robotics: A Heuristic Strategy for Optimisation, Singapore: World Scientific Publication Co.

Davidor, Y. and Goldberg, Y. (1989). "An Evolution Standing on the Design of Redundant Manipulators", Parallel Problem Solving from Nature, 1st Workshop, Germany: Springer-Verlag: 60-69.

Davis, L. ed., (1991) Handbook of Genetic Algorithms. New York, USA: Van Nostrand Reinhold.

Dawson, D., Clough, P. and Silk M. (1990). Ultrasonic Detection and Sizing of Fatigue Cracks, Northern Research and Harwell Laboratories, UK.

Deb, K. and Goldberg, D.E. (1989). "An Investigation of Niche and Species Formation in Genetic Function Optimisation", Proceedings of the 3rd International Conference on Genetic Algorithms, San Mateo, USA: Morgan Kaufmann: 104-109.

De Kleijn B.V. (1994). Blast and Spray Equipment, Published Product Literature, Hogewaard 21 - 6624 KP Heerewaarden, Holland.

Denavit, J. and Hartenberg, R.S. (1995). "A Kinematic Notation for Lower-pair Mechanisms Based on Matrices", Journal of Applied Mechanics: 215-221.

Denby, B. and Schofield, D. (1994). "Open-pit Design and Scheduling by Use of Genetic Algorithms", Mining and Metallurgy, Section A: 21-26.

DIN DEUTSCHES INSTITUT FUR NORMUNG E.V. (DIN ENGLISH) 50315, (1988). Testing the Use and Effectiveness of Metallic Abrasives by Centrifugal Blasting, Germany: October.

Dooner, M. (1983). "Computer Simulation to Aid Robot Selection", Control Engeneering Series 23, IEE, London, UK.

DoT (Department of Transport), (1984). Bridge Inspection Guide, HMSO, London, UK.

DoT, (1985). Code of Practice for Routine Maintenance, D.Tp.,London, UK.

DoT, (1986) Notes for Guidance on the Specification for Highway Works, HMSO, London, UK: August.

DoT, (1988). Specifications for Highway Works, HMSO, London, UK: March.

DoT, (1988). Trunk Road Management and Maintenance Notice TRMM 2/88, Trunk Road and Motorway Structures - Records and Inspection, London, UK: March.

Dym, C.L. (1992). "Representation and Problem-solving: the Foundations of Engineering Design. Environment and Planning B, 19(1):97-105.

Earl, C.F. and Rooney, J. (1983). "Some Kinematic Structures for Robot Manipulator Designs", J. of Mechanisms, Transmissions and Automation in Design: 105:15-22.

EN 292-1-2: (1991). Safety of Machinery.

Eschanauer, H., Koski, J. and Osyczka, A. (editors) (1990). Multicriteria Design Optimization, Springer-Verlag.

Finger, S. and Dixon, J.R. (1989a) "A Review of Research in Mechanical Engineering Design. part I: Descriptive, Prescriptive and Computer-based Models of Design Processes", Research in Engineering Design, 1:51-67.

Finger, S. and Dixon, J.R. (1989b) "A Review of Research in Mechanical Engineering Design. part II: Descriptive, Prescriptive and Computer-based Models of Design Processes", Research in Engineering Design, 1:121-137.

Finger, S. and Rinderle, J.R. (1989). "A Transformational Approach to Mechanical Design using a Bond Graph Grammar", Design Theory and Methodology.

Flow UK - Flow International Corporation, (1998). Published Product Literature, 13 High Trees, Sheffield S17 3GF, UK. http://www.flowcorp.com

Fonseca, C.M. and Fleming, P.J. (1993). "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization", Proceedings of the Fifth International Conference on Genetic Algorithms: Urbana-Champaign, IL, USA: Morgan Kaufmann Publishers: 416-423.

Forde, M.C. (1993). "Impulse Radar Testing of Structures", Proceedings of the Institution of Civil Engineers - Structures and Buildings, London, UK: February.

Forrest, S. (editor) (1993). Proceedings of the Fifth International Conference on Genetic Algorithms, Urbana-Champaign, IL, USA: Morgan Kaufmann Publishers.

French, M. (1998). Conceptual Design for Engineers, 3rd Ed., Springer Verlag.

Garside, A. (1992). "Bridge Access Systems: A48 Neath River Bridge and Briton Ferry Dock Viaduct", The Structural Engineer, UK, 70(11):195-201.

Gates, M. and Scarpa, A. (1990). "Criteria for the Selection of Construction Equipment", ASCE Journal of the Construction Division, USA, 106(2): 207-219.

GEC Alsthom Engineering Systems Ltd., (1998). ASM & Refurbishing Manipulator - Published Product Literature, Cambridge Rd., Whetstone LE8 3LH, UK. http://info.lut.ac.uk/departments/cv/eci/company/gec.html

Gelsey, A., Schwabacher, M. and Smith, D. (1998). "Using Modelling Knowledge to Guide Design Search Space", Artificial Intelligence, Elsevier, The Netherlends, 101:35-62.

Gen, M. and Cheng R. (1997). Genetic Algorithms and Engineering Design, John Wiley & Sons.

Gips, J. and Stiny, G. (1980). “Productions Systems and Grammars”, Environment and Planning B. 7:399-408.

Goldberg, D.E. and Richardson, J. (1987). “Genetic Algorithms with Sharing for Multimodal Function Optimisation”, Proceedings of the 2nd International Conference on Genetic Algorithms, Cambridge, MA, USA: Morgan Kaufmann Publishers: 41-49.

Goldberg, D.E. (1989a). Genetic Algorithms in Search, Optimisation and Machine Learning, Reading, Massachusetts, USA: Addison-Wesley.

Goldberg, D.E. (1989b). “Sizing Populations for Serial and Parallel Genetic Algorithms”, Proceedings of the 3rd International Conference on Genetic Algorithms, San Mateo, CA, USA: Morgan Kaufmann Publishers: 70-79.

Goldberg, D.E., Kord, B. and Deb, K. (1989). “Messy Genetic Algorithms: Motivation, Analysis, and First Results”, Complex Systems, 3(5): 493-530.

Goldberg, D.E. (1990). “A Note on Boltzmann Tournament Selection for Genetic Algorithms and Population-Oriented Simulated Annealing”, Complex Systems, USA, Complex Systems Publications, Inc., 4: 445-460.

Goldberg, D.E. (1994). “Genetic and Revolutionary Algorithms Come of Age”, Communications of the ACM, USA, 37(3):113-5.

Goldberg, D.E. and Deb, K. (1991). “A Comparative Analysis of Selection Schemes Used in Genetic Algorithms”, Foundation of Genetic Algorithms; Rawlins G.J.E., ed., San Mateo, CA, USA, Morgan Kaufmann Publishers: 69-93.

Goldberg, D.E., Deb K., and Clark J.H. (1992). “Genetic Algorithms, Noise and Sizing of Populations”. Complex Systems, USA, Complex Systems Pub.,Inc., 6:333-362.

Golls, J. (1991). An Introduction into NDT, American Society for NDT, USA.

Gregory, N. (1992). Defects - why They Occur, how to Stop Them, TWI Bulletin, June.

Grefenstette, J.J. (1986). “Optimisation of Control Parameters for Genetic Algorithms”, IEEE Transactions on Systems, Man and Cybernetics, 16(1):122-28.

Haas, C. (1996). "Evolution of an Automated Crack Sealer: a Study in Construction Technology Development", Automation in Construction, Elsevier, Amsterdam, The Netherlands, 4(4):293-305.

Handa, M., Hasegawa, Y. and Matsuda, H. (1996). "Development of Interior Finishing Unit Assembly System with Robot: WASCOR IV Research Project Report", Automation in Construction, Elsevier, Amsterdam, The Netherlands, 5(1):31-38.

Haupt, R.L. and Haupt, S.E. (1998). Practical Genetic Algorithms, John Wiley & Sons.

Hartenberg, R.S. and Denavit, J. (1964). Kinematic Synthesis of Linkages. McGraw-Hill, NY, USA.

Holland, J.H. (1992). Adaptation in Natural and Artificial Systems. Re-print Ed., Ann Arbor: University of Michigan Press, USA.

Hubka, V. (1989). Principles of Engineering Design, Springer-Verlag, NY, USA.

IEEE (Institute of Electrical and Electronics Engineers), (1991). Test Standard for Expulsion Fuses in Enclosures.

IEEE C37.48C, (1991). Guide for Application of Expulsion Fuses in Enclosures.

Ipec, (1998). "Alpha 100 - Portable Vacuum, Recycling and Air-drying System", Published Product Literature, Ipec Global Inc., 200 Whitehall Street, Providence, RI 02909, USA. http://www.ipecglobal.com

ISO (International Organisation for Standardisation) 4628-1-6: (1982). Paints and Varnishes.

ISO 8501-1-2: (1988). Preparation of Steel Substrates before Application of Paints and Related Products, (Visual assessment of surface cleanliness).

ISO/IR 8502-1: (1991). Preparation of Steel Substrates before Application of Paints and Related Products, (Tests for the assessment of surface cleanliness).

ISO 8502-2-3: (1992). Preparation of Steel Substrates before Application of Paints and Related Products, (DD 207: 1992).

ISO 8503-1-2: (1988). Preparation of Steel Substrates before Application of Paints and Related Products, (Surface roughness characteristics of blast-cleaned steel substrates).

ISO 8504-1-3: (1992). Preparation of Steel Substrates before Application of Paints and Related Products, (BS 7079: 1993), (Surface preparation methods).

ISO 8601-1: (1988) / BS 7079. Preparation of Steel Substrates before Application of Paints and Related Products.

ISO 9225: (1992). Corrosion of Metals and Alloys. Corrosivity of Atmospheres. Measurement of Pollution.

De Jong, K.A. and Spears, W.M. (1992) “A Formal Analysis of the Role of Multi-point Crossover in Genetic Algorithm”, Annals of Mathematics and Artificial Intelligence Journal, 5(1): 1-26.

De Jong, K.A. and Sarma, J. (1993). “Generation Gaps Revisited”, Foundations of Genetic Algorithms 2, (Whitley, L.D. ed.): San Mateo, CA, USA: Morgan Kaufmann Publishers.

KAJIMA CO-OPERATION, (1986). REPORT 1, Japan.

Kane, T.R. and Levinson, D.A. (1985). Dynamics: Theory and Application. McGraw Hill, New York, USA.

Kaneko, M., Tachi, S., Tanie, K. and Minoru, A. (1987). “Basic Study on Simularity in Walking Machine from a Point of Energy Efficiency”, IEEE Journal of Robotics and Automation, 26(5):19-30.

Kangari, R. and Halpin, D.W. (1989). “Potential Robotics Utilisation in Construction”, J. Constr. Engrg. and Mngmt., ASCE, N.Y., USA, 115(1): 126-143.

Karr, C.L. and Goldberg, D.E. (1990). “Genetic algorithm based design of an Air-injected Hydrocyclone”, Proceedings of Mineral and Metallurgical Processing Symposium, Control ‘90, Salt Lake City, Utah, USA, AIME: 265-272.

Kilne, S.J. (1986). Similitude and Approximation Theory, Springer-Verlag, Berlin, Germany.

Klein Ch.A. and Blaho B.E. (1987). “Dexterity Measures for the Design and Control of Kinematically Redundant Manipulators”. The International Journal of Robotics Research, 6(2): 72-83.

Koza, J.R. (1992). Genetic Programming, MIT Press, Cambridge, MA, USA.

Krishnakumar, K. (1989). “Micro-Genetic Algorithms for Stationary and Non-Stationary Function Optimisation”. SPIE: Intelligent Control and Adaptive Systems, Vol.1196, Philadelphia, PA, USA.

Krottmaier, J. (1993). Optimising Engineering Designs. McGraw-Hill Book Company.

Aarts, E.H, Korst, J. and van Laarhoven,P.J. (1997) "Simulated annealing", Local Search in Combinatorial Optimization, E. H. Aarts and J. K. Lenstra (eds.), John Wiley & Sons.

Larkum, T.J. (1992). Sub-sea Manipulators - A Survey of Current Technology, Technical Software Consultants Ltd. UK.

Lin, H.S., Xiao, J. and Michalewicz, Z. (1994). “Evolutionary Algorithm foe Path Planning in Mobile Robot Environment”, Proceedings of the First IEEE Conference on Evolutionary Computations, Florida, USA, 1:211-216.

Lohmann, R. (1993). "Structure Evolution and Incomplete Induction", Biological Cybernetics, 69:319-326.

Lohmann, R. (1994). "Applications of Structure Evolution", Japan - USA Symposium on Flexible Automation, Kobe, Japan:1293-1300.

Lovejoy, D. (1989). Quality Control of Penetrant Testing, Castrol NDT.

Lovejoy, D. (1991). Penetrant Testing, Chapman & Hill.

Luces, E.F., Moon, S., Rihani, R.A. and Bernold L.E. (1995). "Robotics for Increased Safety in Highway Construction and Maintenance". Transport Research News, National Research Council, USA: 176:12-16.

Maher, M.L. and Kundu, S. (1993). "Adaptive Design Using a Genetic Algorithm", IFIP WG 5.2 Workshop on Formal Design Methods for Computer-Aided Design, Tallinin, Estonia: 211-228.

Mahfoud, S.W. (1995). "A Comparison of Parallel and Sequential Niching Methods", Proceedings of the 5th International Conference on Genetic Algorithms, Pittsburgh, PA, USA: Morgan Kaufmann Publishers: 136-143.

Marchiori, E. (1997). "Combining Constraint Processing and Genetic Algorithms for Constraint Satisfaction Problems", Proceedings of the Seventh International Conference on Genetic Algorithms, East Lansing, MI, USA: 330-337.

Michalewicz, Z. (1996). Genetic Algorithms + Data Structures = Evolution Programs, 3rd Ed., Berlin, Germany: Springer - Verlag.

Mitchell, M. (1996). Introduction to Genetic Algorithms, A Bradford Book, The MIT Press, USA.

Moon, S. and Bernold, L.E. (1996). "A Computer Integrated Control Architecture for Robotic Bridge Paint Removal", Journal of Protective Coatings and Linings, Riverton, NJ, USA: 13(2): 23-32.

Mullins, S. and Rinderle, J.R. (1991). "Grammatical Approaches to Engineering Design, Part 1: An Introduction and Commentary", Research in Engineering Design, 2:121-1355.

Navon, R. (1989). "Determination of Optimal Construction Robot Configuration," Israel Journal of Technology, 25: 71-76.

Navon, R. (1990a). "Weight and Stability Analysis of the Multi Purpose Interior Robot", Proceedings of the 7th ISARC, Bristol, U.K: 47-54.

Navon, R. (1990b). "Cost and Weight Analysis of a Construction Robot at the Conceptual Design Stage", ASME Manufacturing Review, USA, 3(4):238-244.

Navon, R. and Warszawski, A. (1992). Factors Affecting the Productivity of a Construction Robot, International Journal of Construction Information Technology, 1(1): 59,78.

Navon, R. (1995). "Conceptual Design of a Flooring Robot: Development Methodology and Results", Automation in Construction, Elsevier, Netherlands: 4(3): 225-238.

Navon, R. and McCrea, A. (1997). Selection of Optimal Construction Robot Using Genetic Algorithm, Journal of Computing in Civil Engineering - ASCE, New York, USA:11(3), p.1-9.

Nearchou, A.C. (1998). "Solving the Inverse Kinematics Problem of Redundant Robots Operating in Complex Environments via a Modified Genetic Algorithm", Mechanism and Machine Theory, 33(3): 273-292.

Nelco, (1993). Published Product Literature, Nelco Manufacturing Corporation, P.O. Box 36239, Oklahoma City, OK 73136-0239, USA.

Neri, F. and Saitta, L. (1995). "Analysis of Genetic Algorithms under Pure Selection", Proceedings of the 6th International Conference on Genetic Algorithms, Pittsburgh, PA, USA: Morgan Kaufmann Publishers: 32-39.

Nishi, A. and Miyagi, H. (1993). "Propeller Type Wall-climbing Robot for Inspection Use", Proceedings of the 10th ISARC, Houston, Texas, USA, Elsevier: 189-196.

Ohkura, K. and Ueda, K. (1997). "An Extended Framework for Overcoming Premature Convergence", Proceedings of the Seventh International Conference on Genetic Algorithms, East Lansing, MI, USA: 260-267.

Oliver, J.R. (1993). "Discovering Individual Decision Rules: an Application of Genetic Algorithm", Proceedings of the 5th International Conference on Genetic Algorithms, Urbana-Champaign, IL, USA: Morgan Kaufmann Publishers: 165-173.

Pahl, G. and Beitz, W. (1996). Engineering Design: A Systematic Approach, 2nd Ed., Springer-Verlag.

Paredis, Ch.J.J., Brown, H.B. and Khoshla, P.K. (1997). "A Rapidly Deplyable Manipulator System", Robotics and Autonomous Systems, 21(3):289 - 304.

Pham, D.T. and Yang, Y. (1993). "A Genetic Algorithm Based Preliminary Design System", Proceedings of the Mechanical Engineering Institute, Part D (Journal of Automobile Engineering), 207:127-133.

Pritschow, G., Dalacker, M. and Kurz, J. (1996). "Technological Aspects in the Development of a Mobile Bricklaying Robot", Automation in Construction, Elsevier, Amsterdam, The Netherlands, 5(1):3-14.

Pugh, S. (1991). Total Design: Integrated Methods for Successful Product Engineering. Addison - Wesley Publishers Ltd, UK.

Ranky, P.G. and Ho, C.Y. (1985). Robot Modelling: Control and Application with Software, Sringer Verlag.

Rasheed, K. and Hirsh, H.(1997). "Using Case-based Learning to Improve Genetic-Algorithm-Based Design Optimization", Proceedings of the Seventh International Conference on Genetic Algorithms, East Lansing, MI, USA: 513-520.

Rautaruukki, (1998). Published Product Literature, Rautaruukki Engineering Division, Kiilakiventie 1, P.O. Box 217, Fin-90101 Oulu, Finland. http://www.rautaruukki.fi

Rawlins, G.J.E. (1991). (editor), Foundations of Genetic Algorithms and Classifier Systems, San Mateo, CA, USA: Morgan Kaufmann Publishers.

Rinderle, J.R. (1991). "Grammatical Approaches to Engineering Design, Part II: Melding Configuration and Parametric Design Using Attribute Grammars", Research in Engineering Design, 2:137-146.

Robery, P. (1990). "Now You See It - Radar, Vibration and Thermography as Investigating Tools for Structures", Construction, Maintenance and Repair, UK: May-June.

Rockey, K. and Evans, A. (1990). The Design of Steel Bridges, Granada, USA.

Roozenburg, N.F.M and Eekels, J. (1995). Product Design: Fundamentals and Methods, John Wiley & Sons Ltd, UK.

Rosenfeld, Y. and Warszawski, A. (1990). "Robotic Performance of Interior Finishing Works: Development of Full Size Applications", Proceedings of the 7th ISARC, Bristol, U.K: 63-70.

Rosenfeld, Y., Shohet, I.M. and Warszawski, A. (1994). "Optimization Algorithms of Task Planning for Autonomous Construction Robots", The International Journal of Construction Information Technology, University of Salford, UK, 2(3):1-19.

Ross, T. (1977). "Metal Corrosion", New Oxford University Press, UK.

Roston, G.P. (1994). A Genetic Methodology for Configuration Design, Ph.D. Thesis, Carnegie Mellon University.

Sakai, S. and Tokioka, S. (1990). "Improvement of the Painting Robot (FR-2) and its Application on Site", Proceedings of the 7th ISARC, Bristol, U.K: 71-86.

Sandor, G.N and Erdman, A.G. (1984). Advanced Mechanism Design: Analysis and Synthesis, Vol. 2. Prentice-Hall, Inc., Englewood Cliffs, NJ, USA.

Sandroid, (1993). "The Sandroid System", Published Product Literature, Sandroid Systems Inc., 2800 High Ridge Road, Boynton Beach, Florida 33426, USA.

Sarma, J. and De Jong, K. (1997). "An Analysis of Local Selection Algorithms in a Spatially Structured Evolutionary Algorithm", Proceedings of the Seventh International Conference on Genetic Algorithms, East Lansing, MI, USA: 181-186.

Schaffer, J.D. editor (1993). Proceedings of the Third International Conference on Genetic Algorithnis, San Mateo, CA, USA: Morgan Kaufmann Publishers.

Schilling, R.J. (1990). Fundamentals of Robotics, Prentice-Hall International (UK) Ltd., London, UK.

Schilling, (1998). TITAN1-3, Orion, Conan - Published Product Literature, Schilling Development, 1632, Da Vinci Court, Davis, CA 95616, USA. http://www.schilling.com

Schmidt, L.C. and Cagan, J. (1994). "Recursive Annealing: A Computational Model for Machine Design", Research in Engineering Design.

Schoenauer, M. and Michalewicz, Z. (1997). "Boundary Operators for Constrained Parameter Optimization Problems", Proceedings of the Seventh International Conference on Genetic Algorithms, East Lansing, MI, USA: 322-329.

Schwefel, H.P. (1995). Evolution and Optimum Seeking. Wiley & Sons, UK.

Shibata, T., Fukuda, T., Kosuge, K. and Arai, F. (1992). "Selfish and Coordinative Planning for Multiple Mobile Robots by Genetic Algorithm", IEEE Transactions on Robotics and Control: 2686-2691.

Shigley, J.E. (1989). Mechanical Engineering Design. 5th Ed., McGraw Hill.

Shreir, L. (1993). Corrosion, Newnes-Butterworth.

Skibniewski, M. (1988a). "Framework for Decision Making on Implementing Robotics in Construction", ASCE Journal of Computing in Civil Engineering, 2(2): 188-201.

Skibniewski, M. (1988b). Robotics in Civil Engineering, USA: Van Nostrand Reinhold.

Slingsby Engineering Ltd., (1998). ARM 1-3 - Published Product Literature, Ings Lane, Kirkbymoorside, North Yorkshire, YO6 6EZ, UK. http://www.soc.soton.ac.uk

Son, J. and Skibniewski, M. (1995). "Genetic Algorithms in Construction Project Computations", International Journal of Construction Information Technology, Salford, UK: University of Salford Press, 3(2): 1-27.

Spath, D. and Andres, J. (1997). "Concept of a Robot for Interior Building Trades by the Example of Wall Slits in Masonry", Automation in Construction, Elsevier, Amsterdam, The Netherlands, 6(3):205-214.

Spears, W.M. and De Jong, K.A. (1991). "On the Virtues of Parameterized Uniform Crossover", Proceedings of the Fourth International Conference on Genetic Algorithms, San Mateo, CA, USA, Morgan Kaufmann Publishers: 230-236.

Spong, M.W. and Vidyasagar, M. (1989). Robot Dynamics and Control, John Wiley & Sons, Singapore.

SSPC (Steel Structures Painting Council), (1993). Safety and Health in the Protective Coatings Industry, Chapter 26.

Stiny, G. (1980). "Introduction to Shape and Shape Grammars", Environment and Planning B, 7: 343-351.

Strainstall Engineering Services, (1993). Structural Integrity using Acoustic Emission, Strainstall.

Sturges, R.H., O'Shaughnessy, K. and Reed, R.G. (1992). "A Systematic Approach to Conceptual Design Based on Function Logic", International Journal of Concurrent Engineering, 1(2):93-106.

Syswerda, G. (1989). "Uniform Crossover in Genetic Algorithms", Proceedings of the Third International Conference on Genetic Algorithms, Los Altos, CA, USA: Morgan Kaufmann Publishers: 2-9.

Technitube Ltd., (1997). Company Profile and Introduction to Beeche Systems, Thornbury Hill lane, Firbeck nr. Worksop SB1 8JW, UK.

Tensiodyne Corporation, (1993). The Fatigue Fuse Monitoring System, Tensiodyne.

Thierens, D. (1997). "Selection Schemes, Elitist Recombination, and Selection Intensity", Proceedings of the Seventh International Conference on Genetic Algorithms, East Lansing, MI, USA: 152-159.

Thierens, D. and Goldberg, D.E. (1994). "Convergence Models of Genetic Algorithm Selection Schemes", 3rd International Conference on Parallel Problem Solving from Nature, Jerusalem, Israel: 119-129. Lecture Notes in Computer Science, 866 - Springer-Verlag.

Thompson, T., Haas, C., Dailey, C. and Traver, A. (1995). "An Assessment of Automated Surface Finishing Technologies," Proceedings of the Construction Congress, ASCE, San Diego, USA: 76-83.

Titman, T.J. (1989). "Infra-Red Thermal Imaging", Proceedings of IEE College, Digest 84.

Tractel Group, (1991). Access Equipment for Construction, Inspection and Maintenance.

Trelawny, (1994). Published Product Literature, Trelawny - Pneumatic Tools, 93 Great Valley Parkway, Great Valley Corporate Centre, Malvern, PA 19355, USA.

Troitsky, (1990). Prestressed Steel Bridges: Theory and Design, Bridge Series.

Ullman, D.G. (1997). The Mechanical Design Process, 2nd Ed., McGraw-Hill, USA.

Ullman, D.G., Dietterich, T.G. and Stauffer, L.A. (1988). "A Model of the Mechanical Design Process Based on Empirical Data", AI EDAM, 2(1):33-52.

Underwrites Laboratories Inc. (UL), (1993). Subject 2062 - Outline of Investigation for Enclosures for Use in Hazardous (Classified) Locations.

United Technologies, (1994). "The Ship ARMS Depainting System", Published Product Literature, Pratt and Whitney Waterjet Systems Inc., P.O. Box 070019, Huntsville, AL 35807-7019, USA.

Vidal, R (ed.) (1993) "Applied Simulated Annealing", Lecture Notes in Economics and Mathematical Systems no. 396, Springer-Verlag.

Voogd, H. (1988). "Multicriteria Evaluation: Measures, Manipulation, and Meaning - a Reply", Environment and Planning B, 15(1):65-72.

Warszawski, A. (1990). Industrialisation and Robotics in Building, USA: Harper & Row.

Warszawski, A. and Navon, R. (1991). "Robot for Interior Finishing Work", ASCE Journal of Construction Engineering and Management, USA, 117(3):402-422.

White, P.M. and Pettey Ch.C. (1997). "Double Selection in Diffusion Model GAs", Proceedings of the Seventh International Conference on Genetic Algorithms, East Lansing, MI, USA:174-180.

White, T.S., Collie, A.A. and Billingsley, C. (1991). "A Robot Climbing Vehicle and Its Man-Machine Interface for Use in the Nuclear Industry", Proceedings of the 8th ISARC, Stuttgart, Germany.

Whitley, D.L. (1989). "The GENITOR Algorithm and Selection Pressure: Why Rank-based Alocation of Reproductive Trials is Best", The Third International Conferencon Genetic Algorithms, San Mateo, CA, USA, Morgan Kaufmann Publishers:116-121.

Whitley, D.L. editor (1993). Foundations of Genetic Algorithms 2, San Mateo, CA, USA: Morgan Kaufmann Publishers.

Wu, A.S., Lindsay, R.K. and Riolo, R.L. (1997). "Empirical Observations on the Roles of Crossover and Mutation", Proceedings of the Seventh International Conference on Genetic Algorithms, East Lansing, MI, USA: 362-369.

Zhao, M., Ansari, N. and Hou, E. (1994). "Mobile Manipulator Path Planning by a Genetic Algorithm," Journal of Robotic Systems, 11(3):143-153.

Zhao, L., Tsujimura, Y. and Gen, M. (1995). "Genetic Algorithm for Robot Selection and Work Station Assignment Problem", Journal of Computers and Industrial Engineering, 31(3/4): 599-602.

Zummermann, M., Truninger, R. and Schweitzer, G. (1991). "Design of a Sensor Supported Mobile Platform for Rough Terrain", Proceedings of the 8th ISARC, Stuttgart, Germany.