2004

Optimization and visualization of rapid prototyping process parameters.

Ahmed M. El Shenawy
University of Windsor

NOTE TO USERS

Optimization and Visualization of Rapid Prototyping Process Parameters

By

Ahmed M. El Shenawy

A Thesis
Submitted to the Faculty of Graduate Studies and Research
through Industrial and Manufacturing Systems Engineering
in Partial Fulfillment of the Requirements
for
the Degree of Master of Applied Science at the
University of Windsor

Windsor, Ontario, Canada

2004

Your file Votre référence
ISBN: 0-612-97010-8
Our file Notre référence
ISBN: 0-612-97010-8

ABSTRACT

The optimal selection of rapid prototyping (RP) process parameters is a great concern to RP designers. When dealing with this problem, different build objectives have to be taken into consideration. Using virtual rapid prototyping (VRP) systems as a visualization tool to verify the optimally selected process parameters will assist designers in taking critical decisions regarding modeling of prototypes. This will lead to substantial improvements in part accuracy using minimal number of iterations, and no physical fabrication until confident enough to do so. The purpose of this thesis is to demonstrate that virtual validation of optimally selected process parameters can significantly reduce time and effort spent on traditional RP experimentation.

To achieve the goal of this thesis, a multi-objective optimization technique is proposed and a model is generated taking into consideration different build objectives, which are surface roughness, support structure volume, build time and dimensional accuracy. The multi-objective method used is the weighted sum method, where a single utility function has been formulated, which combines all the objective functions together. The orders of magnitudes have been normalized, and finally weights have been assigned for each objective function in order to create the general formulation.

A mixed GA code was then programmed and a toolbox was developed using MATLAB software for selecting near optimal values for the most crucial RP process parameters, namely: layer thickness, build orientation and road width. A case study of a geometric model was built using I-DEAS CAD/CAM package to examine the developed code. The results of the optimal selection of process parameters are then visualized and validated using commercial virtual rapid prototyping software (VisCAM RP).

The proposed research work will provide the RP process designers with a parameter selection tool that is time and cost effective as opposed to the traditional experimentation methods.

ACKNOWLEDGEMENTS

First of all I would like to praise my Creator, my Lord, my Master, Allah the most beneficent the most merciful who blessed me during the years of my research and without His will none of this would have happened.

I would also like to extend sincere gratitude and appreciation to my advisor Professor Waguih H. ElMaraghy for affording me with his great effort and generous help throughout my M.A.Sc. program, which guided me to fulfilling my research work and presenting it in an acceptable form. I would also like to recognize my other committee members Professor Hoda A. ElMaraghy, who shared in building my work by giving me distinguished feedback and valuable suggestions and Professor AbdulFattah Asfour for his kind support, encouragement and review of this work.

Furthermore, I would like to thank all my colleagues in the Intelligent Manufacturing Systems center, especially those who shared with me various research problems and assisted me through some difficulties. Thank you for your cooperation. You know who you are.

Finally, form the bottom of my heart, a special thank you goes to my amazing family for their outstanding moral support, their endless love and care, and continuous encouragement. God bless you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

RP	Rapid Prototyping
LM	Layered Manufacturing
RP&M	Rapid Prototyping & Manufacturing
VP	Virtual Prototyping
VRP	Virtual Rapid Prototyping
SLA	Stereo Lithography Analysis
SLS	Selective Laser Sintering
FDM	Fused Deposition Modeling
LOM	Laminated Object Modeling
3DP	Three Dimensional Printing
BPM	Ballistic Particle Manufacturing
SGC	Solid Ground Curing
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CNC	Computer Numerical Control
GA	Genetic Algorithms
I-DEAS	I-DEAS is a CAD/CAM software package for modeling and analyzing 3D models.
VisCAM RP	VisCAM RP is a commercial virtual rapid prototyping software for preparing rapid prototyping data and visualizing the part in the virtual domain.
STL	Stereo Lithography file format, the standard file format for Rapid Prototyping technologies

IGES	Initial Graphics Exchange Specification, an international standard that defines a neutral file format for representation of geometric data.
STEP	The Standard for the Exchange of Product Model Data, a comprehensive ISO standard that describes how to represent and exchange digital product information.
PVC	Polyvinylchloride
ABS	Acrylonirile butadiene styrene
IVECS	Interactive Virtual Environment for Correction of Stereo lithography

NOMENCLATURE

SE	Stair-Step Error
L	Layer Thickness
A	Face Area
A_{total}	Total Surface Area
Θ	Surface Angle
C	Cusp Height
ACH	Average Cusp Height
$C_{all.}$	Maximum Allowable Cusp Height
N_f	Number of Faces
n	Number of Inclined Faces
VS	Support Structure Volume
$V_{env.}$	Volume of Model Envelope
V_{solid}	Volume of Solid Model
Ap	Projected Area
$H_{ave.}$	Average Height of Overhanging Face Vertices
T	Build Time
Rw	Road Width
AD	Average Absolute Deviation
opt.	Optimal
norm.	Normalized
UF	Utility Function
Or	Orientation

CHAPTER 1

INTRODUCTION

This chapter gives a brief review of the current rapid prototyping practice, the motivations behind the presented research, the objective, the thesis, the approach followed during the research, and the thesis organization.

1.1 Review of Rapid Prototyping

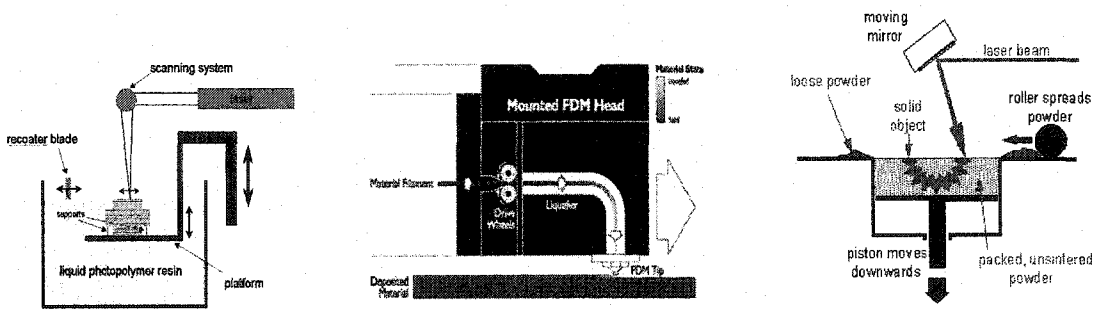
Rapid Prototyping (RP) or Layered Manufacturing (LM) is a technology, emerged in the late 80's, that produces models and prototype parts directly from 3D computer-aided design (CAD) model data. Unlike conventional machining technologies, which are subtractive in nature, RP systems join together liquid, powder, and sheet material to form parts.

Layer by Layer, RP machines fabricate plastic, wood, ceramic and even metal objects using thin horizontal cross sections directly from a computer generated model without any tooling, fixtures or skilled craftsman. This is usually achieved without the need for any, or with the need for very little, machine set-up [Wohlers, 1996].

Product manufacturing industry is facing three important challenging tasks: (1) substantial reduction of product development time; (2) improvement on flexibility for manufacturing small batch size products; and (3) manufacturing products with minimum allowable defects. Computer-aided design and manufacturing (CAD and CAM) have significantly improved the traditional production design and manufacturing. However, there are a number of obstacles in true integration of CAD with CAM for rapid development of new products. Although substantial research has been done in the past for CAD and CAM integration, such as feature recognition, CNC programming and process planning, the gap between CAD and CAM remains unfilled in the following aspects:

(1) Rapid creation of 3D models and prototypes.
(2) Cost-effective production of patterns and moulds with complex surfaces.
(3) High accuracy products with minimal human intervention.

To shorten the time for developing patterns, moulds and prototypes, some manufacturing enterprises have started to use rapid prototyping methods for complex patterns making and component prototyping. Over the past few years, a variety of new rapid manufacturing technologies, generally called *Rapid Prototyping and Manufacturing (RP&M)*, shown in Figure 1.1, have emerged; the technologies developed include *Stereo Lithography Apparatus (SLA), Selective Laser Sintering (SLS), Fused Deposition Modeling (FDM), Laminated Object Manufacturing (LOM), Ballistic Particle Manufacturing (BPM), and Three Dimensional Printing (3D Printing).* These technologies are capable of directly generating physical objects from CAD databases. They have a common important feature; the prototype part is produced by adding materials rather than removing material. This simplifies the 3D part producing processes to 2D layer adding processes such that a part can be produced directly from its computer model [Yan, 1996].



(a) (b) (c)

Figure (1.1) Rapid Prototyping technologies (a) Stereo Lithography Analysis, (b) Fused Deposition Modeling, (c) Selective Laser Sintering

To build a part using rapid prototyping, as will be discussed in detail in chapter 2, process parameters need to be selected and fed to the RP machine. This requires a detailed understanding of the effect of the control parameters on a specific process. The influence of the control parameters vary from one process to another. The selection of the

parameters will affect the functional build objectives of an RP manufacturing process. Optimal RP parameter selection can be done by either using experimentation methods or optimization techniques. Optimization of rapid prototyping process parameters has been a great concern for designers in the early stages of product development, since it will lead to a great deal of time and material consumption, as well as product accuracy and cost efficiency. However, research work has so far focused on the optimization of a single parameter. In general, these techniques lack the flexibility to incorporate multiple requirements or parameters according to the desired quality. Furthermore, they only provide numerical results. Given the geometric complexity of a part, it is often difficult to interpret the numerical data for the corresponding topological changes on the part. Visualization of the part prior to physical fabrication will definitely enhance the designer's understanding of the part. The effect of multiple process parameters on the part quality, along with the visual representation of the final part, can be realized by applying virtual prototyping (VP) to the RP process [Choi, 2001].

Therefore, we could base the development of a product using rapid prototyping technologies on the following stages: (1) identification of the designer's requirements or build objectives; (2) identification of the key process parameters using optimization; and (3) verification of the influence of the chosen parameters on the build objectives using visualization. [Choi, 2002]

1.2 Motivation

The available literature in rapid prototyping parameter optimization covered either single build objectives or multi-objectives for one process parameter. A couple dealt with multi objectives and more than one parameter, but mainly focused on SLA process, raising the need to address multi-objective optimization problem with respect to FDM processes. When selecting mathematical models for the optimization problems, some build objective models developed in previous researches needed enhancement. On the other hand, no research has used visualization for validation of optimization results.

All this raised a need to focus on FDM process parameters optimization. An optimization tool is needed for a multi-objective problem that optimizes the most influential and controllable parameters. Finally, there is a need for a virtual rapid prototyping (VRP) system to validate the optimization results through visualization.

1.3 Objective, Thesis and Approach

The objective of the research reported in this thesis is to generate a multi-objective model and build a tool for selecting near optimal values for the most crucial RP process parameters and utilize VRP to validate the outcomes of this tool. This goal is achieved in this thesis using the following approach:

1. Develop a model for different build objectives as a function of most crucial process control parameters.
2. Develop a multi-objective utility function to evaluate different possibilities of build objectives.
3. Develop a mixed GA code and use it as a global optimization method for selection of process parameters.
4. Build a case study on I-DEAS package in order to test the developed code.
5. Use VRP to visualize and validate outcome results.

The purpose of this thesis is to prove that:

"Visualization and virtual validation of optimally selected process parameters using a virtual rapid prototyping system can be considered a powerful tool that will significantly assist designers with an advantage over traditional RP experimentation."

1.4 Contributions

The reported research makes the following contributions in the fields of RP process parameters optimization:

1. An extensive critical literature review has been prepared. The review highlights the latest work related to rapid prototyping, virtual rapid prototyping, and optimization of rapid prototyping process parameters for several rapid prototyping technologies.

2. The development of new and more indicative objective function models to evaluate the performance of process parameters, namely; average cusp height and support structure volume.

3. The use of multi-objective optimization to handle a combination of the most important build objectives with the most crucial process parameters for the first time.

4. An optimization tool box has been developed on MATLAB, utilizing mixed GA's.

5. The use of VRP to visualize and validate outcomes of optimization process will be considered for the first time.

1.5 Thesis Organization

This thesis is divided into 8 chapters including the following:

- Chapter one includes a brief introductory review of rapid prototyping and manufacturing technology. It also includes the motivation, research objective, thesis and approach.
- Chapter two presents an overview of related topics to rapid prototyping, virtual prototyping, and virtual rapid prototyping.
- Chapter three presents a literature survey covering areas of rapid prototyping, virtual prototyping, virtual rapid prototyping, and optimization of rapid prototyping parameters. The chapter concludes by pointing out research gaps and several key issues directly related to the research topic.
- Chapter four discusses the crucial RP process parameters and their influence on the suggested build objectives, followed by the development of the mathematical models, and the formulation of the utility function for the multi-objective optimization problem.
- Chapter five provides an overview of the Genetic Algorithm (GA) technique and describes its implementation to solve the optimization problem in hand.
- Chapter six presents the results of applying the developed algorithm to a case study that was built using I-DEAS CAD/CAM package.
- Chapter seven demonstrates the outcomes of using the virtual rapid prototyping software for the purpose of visualizing and virtually validating the optimized results.
- Chapter eight concludes the thesis work and highlights the significance of the added research contributions and those that can be expected in the future work.

CHAPTER 2

OVERVIEW OF RAPID PROTOTYPING

This chapter presents an understanding of the rapid prototyping technologies and machines involved in the layered manufacturing industry. It also defines other relevant approaches to rapid prototyping analysis, such as virtual prototyping and virtual rapid prototyping.

2.1 Rapid Prototyping Technologies

Over the last decade over 30 companies developed and marketed rapid prototyping machines based on different physical principles and implementation concepts as seen in Table 2.1. All have in common that the components are generated layer by layer also known as "Material Increase Manufacturing". In general they use the same virtual database, i.e. a volume 3D CAD model in one of the commonly used data formats (STL, DXF, IGES, STEP, etc.) [Levy 2003].

Table 2.1: Rapid Prototyping technologies, acronyms and development years [Levy 2003]

Name	Acronym	Development Years
Stereo Lithography	SLA	1986 - 1988
Solid Ground Curing	SGC	1986 - 1988 (disappeared 1999)
Laminated Object Manufacturing	LOM	1985 - 1991
Fused Deposition Modeling	FDM	1988 - 1991
Selective Laser Sintering	SLS	1987 - 1992
3D Printing	3DP	1985 - 1997

The major differences among these technologies are in two aspects: (1) materials used; and (2) part building techniques. The following sections will explain in detail some of these rapid prototyping technologies with respect to the above two aspects.

2.1.1 Stereo Lithography Analysis (SLA)

SLA was invented by Charle Hull of 3D Systems Inc. It is the first commercially available rapid prototyper and is considered as the most widely used prototyping machine. The material used is liquid photo-curable resin, acrylate. Under the initiation of photons, small molecules (monomers) are polymerized into large molecules. Based on this principle, the part is built in a vat of liquid resin as shown in Figure 2.1.

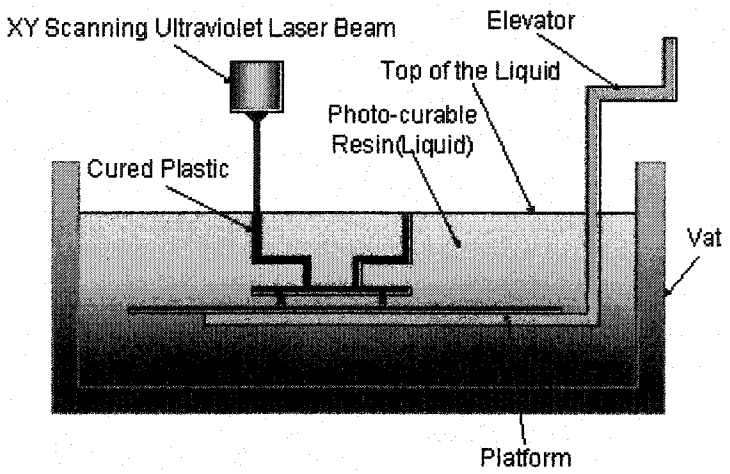


Figure (2.1) The working principle of SLA

The SLA machine creates the prototype by tracing layer cross-section on the surface of the liquid photopolymer pool with a laser beam. Unlike the contouring or the zigzag cutter movement used in CNC machining, the beam traces in parallel lines, or vectorizing first in one direction and then in the orthogonal direction. An elevator table in the resin vat rests just below the liquid surface whose depth is the light absorption limit. The laser beam is deflected horizontally in X and Y axes by galvanometer-driven mirrors so that it moves across the surface of the resin to produce a solid pattern. After a layer is built, the elevator drops a user specified distance and a new coating of liquid resin covers the solidified layer. A wiper helps spread the viscous polymer over for building the next layer. The laser draws a new layer on the top of the previous one. In this way, the layer is built layer by layer from bottom to top. When all layers are completed, the prototype is about 95% cured. Post-curing is needed to completely solidify the prototype. This is done

in a fluorescent oven were ultraviolet light floods the object (prototype). There are several features worthy of mention of SLA.

Material. There are five commercially available photopolymers. All of them are a kind of acrylate.

Support. Because a model is created in liquid, the overhanging regions of the part (unsupported below) sag or float away during the building process. The prototype thus needs some pre-designed support until it is cured or solidified. The support can be pillars, bridges and trusses. Sometimes posts or internal honeycomb sections are needed to add rigidity to tall thin-walled shapes during the process. These additional features are built on the model parts and have to be trimmed after the model is completed [Yan, 1995].

2.1.2 Selective Laser Sintering (SLS)

DTM Corp. (Austin, TX) offers an alternative to liquid-curing systems with its SLS systems which were developed by Carl Deckard and Joseph Beaman at the Mechanical Engineering Department of the University of Texas at Austin. SLS uses carbon dioxide laser to sinter successive layers of powder instead of liquid. In SLS processes, a thin layer of powder is applied by a counter-rotating roller mechanism onto the work place. The powder material is preheated to a temperature slightly below its melting point. The laser beam traces the cross-section on the powder surface to heat up the powder to the sintering temperature so that the powder scanned by the laser is bonded. The powder that is not scanned by the laser will remain on place to serve as a support to the next layer of powder, which aids in reducing distortion. When a layer of the cross-section is completed, the roller levels another layer of powder over the sintered one for the next pass. Figure 2.2 shows the working principle of SLS. SLS has several features.

Material. SLS uses a wide range of materials for model production including polycarbonate, PVC (polyvinylchloride), ABS (acrylonirile butadiene styrene), nylon, resin, polyester, polypropane, polyurethane and investment casting wax. The machine that is capable of using metal and ceramic powder is in the process of development.

Support. The SLS systems usually do not need pre-designed support structures. The un-fused powder on every layer acts as a support during the building process [Yan, 1995].

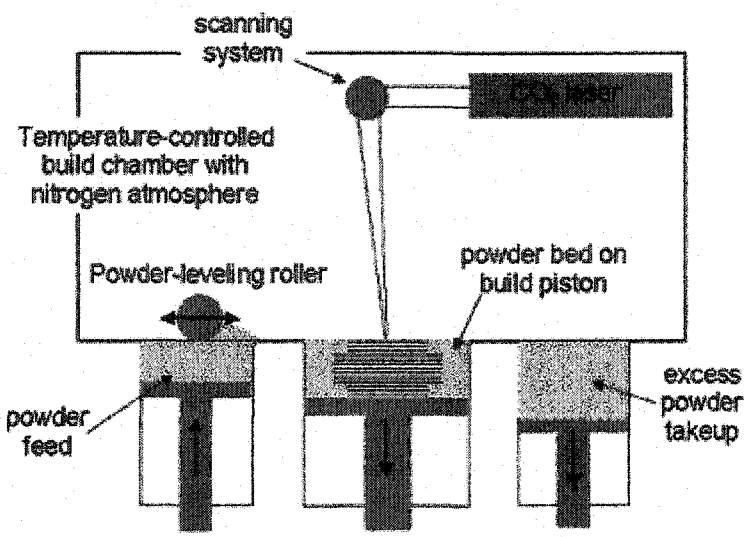


Figure (2.2) The working principle of SLS

2.1.3 Fused Deposition Modeling (FDM)

Rapid prototyping system – 3D modeler developed by Stratasys Inc. – constructs parts based on deposition of extruded thermoplastic materials called FDM process. In an FDM process, a spool of thermoplastic filament feeds into a heated FDM extrusion head, consisting of a heater and a nozzle at the end. The movement of the FDM head is controlled by computer. Inside the flying extrusion head, the filament is melted into liquid (1° above the melting temperature) by a resistant heater. The head traces an exact outline of each cross-section layer of the part on a table (also known as build platform) that moves in the z direction. As the head moves horizontally in the x and y axes, over the table, the thermoplastic material is extruded out the nozzle by a precision pump. The material solidifies in 1/10 seconds as it is directed on to the workplace. After one layer is finished, the table moves down a programmed distance in the z direction for the building of the next layer. Each layer is bonded to the previous layer through thermal heating. The entire system is contained within a chamber which is held at a temperature just below the melting point of the plastic. Figure 2.3 illustrates the working principle of FDM. The FDM has the following features:

Material. The FDM technology allows a variety of modeling materials and colors for model building. Available materials are wax-filled plastic adhesive material, proprietary nylon, ABS, investment casting, wax polycarbonate, polyphenylsulfone, elastomer, and polyester. All the materials are non-toxic and can be in different colors.

Support. In some cases, the FDM process does not need support to produce part. The FDM extrusion head forms a precision horizontal support in mid-air as it solidifies. For overhanging parts, a support is still required to reduce part distortion. Support structures must be designed and fabricated for the overhanging geometries and are later removed in secondary operations [Yan, 1995].

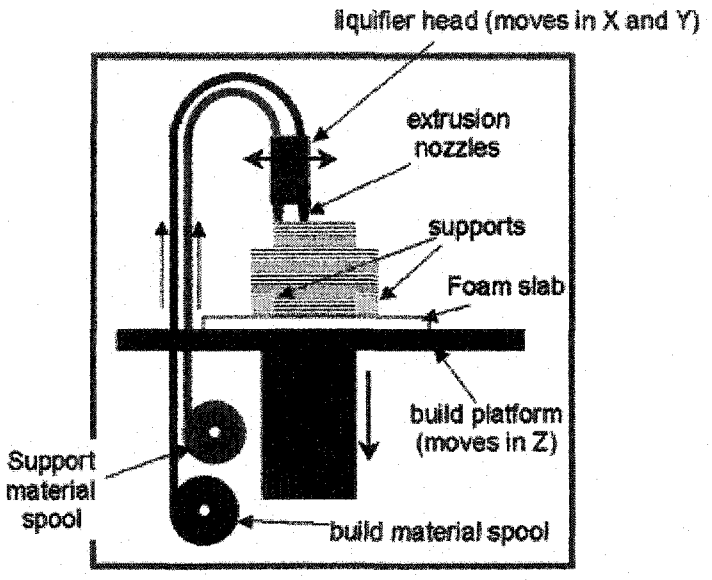


Figure (2.3) The working principle of FDM

2.1.4 Laminated Object Manufacturing (LOM)

The LOM processes produce parts from bonded paper, plastic metal or composite sheet stock. LOM machines bond a layer of sheet material to a stack of previously formed laminations, and then a laser beam follows the contour of a part of a cross-section generated by CAD to cut it to the required shape. The layers can be glued or welded together. The excess material of every sheet is either removed by vacuum suction or

remains as the next layer's support. Figure 2.4 shows the working principle of LOM. The features of LOM are as follows:

Material. Virtually any foil (sheet material) can be applied; paper, metals, plastics, fibers, synthetic materials, glass or composites. Helisys Inc. uses cellulose foils.

Support. The LOM process uses solid-state materials and therefore usually does not need pre-designed support structure [Yan, 1995].

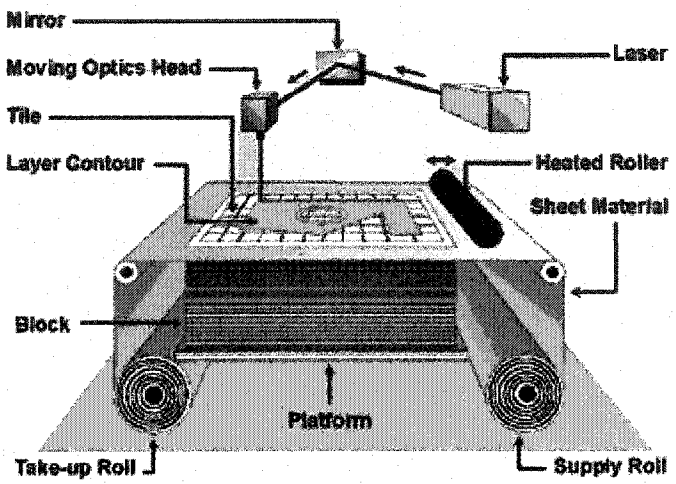


Figure (2.4) The working principle of LOM

2.1.5 Three-Dimensional Printing (3DP)

Three dimensional printing was developed at Massachusetts Institute of Technology (MIT). In the 3D Printing process, a 3D model is sliced into 2D cross-section layers in computer. A layer of powder is spread on the top of the piston, the powder bed, in a cylinder, and then an inkjet printing head projects droplets of binder material onto the powder at the place where the solidification is required according to the information from the computer model. After one layer is completed, the piston drops a predefined distance and a new layer of powder is spread out and selectively glued. When the whole part is completed, heat treatment is required to enhance the bonding of the glued powder, and then the un-bonded powder is removed. Figure 2.5 shows the working process of 3DP. Features of 3DP are summarized below:

Material. The 3DP process can use aluminum-oxide and alumina-silica ceramic powders. The binder material is amorphous or colloidal silicon carbide.

Support. With the 3D Printing technique, the design of support structure for the part is not needed, since the un-bonded powder of each layer remains to form a natural support during the layering process. [Yan, 1995].

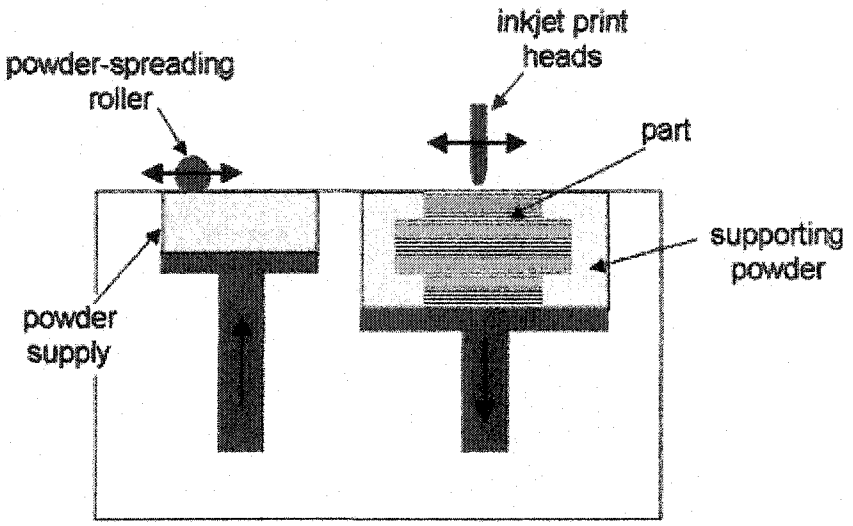


Figure (2.5) The working principle of 3DP

2.2 How Rapid Prototyping Works

Currently, there is no fundamental difference for the data preparation among the existing RP & M technologies. A product is first designed with a 3D modeler. Surfaces of the product are then approximated to STL format. In the approximation, the precise representations of surfaces such as spline surfaces or boundaries of CSG primitive solids are tessellated into the facet format. Most CAD solid modeling software products today can output a stereo lithography (STL) file generator, the *de facto* standard input format used by RP systems in the representation of the solid 3D CAD models. An STL file approximates the shape of a solid model using small triangles called facets. Figure 2.6 shows the tessellation of a sphere. The accuracy of a non-planar surface depends on the number of facets used to approximate the surface. The smaller the facet size the better the surface approximation, but at the expense of the file size and processing speed. If you would open and view the contents of an STL file, you would see a list of X and Z coordinate triplets that describe a surface mesh of triangular facets.

The job of the CAD modeler is completed once it has exported a valid STL file. At that point, the RP system software takes over. Using special slicing software, RP systems cuts a series of thin, parallel, horizontal cross sections through the STL file. If you want to build a part using 0.2mm thick layers for example, you would set the software to slice the model at this increment. Again, the smaller the thickness layer the better the surface finish, but at the expense of the processing speed. The RP system control uses the stack of digital cross sections to produce each layer of material, one on top of the next [Yan, 1995], [Wohlers, 1996].

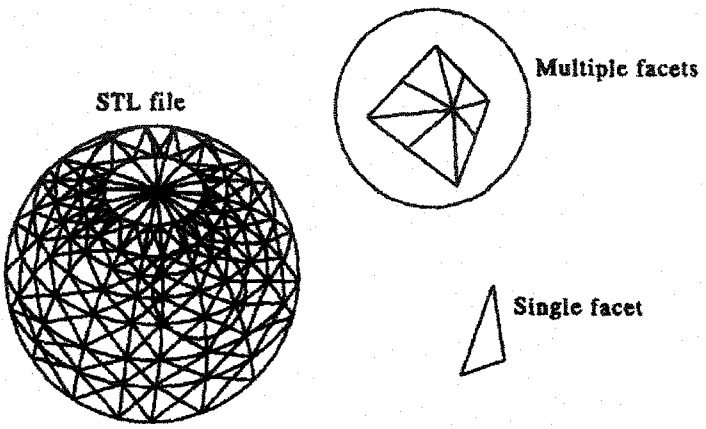


Figure (2.6) The tessellation of a sphere

2.3 Benefits of Rapid Prototyping

A seemingly never ending need to reduce product development time has created a demand for fast approaches to prototyping. This, coupled with a growth of computers in design offices, has motivated inventors to create new ways of producing physical objects from computer model data. Countless entrepreneurial companies, researchers and investors have developed RP, an industry that today is approaching over half a billion dollars. What's more, RP has helped scores of manufacturing companies shorten their product development time, discover design flaws and improve product quality [Wohlers, 1996].

In general, the main benefit of rapid prototyping and manufacturing is the saving in time and cost on tooling and re-engineering therefore reducing the Time-to-Market.

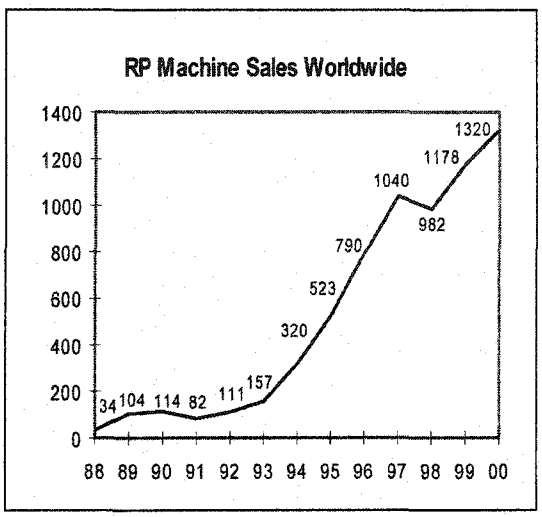
2.4 Current application areas of RP

RP has streamlined the production of engine parts, cellular phones, jewelries, toys, hip joints, architectural models, skeletal replicas, and even art mathematical models. The kind of parts and assemblies achievable with RP is impressive to say the least. With RP's layer-building approach, RP machines can produce virtually any shape. Moreover, they can produce complex parts just as easily as simple ones. Companies often reserve RP for the really tough jobs and use traditional processes for simple shapes [Wohlers, 1996].

Industry surveys indicate that the automobile and aerospace industries make up a significant portion of the worldwide RP customer base. Other major users of RP are producers of industrial equipments, electronic devices, computers, business machines, medical devices, and consumer products. Promising new developments are occurring in the field of medicine as well. Figure 2.7a shows the increase in RP&M technology sales.

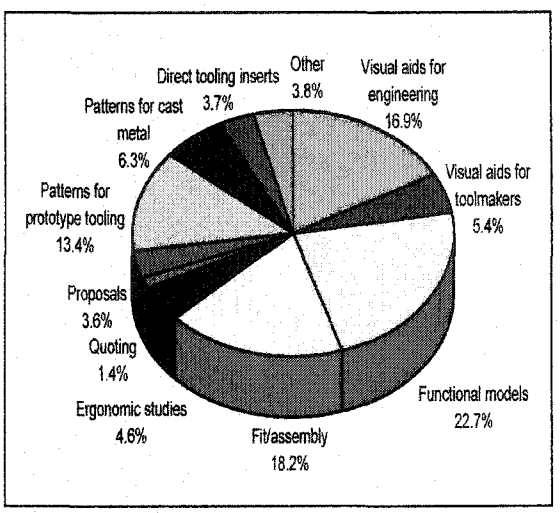
Although RP&M technologies are still fairly at their early stage, a number of industrial companies such as Texas Instruments Inc., Chrysler Corporation, Amp Inc., and Ford Motor Co., have benefited from applying the technologies to improve their product development, specifically in design engineering, manufacturing, and marketing [Yan, 1996].

A survey was conducted by Wholers and Associates and it was found that around 23.4% of RP parts are used as vital aids, whereas 27.5% of them are used as master patterns for secondary manufacturing processes and for direct tooling. Industries use 15.6% of them for fit and assembly tests, 16.1% for functional tests and the rest for quoting, proposal, ergonomic, etc. as shown in Figure 2.7b.



Source: Wohlers Associates, Inc.

(a)



Source: Wohlers Associates, Inc.

(b)

Figure (2.7) (a) RP machine sales (b) RP application areas

2.6 Physical vs. Virtual Prototyping

Physical prototyping is referred to as the production of a physical model from real material such as wood, clay, foam, metal, or any other used to make physical models, although they do not necessarily possess the same properties as those of the finished final product. These prototypes can be classified into three main groups according to the possible nature of physical change used to create them:

(1) Traditional Prototypes – material removal.
(2) Rapid Prototypes (RP) – material addition.
(3) Hybrid Prototypes – both material removal and addition [Zorriassatine, 2003].

Virtual Prototyping (VP) on the other hand refers to the creation of a model in the computer, often referred to as CAD/CAM/CAE. Virtual or computational prototyping is generally understood to be the construction models of products for the purpose of realistic graphical simulation. It provides the ability to test final part behavior in a simulated context without the need to manufacture the physical part first [Chua, 1999].

Virtual prototyping is also known as the subsequent manipulation of a solid CAD model as a substitute for a physical model for the purpose of simulation and analysis, and is not inclusive of the construction of the 3D solid model. VP includes the following functions:

(1) Finite Element Analysis
(2) Mechanical Form, Fit and Function Tests
(3) Interference Checks
(4) Mechanical simulation
(5) Virtual Reality Application
(6) Cosmetic Modeling
(7) Assemblability

The relationship between RP and VP are shown in Figure (2.8)

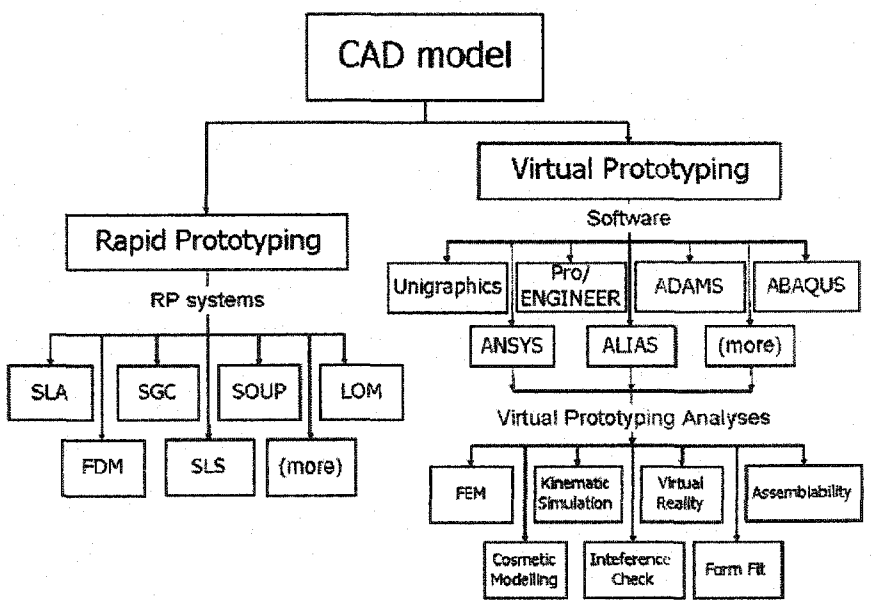


Figure (2.8) Classification of RP and VP

Repeated, efficient and extensive use of prototypes is a vital activity that can make the difference between successful and unsuccessful entry of new products into the competitive world market. In this respect, physical prototyping can prove to be very lengthy and expensive, especially if modifications resulting from design reviews involve tool redesign. The availability and affordability of advanced computer technology has

paved the way for increasing utilization of prototypes that are digital and created in computer based environments, i.e. they are virtual as opposed to being physical [Zorriassatine, 2003].

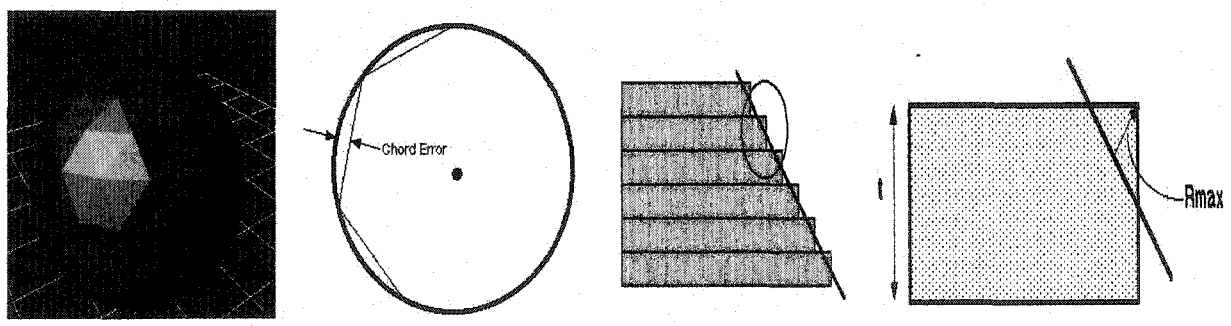
2.7 Virtual Rapid Prototyping

A novel system has been recently developed for the simulation of rapid prototyping fabrication processes. It enables a designer to visualize and subsequently optimize an RP process with a set of process parameters. This system is called *Visualization of Rapid Prototyping* or *Virtual Rapid Prototyping* (VRP). It is simply the integration of Rapid Prototyping (RP) with Virtual Prototyping (VP) to give (VRP).

Visualization in general, is a method of extracting meaningful information from complex data sets by the use of interactive graphics and imaging. It provides processes for seeing and steering the unseen, thereby enriching existing scientific methods [Jee, 2000]. Extended work and research have been done on both physical and virtual prototyping. Ideally, once the extensive simulations of virtual prototypes are over, it must be possible to build the final product right first time and with no safety risks or product failures. In spite of its great achievements and widespread use, it seems that the existing limitations of VP technology have not permitted the full realization of the above ideal [Zorriassatine, 2003]. Virtual testing of prototypes is still evolving, while physical testing is proven and reliable, but both are now relatively well established especially in the automotive and aerospace industries.

Clearly both physical rapid prototyping and virtual prototyping deal with the design, analysis, simulation and testing of the final product(s). Evidently, both of these techniques give out errors in the final products, which are well known in the rapid prototyping world. These errors include:

(1) Tessellation errors; shown in Figure 2.9a
(2) Stair-Case effect; shown in Figure 2.9b
(3) Shrinkage of built objects
(4) Overhanging features
(5) Missing features



(a) (b)

Figure (2.9) (a) Tessellation error (b) Stair-stepping effect

Visualization of the part prior to physical fabrication will definitely enhance the designer's understanding of the part. The effect of multiple process parameters on the part quality, along with the visual representation of the final part, can be realized by applying VP to the RP process. Virtual Rapid Prototyping, as defined above, is the system for simulating the rapid prototyping process, enabling the designer to visualize and optimize the fabrication process with a set of process parameters. The visualization of a virtually simulated part prior to physical fabrication helps reduce unwanted prototyping iterations. As the design cost might increase in direct proportion to the number of iterations for model fabrication, developing an intermediate geometric model of the design and then carefully inspecting the simulated model before fabricating the physical model might help avoid unnecessary part fabrication for the design verification purpose [Choi, 2001]. The integration of VP and RP allows a designer to analyze and visualize the influence of process parameters on the part quality. Therefore, it enhances the RP process by enabling the designer to visualize the part before building it. The system simulates an RP with actual physical phenomena. This provides the designer with a tool to visualize the unseen fabrication capabilities of an RP machine.

VRP systems can facilitate design validation in the early stage of product development. The designer can have a clear representation of the product to examine its aesthetic and structural features. If any problems are identified, the design can be promptly improved before it goes too far down the development cycle. This is particularly important to help enhance the competitiveness of the manufacturing industry,

which is faced with increasing pressure to satisfy demands for small-batch production of different varieties of customized products. In such situations, it would not be economical to make a mould for small-batch production. On the other hand, rapid prototyping may be a convenient tool for direct production of customized products, provided it can fabricate prototypes of the required accuracy and of appropriate materials [Choi, 2003].

Indeed, by providing realistic visualization as well as numeric quantification of the simulation results, the designer can effectively explore the potential problems of the product design and the prototypes that the RP machines will subsequently fabricate. The process parameters can therefore be optimized before physical fabrication.

CHAPTER 3

LITERATURE REVIEW

This chapter presents a literature survey covering areas of rapid prototyping, virtual prototyping, virtual rapid prototyping, and optimization of rapid prototyping parameters. The chapter concludes by pointing out research gaps and several key issues directly related to the research topic.

3.1 Physical & Virtual Rapid Prototyping

Yan [1996] reviews the main technologies and applications of rapid prototyping and presents the principles and features of those rapid prototyping technologies. Chua *et al.* [1999] investigated a comparative study of rapid prototyping versus virtual prototyping technologies with respect to their relevance in product design and manufacturing to study the suitability and effectiveness of both technologies in various aspects of prototyping. Furthermore, Zorriasatine [2003] performed a thorough survey of virtual prototyping techniques providing a broad picture of the field of virtual prototyping and identifying issues and information relevant to the deployment and implementation of VP technology.

Realizing the advantages of VP, researchers combined it with RP at various stages. Gibson [1993] investigated the contributions of Virtual Reality (VR) and RP towards a more efficient product development in ergonomic, aesthetic and functional aspects of design. Fadel *et al.* [1995] linked virtual prototyping with the rapid prototyping process to visualize the support structures of a part and to aid the user to identify improper support structures and to enhance the designer's understanding of manufacturing issues.

Morvan and Fadel [1996] further coupled RP with VR by developing the Interactive Virtual Environment for Correction of Stereo lithography Tessellated List files (IVECS) system. This system detects errors in a STL file and allows the designer to fix them by laying his/her hands on the STL model. Indeed, correcting a faulty STL file interactively

is tedious and daunting job. Tata and Fadel [1998] presented an adaptive slicing algoritm that can vary the layer thickness in relation to local geometry. The algorithm is interfaced with adaptive laminated machining and the stereolithography process through a CNC post processor and a hatching algorithm respectively. A comparison of the estimated surface quality and build time indicates that adaptive slicing produces superior parts in a shorter build time. Lin *et al.* [2001] used a non-linear optimization to predict the rapid prototyping layered process error and developed an optimization algorithm to define the fabricating orientation based on minimum process error for RP fabrication. Jee and Sachs [1998] developed a visual simulation system for 3D printing. However, their system was aimed at developing a visual tool to examine surface textures only. It is a voxel-based approach, which is only suitable for simple objects. Voxels represent geometric detail in small cubes, and hence suffer from large storage requirements.

Choi and Samavedam [2001] developed a virtual prototyping system for simulation of rapid prototyping processes, enabling designers to visualize and optimize an RP process with a set of parameters. The system focuses on further integration of VR and RP to provide a test-bed for selection of optimal process parameters. Later in [2002], they proposed a Virtual Reality (VR) system for modeling and optimizing rapid prototyping processes. The system aims to reduce the manufacturing risks of prototypes early in the product development cycle, and hence, reduce the number of costly design-build-test cycles. It involves modeling and simulating RP in a virtual system, which facilitates visualization and testing the effects of process parameters on the part quality. Furthermore Choi and Chan [2003] proposed a layer based VP system, which builds virtual or digital prototypes to facilitate product development. The approach resembles the physical fabrication process of laminated sheet based RP systems. It simulates such an RP process to create a virtual prototype of a product design. Thus, the designer can perform design validation and accuracy analysis easily in a virtual environment as if using a physical prototype.

From the above literature review a preliminary review matrix shown in Figure 3.1 was developed to clarify the gaps missing in this field of research. It is apparent that the

	RP technology		RP machine involved			Control Parameters				Remarks
	Physical RP	Virtual RP	Stereo Lithography	Fused Deposition Modeling	Other	Layer Thickness	Build Direction	Hatch Space	Part Orientation	
Wohlers [1996]	√	O	x	x	x	O	O	O	O	Genaral guide to rapid prototyping
Yan, [1996]	√	O	x	x	x	O	O	O	O	Survey on RP technologies
Chua [1999]	√	√	x	x	x	O	O	O	O	RP vs. VP comparison
Zorriassatine [2003]	x	√	x	x	x	O	O	O	O	Survey on VP technologies and functions
Jee [2000]	√	√	O	O	3DP	√	√	O	O	Examined surface textures only
Choi [2001]	x	√	√	O	SLS	√	O	√	O	Visualization of RP processes
Choi [2002a]	x	√	√	O	SLS	√	O	√	O	Modeling and optimization of RP
Choi [2003]	x	√	O	O	LOM	√	√	O	O	VP system for product development
Fadel [1996]	√	O	√	O	O	O	O	O	O	Software to bridgethe CAD to RP gap
Tata [1998]	√	O	√	O	ALM	√	O	√	O	Efficient slicing for RP layered manufacturing
Lin [2001]	√	O	x	x	LOM	x	x	O	√	Optimization with minimum RP process error
Gibson [1993]	√	√	x	x	x	O	O	O	O	Contributions of VR and RP

Direct connection	√
Indirect connection	x
No connection	O

Figure (3.1) Preliminary literature review matrix

area of virtual prototyping in integration with rapid prototyping is a new topic of research, and not many researchers have covered it extensively. Furthermore, there is limited work in the application of virtual rapid prototyping systems to FDM processes. The available research work done in this area considered the selection of just a few if not only one optimal control parameter at a time, which again was another missing feature that needed to be covered.

The next section reviews the current state of the art regarding the optimization of rapid prototyping parameters.

3.3 Optimization of Rapid Prototyping Parameters

Many studies have been implemented on optimal selection of RP process parameters. Different problems were examined along with different RP systems. Some of the studies included the study of the build time estimation problem, as some might argue is the most critical or at least a very important factor.

The three main phases included in the build time of a part on an RP system are:

1) Pre-build or data preparation phase, where several prefabrication tasks such as support generation and slicing are performed.
2) Build or fabrication phase, during which the actual fabrication or building of a part is carried out.
3) Post-processing or finishing phase, by then cleaning and finishing of the part take place.

Usually the data preparation time is small compared to the duration of fabrication and post processing time. The post-processing time is related to the part geometry and the post-processing equipment used. But again, the post-processing time of a part is usually small. Among these three phases, the build or fabrication time is usually the most time consuming and costly.

Giannatsis, *et al.* [2001] examined the problem of build time for stereo lithography systems. The study was mainly focused on the build time itself and also analyzed the factors affecting it using experimental investigations. Results indicated that hatching space depend not only on the hatching distance and speed, as originally assumed, but also on the number of hatching vectors employed.

Build time was also tackled in other studies when considering other optimization problems such as selection of optimal orientation also known as preferred build up direction. Orientation is a function of the number of layers needed to build a part, and this also depends on the layer thickness predefined by the user. Theoretically, the larger the layer thickness the smaller the number of layers, and therefore the shorter time it would

take the machine to process the part. The conflict on the other hand is that the larger the layer thickness the greater the stair-stepping effect, which directly affects the surface roughness or accuracy of the modeled part.

Lan *et al.* [1996] investigated the effects of surface quality and build time factors to illustrate the determination of the orientation of a designed part to be fabricated on SLA.

Han *et al.* [2001] addressed in their research the methodology to find the optimal build layout, by considering an orientation and packing of multiple parts in SLS processing. They approached their optimization problem by employing genetic algorithms and were then demonstrated in real prototypes for processing with SLS, which from their results and conclusion, illustrates a good enabling optimization building system to the real industries.

The build time models that the different researchers used for their optimization problems with respect to the most common RP technologies are summarized in Table 3.1.

Table 3.1: Comparison table for build time models of different RP technologies

SLA	SLS	FDM	Others
$BT = 117(V)(Z) - 203(H)(Z) - 5135(H)(V) - 33813(L)(Z) + 1125(L)(V) - 1910000(L)(H) + 95(Z)^2 + 42(V)^2 + 693238(H)^2 9470000(L)^2 + 33(Z) - 81(V) - 4479(H) - 70138(L) + 301$		$BT = 6320 - 2005L - 2299R_w + 454L^2$	$T_d(n) \propto N$ $T_m(n) \propto V$ [Hybrid]
$T_f = n.T_w + (V/L).t_f$	$T_f = n.T_w + (V/L).t_f$	$T_f = n.T_w + (V/L).t_f$	$T_f = n.T_w + (V/L).t_f$
$BT = \sum TLayer_{(i)}$	$BT = \sum TLayer_{(i)}$	$T_{total} = \sum T_i$	$BT = \sum TLayer_{(i)}$

Hu *et al.* [2002] also presented an algorithm to determine the build orientation, but this time for hybrid rapid prototyping.

Orientation not only affects the build time but also the surface roughness of a part. At different orientations some surfaces would tend to change their angle of inclination. Doing so will also affect the stair-stepping effect and again affecting the surface finish of a prototyped model. A lot of researches have analyzed surface roughness in layered forming processes.

Perez *et al.* [2001] characterized effective roughness by carrying out a study of the roughness average obtained through use of these manufacturing processes. Prototyped parts were manufactured using SLA technique to compare the theoretical models proposed with experimental values. An experimental analysis was also carried out of the resulting surface roughness. They concluded that when manufacturing with constant layer thickness, which is the usual situation, it was shown that roughness was not constant and that it can be characterized, in the case of stereo lithography, by means of their proposed models.

Campbell *et al.* [2002] developed a surface roughness visualization algorithm and implemented it with a CAD package. The surface roughness values were obtained through an extensive empirical investigation of several RP techniques, showing how the values will vary across a full range of surface angles. It has been demonstrated that for the majority of the systems there is at least a range of angles in which the surface roughness can be reasonably well predicted. Using the algorithm gives the user the ability to predict the surface roughness of an RP model before it has been built. Areas of unaccepted surface roughness can be identified and alternative build orientations can be investigated in an attempt to eliminate them.

Zhou *et al.* [2002] conducted a scientific and experimental study improving RP part accuracy through parameter tuning and optimization of SLA manufacturing processes. In terms of Taguchi experimental design techniques, an orthogonal array of experiments has

been developed which has the least number of experimental runs and desired process parameter settings. Using a 3D coordinate measuring machine, as series of measurements in evaluating the SLA parts quality has been conducted to find the functional relationships between output part quality and input manufacturing process parameters. The optimal setups of SLA manufacturing parameters for both individual features and a general part with various features have been conducted from this study.

Reeves and Cobb [1997] established a mathematical representation of the surface roughness of stereo lithography parts. The intention of their research was to use this modeling technique as a design tool for defining optimum build orientation and planning post-process finishing operations.

The surface roughness models that the different researchers used for their optimization problems with respect to the most common RP technologies are summarized in Table 3.2.

Table 3.2: Comparison table for surface roughness models of different RP technologies

SLA	SLS	FDM	Others
$SV = \sum (A_i.L \cos\Theta_i)/2$	$SV = \sum (A_i.L \cos\Theta_i)/2$	$SE = \sum (L/2)(A_i)(\cos\Theta_i)$	$SV = \sum (A_i.L \cos\Theta_i)/2$
$R_a = L \sin(\Theta/4)*\tan\Theta$	$R_a = L \sin(\Theta/4)*\tan\Theta$	$R_a = L \sin(\Theta/4)*\tan\Theta$	$R_a = L \sin(\Theta/4)*\tan\Theta$
Cusp Area $= C^2 / (2\sin\Theta \cos\Theta)$	Cusp Area $= C^2 / (2\sin\Theta \cos\Theta)$	$SV = \sum (A_i.L \cos\Theta_i)/2$	Cusp Area $= C^2 / (2\sin\Theta \cos\Theta)$
$R_{a(up)} = (L(\tan\varnothing \sin\Theta + \cos\Theta))/4 + K$ $R_{a(down)} = (L(\tan\varnothing_1 \sin\Theta_1 + \cos\Theta_1))/4 + K_1$		Cusp Area $= C^2 / (2\sin\Theta \cos\Theta)$	
$SA = -12.6E6(H)(L) + 11125(Z)(L) + 48344(Z)(H) + 96.2E6(L)^2 + 3470000(H)^2 577(Z)^2 - 703617(L) - 41629(H) - 321(Z) + 26655$			

Lin *et al.* [2001] developed a mathematical model to describe and analyze layered process error and developed an optimization algorithm to select the fabrication orientation with minimum processing error for layered manufacturing fabrication. Using the developed model and optimization algorithm, case studies have been conducted to show how to determine the preferred fabrication orientation for different geometrical objects.

McClurkin and Rosen [1998] applied a method based on response surface methodology and multi-objective decision support for relating build goals to the build style variables to provide support for making build style decisions.

Charney and Rosen [2000] presented an empirical model for SLA accuracy, as specified by geometric tolerances, and a process planning method based on response surface methodology and multi objective optimization.

Williams and Deckard [1998] performed physical experiments and conducted implementation of a numerical simulation for an SLS process. The effects of selected parameters on the SLS process response are examined, where the primary parameters of interest are the laser power, laser beam, laser beam velocity, hatching spacing, laser beam spot size and scan line length. Their study showed that secondary process parameters such as delay period had significant influence on the process response.

Tong *et al.* [2003] generated a generic approach to evaluate the volumetric accuracy of rapid prototyping machines. The approach included using an SLA machine to produce generic artifact which was then measured using a master CMM and the measurement results were used to infer the RP machine's parametric error functions.

Xu *et al.* [1999] discussed the selection of building direction for four RP processes, namely SLA, SLS, FDM, and LOM. The manufacturing time, building cost, dimensional accuracy and surface finish were taken into consideration when selecting appropriate orientation for part building. The building cost is chosen as the main optimization

objective. Other criteria like the volume of building inaccuracy the manufacturing time, the surface finish, are imposed as secondary optimization objectives to resolve tie breaks for orientations with the same building cost for a given model and process. The optimal orientations for part building with different RP processes have been demonstrated by the case study to be different for different RP processes. An optimal orientation algorithm was demonstrated on a part considered for processing with SLA. The influence of the process characteristics on the selection of appropriate orientation is illustrated in the example.

Han *et al.* [2003] studied enhancing FDM process efficiency. A build time analysis was conducted and the deposition parameters that can be used to speed up fabrication processes are identified. The tool-path deposition planning approach is extended for ensuring layer quality when the building process is expedited under adjusted deposition parameters.

Cheng *et al.* [1995] presented a multi-objective approach for determining the optimal part building orientation in SLA process. Different objectives such as part accuracy and build time have been considered, and objective functions were developed based on known sources of errors affecting part accuracy and the requirements of good orientations during the building of a model. Attaining the specified accuracy achievable with the process was set as the primary objective, following as a secondary objective was to minimize build time.

Ziemian and Crawn [2001] developed a multi-objective decision support system to aid the user in setting FDM process variables in order to best achieve specific build goals and desired part characteristics. Their method uses experimentation to quantify the effects of FDM process variables on part build goals, and to predict build outcomes and expected part quality.

Another issue affecting the quality of the fabricated part is the path plan for each layer. In fused deposition this is considered the deposition strategy and refers to the path

that the nozzle tip follows in tracing out the geometry of each layer. Various methods of filling the interior of each layer have been researched in order to produce parts quickly, that are strong, or that have a good surface finish. Work in this area includes that by Kulkarni and Dutta [2000], Qui and Langrana [2001], [2002], Ahn, *et al.* [2002], Vasudevarau, *et al.* [2000], Yang, *et al.* [2002], McMains, *et al.* [2000], Arni and Gupta [1999], Onuh and Hon [1998], Han, *et al.* [2002].

Researchers are now focusing on a relatively new technology trend, which builds parts using variable rather than uniform layer thickness, better known as adaptive slicing. Adaptive slicing refers to a situation where the layer thickness varies in different regions of the part, allowing thicker layers where surface accuracy is not important, and thinner layers where it is crucial to minimize the stair-stepping effect. This offers a trade-off between surface finish and build time, which allows a part to be built as quickly as possible while retaining the accuracy of functionally crucial part features. Relevant research in this area includes: Jeng *et al.* [2000], Zhou *et al.* [2004], Lou *et al.* [2001], Hope *et al.* [1997a,b], Tyberg [1998], Tata [1998], Choi [2002b], Pandey *et al.* [2003], Zhang [2002], Lee [2000], Xu [1997], Ma and He [1999]. The work in this research will not be focusing on this new trend, since many RP machines still do not support this technology.

Another important factor to be considered when searching for optimal RP fabrication processes is the minimization of support structure. For processes where external support may be required, certain orientation may result in the use of a greater volume of external support and hence, longer time. Support structures enable a floating component to be built without dropping. They also prevent the overhanging surface from toppling.

Hur and Lee [1998] addressed the development of a CAD environment to determine the preferred build up direction for layered manufacturing taking in consideration the minimization of support structures.

The surface roughness models that the different researchers used for their optimization problems with respect to the most common RP technologies are summarized in Figure 3.1.

	RP Machine Technology				Control Parameters				Design Objectives					
Paper/ Sources	SLA	SLS	FDM	Others	Layer Thickness	Build Direction	Road Width	Others	Surface Roughness	Support Structure Volume	Build Time	Others		
Zieman [2001]	o	o	x	o	x	x	x	Deposition Strategy	x	o	x	Dimensional Accuracy	Experimental Design	
Han [2003]	o	o	√	o	√	o	√	Work Speed	o	o	√	Layer Quality Path Plan	Analysis[time]	
Xu [1999]	√	x	x	x	√	√	o	o	√	o	√	Cost	Primary Optimization	
Hur [1998]	√	o	o	o	√	√	o	o	√	√	√	o	Primary Optimization	
Cheng [1995]	√	o	o	o	√	√	o	o	√	o	o	o	Multi-Objective	
Reeves [1997]	√	o	o	o	√	√	o	o	√	o	o	o	Experimental	
Campbell [2002]	x	x	x	x	√	√	o	Layer Profile	√	o	o	o	Empirical Investigation	Surface Roughness Visualization Algorithm
Zhou [2000]	√	o	o	o	o	o	√	Work Speed	√	o	√	o	Experimental Design	
Perez [2001a]	√	o	o	o	√	√	o	o	√	o	o	o	Experimental	
Perez [2001b]	√	o	o	o	√	√	o	o	√	o	o	o	Experimental Analysis	
Giannatsis [2001]	√	o	o	o	√	o	√	Work Speed	o	o	√	o	Experimental Investigation	
Lan [1997]	√	o	o	o	√	√	o	o	√	√	√	o	Algorithm	
Hur [2001]	o	√	o	o	√	√	o	o	o	o	√	Fab. Orient.	Genetic Algorithm	
Hu [2002]	o	o	o	Hybrid	√	√	o	Tool Path	o	√	√	Fab. Orient+ Cost	Algorithm	
Lin [2001]	x	o	x	LOM	√	√	o	Layer Profile	o	o	o	Fab. Orient.	Optimization Algorithm	
McClurkin [1998]	√	o	o	o	√	√	√	o	√	o	√	o	Response Surface Methodology	
Lynn-Charney [2000]	√	o	o	o	√	√	√	o	o	o	√	o	Response Surface Methodology	
Tong [2003]	√	o	o	o	√	o	√	o	o	o	o	Parametric Error	Experimental	CMM
Williams [1998]	o	√	o	o	√	o	√	Work Speed	o	o	√	Strength + Energy	Numerical Model Simulation	Experimental
Yang [2002]	x	x	x	o	√	o	√	Path Plan.+ Work Speed	√	o	√	o	Experimental+ Simulation Analysis	

Direct connection	√
Indirect connection	x
No connection	o

Figure (3.2) Optmization literature review matrix

3.5 Literature Review Summary

It is evident from the literature review summary that, in the past few recent years, there is a substantial amount of research that covered the area of layered manufacturing modeling and optimization. The work available so far considered the selection of build orientation, support structures, layer thickness, road width, and layer path planning, in optimizing build time, surface finish, dimensional accuracy or part strength. Most of the work dealt with various process variables towards optimization of a single objective, but far less research considered the multi-objective optimization of several objectives with respect to numerous variables, as shown in the literature review matrix in Figure 3.1. The multi-objective optimization has been addressed in some detail, however, with respect to SLA. Such work has examined the effects of two or more process variables on the quality of the SLA parts.

The research presented in this report addresses a multi-objective optimization problem associated with the FDM process. The goal is to minimize the following build objectives: (1) surface roughness, (2) support structure volume, (3) build time, and (4) dimensional deviation, and to optimize them with respect to several different FDM process parameters: (1) build orientation, (2) layer thickness, and (3) road width.

The next chapter will explain in details the optimization of the rapid prototyping process parameters with respect to the suggested build goals.

CHAPTER 4

OPTIMIZATION OF RAPID PROTOTYPING PROCESS PARAMETERS

This chapter discusses the crucial RP process parameters and their influence on the suggested build objectives, followed by the development of the mathematical models, and the formulation of the utility function for the multi-objective optimization problem.

4.1 Process Parameters

Researchers have classified RP process parameters into three main classifications: (1) Nuisance, (2) Constant, and (3) Control parameters. Nuisance parameters include age of laser, beam position accuracy, humidity and temperature, which are not controlled in the experimental analysis, but may have some effect on a part. Constant parameters normally include beam diameter, laser focus and material properties, etc. The control parameters will affect the output of the process and are controllable in a run. These include layer thickness, build orientation, road width, path plan, shrinkage of the material, etc. The layer thickness, build orientation, and road width are the most vital among control parameters. It is also agreed by researchers that control parameters are the most influential among other process parameters. The next sub-sections will explain the vital control parameters and their effects on the suggested build objectives.

4.1.1 Build Orientation

Build orientation, also known as part orientation, is one of the most important factors affecting surface roughness. One of the main reasons is that the orientation decides whether a particular surface of a part is going to be fabricated as a sloped surface or as an orthogonal surface with respect to the build platform. A sloped surface is going to be approximated by layers of definite thickness offset from each other, leading to the infamous "stair-stepping effect" as can be seen in Figure 4.1. Also, build orientation

decides among other things, build time, part strength, the amount and location of support structures and hence the resultant surface finishes, etc.

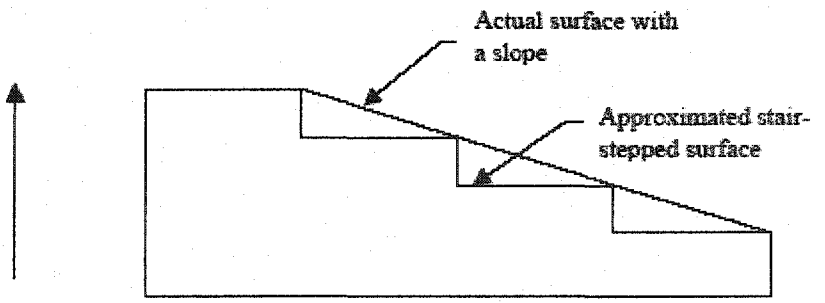


Figure (4.1) An illustration of the "stair-stepping effect"

Stair-steps result in a higher surface roughness. The amount of roughness on a surface due to stair-steps is directly dependent on the inclination of that surface with the horizontal base plane. On a FDM system, vertical walls or surfaces have the best surface finish or the least surface roughness followed by horizontal surfaces. Orienting a part "correctly" is very important when definite surface finishes are expected on individual surfaces of the part. Orientation can thus be used as a tool to moderate the undesirable effect of stair-stepping.

Therefore the importance of orientation when building a part on any rapid prototyping system cannot be understated. A good orientation is going to ensure among other things, a good surface finish on the critical surfaces of the part if not on all of the part's surfaces.

4.1.2 Layer Thickness

Layer thickness is the term given to the height of one layer in the z direction or in other words the user specified thickness increment of layers in the build direction. Since all rapid prototyping processes are layered manufacturing processes, the generation of layers is inherent to the process. The layer thickness determines the height of the vertical part of a step on sloped and vertical surfaces. The thickness of the layers will determine various aspects of the built part including: surface roughness, build time, ability to accurately represent a feature on the part, etc. The rapid prototyping machines

commercially available, can build parts at various layer thicknesses starting from a low of 0.001" (0.0254 mm) found in the new machine from 3D Systems, SLA 7000, among others [3D Systems, 1999]. On the upper end, a user would typically limit the maximum layer thickness to around 0.010" since thicker layers will lead to unimpressive prototypes due to the rough surfaces caused by stair-stepping. Of all the aspects of the part that layer thickness is going to affect, perhaps the most serious is the occurrence of stair-stepping or distortion on sloped surfaces, leading to high roughness on such surfaces, as shown in Figure 4.2.

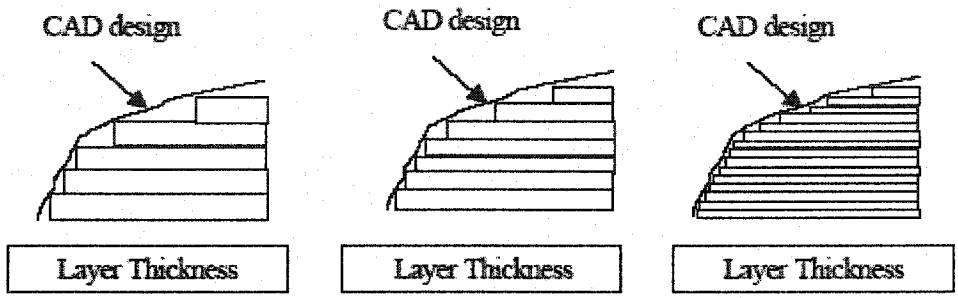


Figure (4.2) Effect of layer thickness on stair-stepping

It is evident that the layer thickness increment determines the height of the stair-step. The lesser the layer thickness the closer it is to the original CAD design and therefore the lesser the stair-step produced on the prototype. On the contrary the larger the layer thickness the more surface roughness would occur.

In this research, part orientation and layer thickness are the two parameters that have been considered to study their effect on surface finish or surface deviation as will be explained later in this chapter.

4.1.3 Road Width

As the nozzle is moved over the table in a prescribed geometry, it deposits a thin bead of extruded plastic, referred to as "roads" which solidify quickly upon contact with substrate and/or roads deposited earlier. Solid layers are generated by following a rastering motion, where the roads are deposited side by side within an enveloping domain

boundary, shown in Figure 4.3. Once a layer is completed, the platform is lowered in the z direction in order to start the next layer. This process continues until the fabrication of the object is completed.

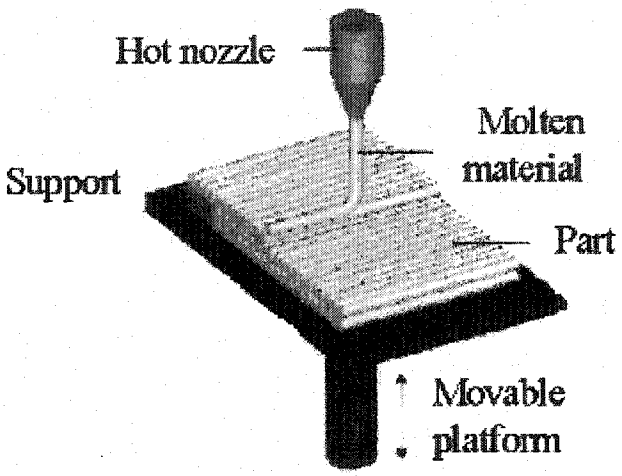


Figure (4.3) Schematic illustration of a road deposition process

The width of the road is referred to as road width.

4.2 Build Objectives

A model has been developed for different objective functions for the most crucial and controllable decision variables. The objective functions chosen to be optimized were the surface roughness, overhanging volume, build time and dimensional accuracy. Part orientation, layer thickness and road width were identified as the key control parameters.

The chosen objective functions are largely determined by the identified parameters. This chapter will show how each competing objective function is affected by the identified parameters.

4.2.1 Surface Roughness

Surface finish and surface texture are great concerns in many RP applications such as those involving the use of prototypes as investment casting patterns or as aerodynamic test models. By and large, the most dominant surface feature in most RP applications is

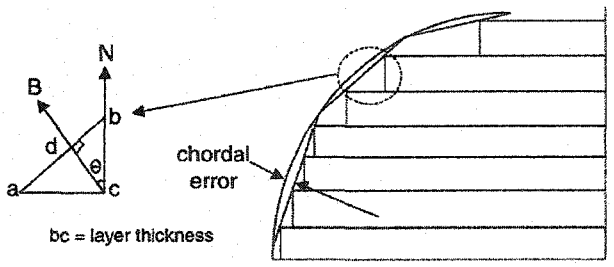
the stair-stepping effect caused by orienting a flat or contoured surface not orthogonal to the x-y plane. In general, higher resolution of contoured surfaces can be obtained by orienting the surface orthogonal to the x-y plane. This stair-stepping effect, as demonstrated in Figures (4.1 and 4.2), is common to all current rapid prototyping fabrication processes [Degarmo, 2003].

The orientation of an RP part will cause change in the inclination of certain faces of the fabricated part. Orienting a part in the optimal direction will induce a relatively smaller angle between the facets and the build direction, resulting in a lower surface roughness. Higher resolutions can also be obtained by reducing the layer thickness during the build cycle. However, a trade-off typically exists between build speed of the machine and the thickness of a layer.

Much research predicted the surface quality of a part by analytically computing the stair-step error for a part, such as [Ziemian, 2001]. Therefore, the equations would be functions of layer thickness (L), surface area (A), and surface angle (Θ). The surface area and angle of each part facet (i) are determined from the STL file, and the total surface error for the part is computed as a weighted summation of the facet errors. This estimated stair-step volume as a representation of the surface roughness is computed using equation (4.1).

$$SE = \sum_i (L / 2) (A_i) (\cos \Theta_i) \qquad (4.1)$$

However, using this equation does not accurately indicate the surface roughness of the part with respect to its total surface area. Others calculated the average cusp height, shown in Figure 4.4, but with respect to the number of faces [Choi, 2002], which again is not completely indicative of the total surface roughness with respect to the whole part, incase having parts with same number of faces, same inclination angles, but smaller dimensions.



Where,

Cusp height (C) = bc cos Θ

Avg. cusp height (ACH) = Σ C / Nf

Nf : number of faces

Figure (4.4) Cusp height in layered manufacturing

To enhance this representation, a model was developed to calculate the average cusp height, but this time with respect to the overall surface area of a part. This representation is shown in equation (4.2), which is functions of layer thickness (L), cusp height (C), surface angle (Θ), total surface area (A_{total}), area of the i^{th} inclined face (A_i), and the number of inclined faces (n).

$$ACH = \frac{\sum_{i=1}^{n} A_i * C}{A_{total}} \qquad (4.2)$$

4.2.2 Support Structure Volume

Support structure is another important factor that influences the rapid prototyping process. A support structure is almost always necessary to build a part. The most common situation is to support the surfaces of the part so that they will not warp, sag, or parachute as the elevator or platform moves up and down. The need for support structures increases for overhanging surfaces on which material is solidified continuously. In this case, support structures prevent the overhanging surface from toppling. The support structures are formed simultaneously with the original part. After solidification, the support structures must be removed. Because this post-treatment process is often performed manually, the more support structures, the more time is required for the finishing operations. Not only that, but also the cost of the prototype will increase with the increase of support structure. Therefore, our objective was to minimize the support structure used in building the prototype.

Researchers such as Hur, *et al.* [1998] considered the sum of projected areas of inclined faces to calculate the support structure usage. A more accurate method is to multiply the projected area by the average height of overhanging faces, shown in equation (4.3), which is a function of the projected area of i^{th} overhanging face (Ap_i), surface angle (Θ), number of overhanging faces (m), and average height of overhanging face vertices ($H_{ave.}$).

$$VS = \sum_{i=1}^{m} Ap_i * H_{ave} \tag{4.3}$$

4.2.3 Build Time

A major motivating factor in the development of RP processes has been the reduction in product development time. Therefore, the build time of RP processes is a major concern. Build time in rapid prototyping processes consists of three major components: (1) preprocessing, (2) fabrication, and (3) post-processing. Preprocessing involves the conversion of CAD solid models into the control data needed to operate RP machine tools and make the part, the slicing procedure, and the generation of paths or roads for each layer. With support structure requirements, the preparation time also includes the determination and modeling of support volumes. Post-processing involves any manual finishing of the part after the automated fabrication cycle such as, the detachment of the part from the foam base, and the removal of the support material from the part surfaces, etc.

While variations in preprocessing and post-processing times exist among RP processes and machine tools, preprocessing times are becoming less important with the development of faster computers. The largest component of build time, and consequently cost, is the actual time required to fabricate the model. Currently, pre- and post-processing costs added together range from 10% to 50% of fabrication costs with most processes averaging 20%. This can vary depending upon the geometric complexity of the part as well as the number of part produced at one time (i.e., the batch size). All RP processes have the ability to nest multiple work pieces within their respective work

envelopes (i.e., their maximum work space), which can save time and money. When batch sizes exceed about 10 to 20 parts, post-processing times can become quite significant depending upon the process used.

In general, for RP equipment, the fabrication time is made up of two components: (1) layering, and (2) patterning. Layering involves the bulk deposition of the raw material to be patterned. (Deposition-based processes do not require a separate layering step). Of these two components, the patterning step is usually the longest. As a result, the material addition rates (MARs) associated with the patterning step, are the most representing of the total fabrication time for model. The MAR can be defined as the volume of material added per unit time [Degarmo, 2003].

Ziemian and Crawn [2001], created a model by normalizing individual response surfaces with respect to part volume and shape, and averaging the normalized regression coefficients. The build time regression results demonstrated a good fit between the response surface and the data for each of the different fabricated shapes. To function properly as a predictive model, the build parameter of part volume is explicitly incorporated into the response surface. The final model that was adopted by this research, representing build time per unit volume (seconds/square inch), can be seen in equation (4.4). The build time (T) is a function of layer thickness (L), and road width (Rw).

$$T = 6320 - 2005L - 2299Rw + 454L^2 \qquad (4.4)$$

4.2.4 Dimensional Accuracy

Several points must be considered when evaluating the accuracy of prototypes made on RP processes. First, and most important, is that operating conditions greatly affects the dimensional accuracy of prototypes. That is, a prototype fabricated under one set of processing conditions may have different overall accuracy than a part built under another set of conditions.

Another point to consider is the size of the parts to be fabricated. For most processes, the part accuracy greatly improves as the measured dimension decreases. This is largely due to phase changes in the material as a result of processing. Specifically, a material is transformed from a liquid to a solid or, in some cases, from a solid to a liquid and then back to solid again. In each case, the phase change from liquid to solid involves an increase in density and a resulting shrinkage. The total volumetric shrinkage varies from process to process and from material to material. However, all RP processes experiencing a phase change involve some level of volumetric shrinkage or some sort of deviation.

The simplified model representing the average absolute deviation (AD) in inches, as seen in equation (4.5), was again adopted from Ziemian and Crawn [2001].The model is also a function of layer thickness and road width.

$$AD = 0.005961 - 0.000714L + 0.000625Rw^2 \qquad (4.5)$$

Both build time and dimensional accuracy models that were adopted have been tested and based on design of experiments.

4.3 Multi-Objective Optimization

In order to optimize the selected control parameters, multi-objective optimization had to be utilized to gather the different competing objectives previously presented in a single objective function. This section describes the formation of the utility function, the normalization of the objective function magnitudes, the weighting of the objectives within the utility function, and the general formulation of the problem.

4.3.1 Utility Function

To solve the multi-objective optimization problem, all the objectives were gathered in a single utility function. The most common method for multi-objective optimization is the weighted sum method [Marler, 2004], by determining weighting factors for each objective, and summing them together as shown in equation (4.6).

$$\text{Utility Function} = w_1 * \text{ACH} + w_2 * \text{VS} + w_3 * \text{T} + w_4 * \text{AD} \tag{4.6}$$

4.3.2 Normalization

When modeling any utility function, it is not logical to add objective functions with orders of magnitude that are too far apart. Therefore, the orders of magnitude of the different objective functions had to be normalized in order to vary between the range of 0 and 1. Equations (4.7-4.10) present the formulae implemented to obtain the normalized values of the different objective functions.

$$ACH_{norm.} = \frac{ACH - ACH_{min}}{ACH_{max} - ACH_{min}} \tag{4.7}$$

$$T_{norm.} = \frac{T - T_{min}}{T_{max} - T_{min}} \tag{4.8}$$

$$AD_{norm.} = \frac{AD - AD_{min}}{AD_{max} - AD_{min}} \tag{4.9}$$

$$VS_{norm.} = \frac{VS}{V_{env} - V_{solid}} \tag{4.10}$$

Where

- ACH_{max} and ACH_{min} are the absolute maximum and minimum values of the average cusp height that can be obtained from the available domains of the different decision variables.
- T_{max} and T_{min} are the absolute maximum and minimum values of the build time that can be obtained from the available domains of the different decision variables.
- AD_{max} and AD_{min} are the absolute maximum and minimum values of the average cusp height that can be obtained from the available domains of the different decision variables. V_{env} is the total envelope volume of the part.
- V_{solid} is the total solid volume of the part.

4.3.3 Weighting

By observing the relationship between the crucial parameters and the suggested build objectives, illustrated in Table 4.1, it is obvious that there is a conflict making the choice of weighting factors complicated.

For instance, the surface roughness and the support structure volume are both competing objectives that are function of build orientation while build time and dimensional accuracy are not. Therefore, it made sense that surface roughness and support structure volume should have equal weights. Built time and dimensional accuracy are both functions of the same variables, which are layer thickness and road width. They both seek to increase layer thickness, but dimensional accuracy seeks a decrease in road width while build time seeks to increase it. Therefore, it was logical to give them equal weights to have a fair competition. Finally, by looking at surface roughness with relation to layer thickness, it requires the layer thickness to be decreased to minimum, unlike build time and dimensional accuracy. In this case, it would be a good choice to give the surface roughness a weight that is double the weight of build time and dimensional accuracy.

Table 4.1: Relationship between process parameters and build objectives

	Orientation	**Layer Thickness**	**Road Width**
Surface Roughness	✓	↓	×
Support Structure Volume	✓	×	×
Build Time	×	↑	↑
Dimensional Accuracy	×	↑	↓

Therefore the weighting factors assigned to the suggested objective functions according to the above table were as follows:

Surface roughness	= 0.33
Support structure volume	= 0.33
Build Time	= 0.17
Dimensional accuracy	= 0.17

4.3.4 General Formulation

According to the process parameters described earlier, we have three decision variables, two of which are continuous while the third is discrete. The orientation (Or) is considered as a discrete variable whose domain consists of the different possible alternative orientations that makes the part rest on one of its flat surfaces. The continuous variables are the layer thickness and road width. When dealing with FDM machines, each machine has different settings for both the layer thickness and road width values. According to Ziemian, *et al.* [2001], the layer thickness values for the FDM2000 machine vary between 0.178 - 0.33mm and the road width between 0.333 - 0.706mm. These values were set to run the algorithm for the case study in hand.

After forming the utility function, normalizing the values of the different objective functions and selecting their appropriate weights, the general formulation of the optimization problem in hand is expressed in equation (4.11).

$$\text{Minimize UF} = 0.33 * \text{ACH} + 0.33 * \text{VS} + 0.17 * \text{T} + 0.17 * \text{AD} \qquad (4.11)$$

Subject to:

$\text{Max.}(C) \leq C_{all.}$

$\text{Or} = \{1,2,3,\ldots N_{or.}\}$

$0.178\text{mm} \leq L \leq 0.330\text{mm}$

$0.333\text{mm} \leq Rw \leq 0.706\text{mm}$

As noted above there has been a set constraint for the maximum allowable cusp height ($C_{all.}$). This is predefined by the user, so as to make sure that the surface roughness of the produced part will not exceed a certain limit of his desire. To put this constraint into consideration, a penalty function, that increases exponentially as the cusp height constraint is violated, was added to the equation.

4.4 Utilized Optimization Method

Previously, as mentioned earlier in Chapter 2, most of the research work dealt with the problem by primary optimization methods. These problems were usually uni-modal and that is why it was sufficient to use these methods.

As for the problem in hand, it is much more complicated since it deals with more than one parameter and more than one objective function, causing it to be a multi-modal problem. In this case it is desirable that the optimization method used is capable of arriving at a global optimum solution rather than the use of classical optimization that will always be trapped into local optimality. Genetic Algorithms is the selected global optimization method used for this optimization problem. The next chapter gives a brief description of GAs and its adaptation to the problem in hand.

CHAPTER 5

GENETIC ALGORITHM

This chapter presents a variant of the Genetic Algorithm known as "real-coded genetic algorithms". It is suitable for the global optimization of problems containing continuous parameters. Figure 5.1 shows examples of multi-modal functions (i.e. functions with several minima). If direct search (or gradient based) methods are used to optimize such functions, the minima at which the search will arrive, depends on the start point as shown in Figure 5.1. However, unless a good guess is found for the start point, there is no guarantee the search will arrive at the global minimum, or at least settle at a point in its close vicinity.

Random search algorithms have achieved increased popularity due to the shortcomings of calculus-based and enumerative based techniques. There are three main methods that fall in the category of such algorithms. These are: (1) Genetic Algorithms, (2) Simulated Annealing and (3) Tabu Search. Genetic Algorithms are an example of a search procedure, which uses random choice to guide a highly exploitative search through coding of the parameter space and iterative application of search movements which mimic natural genetics.

It should be noted that all of the above methods arrive at a near global optimum due to their semi-random nature.

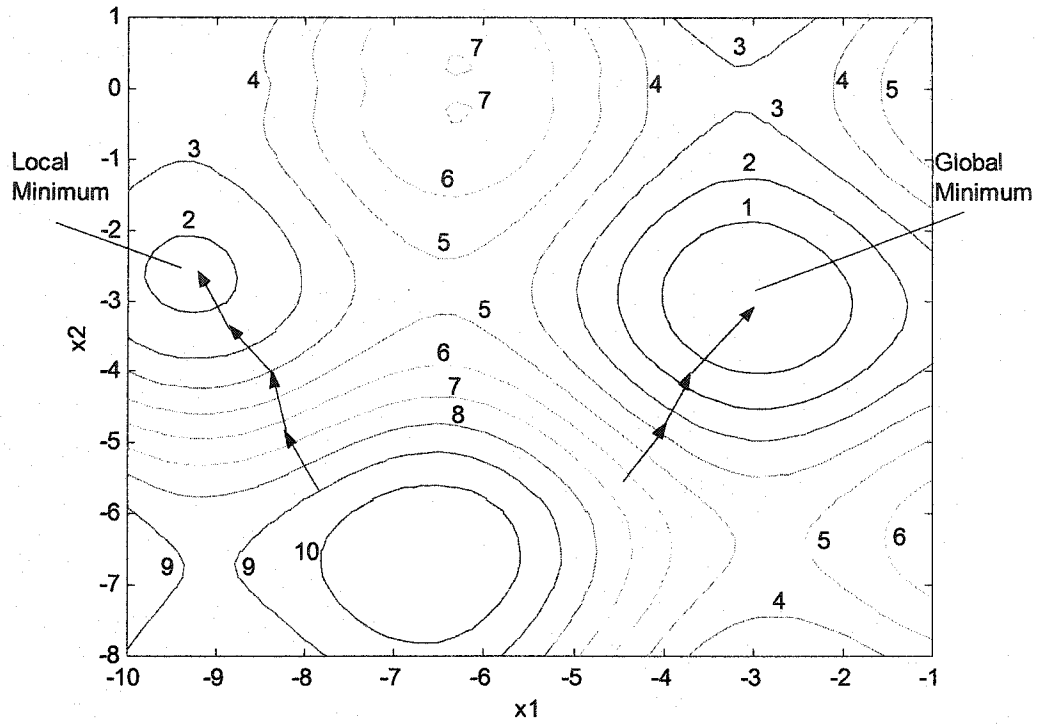


Figure (5.1) Multi-modal function

5.1 Overview of Genetic Algorithms

Genetic Algorithms are different from other normal optimization and search procedures in four ways:

1. GAs work with a coding of the parameter set, not the parameters themselves.
2. GAs search from a population of points, not a single point.
3. GAs use payoff (objective function) information, not derivatives or other auxiliary knowledge.
4. GAs use probabilistic transition rules, not deterministic rules.

The correspondence of Genetic Algorithm terms and optimization terms is summarized in Table 5.1

Table 5.1: Explanation of GA Terms (Gen, 1997)

Genetic Algorithms	Explanation
Chromosome (string, individual)	Solution (Candidate)
Genes (bits)	Part of solution
Locus	Position of gene
Alleles	Values of gene

A genetic algorithms (GA) starts with a population of randomly generated *chromosomes*, and advances toward better chromosomes by applying genetic operators, modeled on the genetic processes occurring in nature. The population undergoes evolution in a form of natural selection. During successive iterations, called *generations*, chromosomes in the population are rated for their adaptation as solutions, and on the basis of these evaluations, a new population of chromosomes is formed using a selection mechanism and specific genetic operators such as cross-over and mutation. An *evaluation* or *fitness function*, f, must be devised for each problem. The fitness function returns a single numerical fitness, which is supposed to be proportional to the utility of the solution which the chromosome represents. The following section details the above.

5.2 The Simple Genetic Algorithm

The simple genetic algorithm was first introduced by John Holland in 1975. The algorithm operates on binary strings, which means that the variable space should be discretized into binary code. Such a step is known as the *coding* step. Please refer to appendix A for more detail on Simple Genetic Algorithms.

5.3 The GAs Used in the Optimization Problem.

As seen in Appendix A, the traditional genetic algorithms discretize the continuous domain variables. Coarse discretization limits the search resolution and might lead to near-to-global optimal solutions. On the other hand, fine discretization leads to long

binary chromosomes and hence would increase the search space. Such increase may be drastic leading to prohibiting large search spaces.

The GAs used in solving the optimization problem use mixed discrete and continuous values instead of binary strings for each variable. In the problem used there are three variables; layer thickness, road width, and orientation. Layer thickness and road width are continuous numbers and the orientation is discrete. Figure 5.2 shows an example of a typical population that is used.

x_1 (Orientation)	x_2 (Layer Thickness)	X_3 (Road Width)
1	0.308	0.404
4	0.181	0.442
.	.	.
.	.	.
.	.	.
.	.	.
2	0.180	0.627

Figure (5.2) A population of chromosomes -Mixed GAs

5.4 General Procedure for Mixed GAs Used

Procedure: General Procedure for Mixed GAs

1. Let $F(x_1, x_2, x_3)$ be an objective function to be optimized, where (x_1, x_2, x_3) are the independent variables, where each variable x_i ranges between a lower and an upper limit $[L, U]$.
   x_1: Is the Orientation (discrete)
   x_2: Is the Layer Thickness (continuous)
   x_3: Is the Road Width (continuous)

2. Generate a random population P of N instances of the independent variables (known as *chromosomes*).
3. For a pre-specified number of generations (iterations)
   a. Let the total number of offspring chromosomes due to the application of the mutation and cross-over operators be denoted by M.
   b. Use the *selection* operator to fill a new population with N-M high fitness chromosomes.
   c. Use the *selection* operator along with the *mutation* and *cross-over* operators to fill the remaining M locations in the population.
   d. For the new population, evaluate the objective function (and fitness) value for the chromosomes changed by cross-over and mutation, and retain the fitness values of the unchanged chromosomes.

End Procedure

In this work there is a special form of selection, mutation and cross-over operators, which in a sense mimic those used in the binary-coded GAs. The following sections describe the operators that have been used.

5.5 Selection operator

The roulette-wheel selection is replaced with a fitness ranking selection method. The whole population is sorted in an ascending order according to fitness. The population is then assigned a geometric distribution, which is then used in the selection process as shown in Figure 5.3.

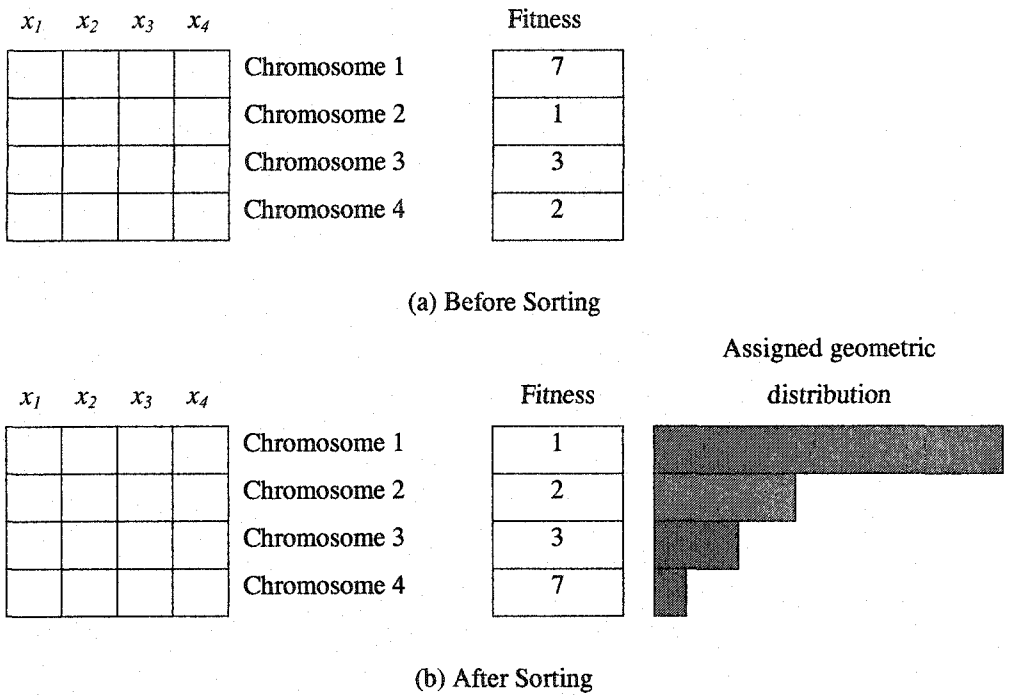


Figure (5.3) Ranking selection for a minimization process

This method of selection was used because in many cases the differences between the objective function values in the population become so small and the roulette wheel selection can loose the better chromosomes.

5.6 Mutation Operators

Mutation operators are random search elements within the genetic search that diversify the search within the domain of the independent variables. Since there is no guarantee that the generated chromosomes will have a better objective function values, then the parent chromosome on which the operator is applied should be chosen from

among the low fitness chromosomes. The different mutation operators that are used within the genetic algorithm developed are given in the following section as illustrated in Figure 5.4.

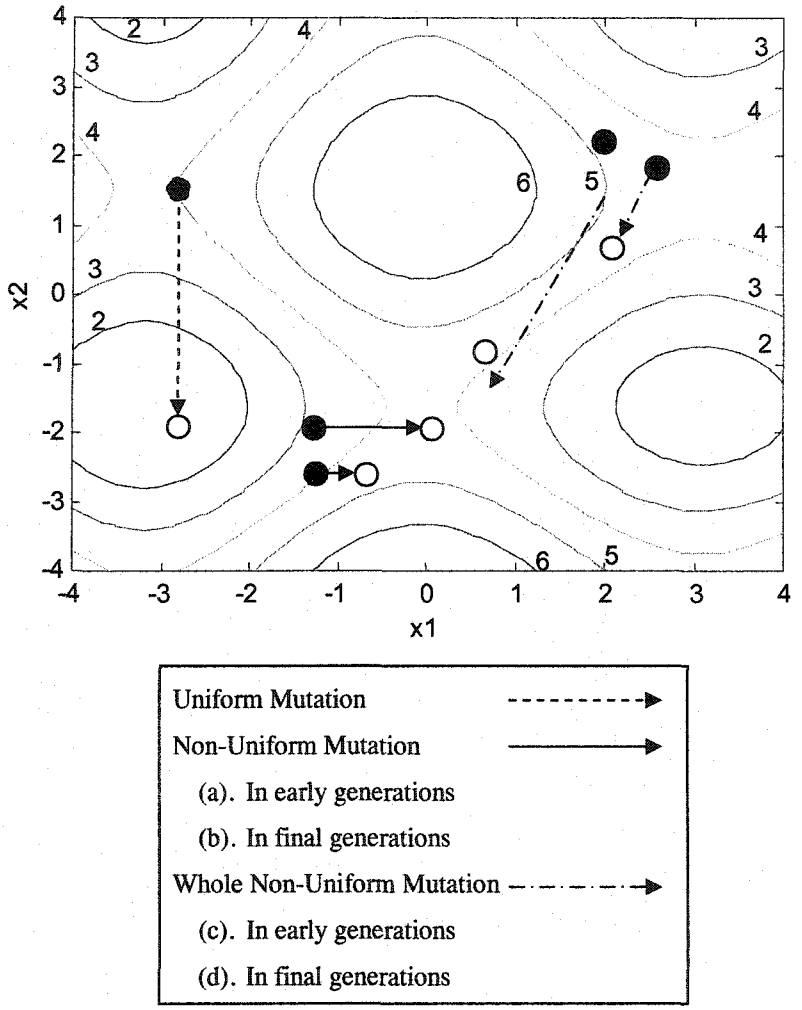


Figure (5.4) Different types of mutation operators

5.6.1 Uniform Mutation

Given a chromosome $\underline{X} = \{x_1, x_2, x_3\}$, replace x_k with a random number between $[L,U]$, where $[L,U]$ are the bounds on the variable x_k, where the location k is chosen randomly between 1 and n. Uniform mutation diversifies the search along a randomly chosen variable within the set of independent variables.

5.6.2 Non-Uniform Mutation

Non-uniform mutation is an operator that starts as a diversifying search element over large spaces around the mutated chromosome at the early stages of the search, and ends up with small variations around the mutated chromosome in the final generations. Non-uniform mutation is applied as follows: Given a chromosome $\underline{X} = \{x_1, x_2, x_3\}$, replace x_k by $x_k^{'}$ (k randomly chosen), where:

$$x_k^{'} = \begin{cases} x_k + \Delta(t, U_k - x_k) \\ x_k - \Delta(t, x_k - L_k) \end{cases}$$

Either of the above equations is chosen randomly.

$$\Delta(t, y) = y \cdot r\left(1 - \frac{t}{T}\right)^6$$

t =The number of the current generation

T =Maximum number of generations

r =Random value between [0, 1]

At the early stages of the search, the value [1-t/T] is large, and hence large variations from the mutated chromosome can be obtained. This value decays with generations, thus producing small variations.

5.6.3 Whole Non-Uniform Mutation

Given a chromosome $\underline{X} = \{x_1, x_2, x_3\}$, apply non-uniform mutation on all variables. This operator diversifies the search along the space of all variables

5.7 Cross-Over Operators

Cross-over operators vary chromosomes in a semi-local fashion to produce new chromosomes in the vicinity of the old ones, and hence should be used on chromosomes with high fitness values. The different cross-over operators that are used are shown below as illustrated in Figure 5.5.

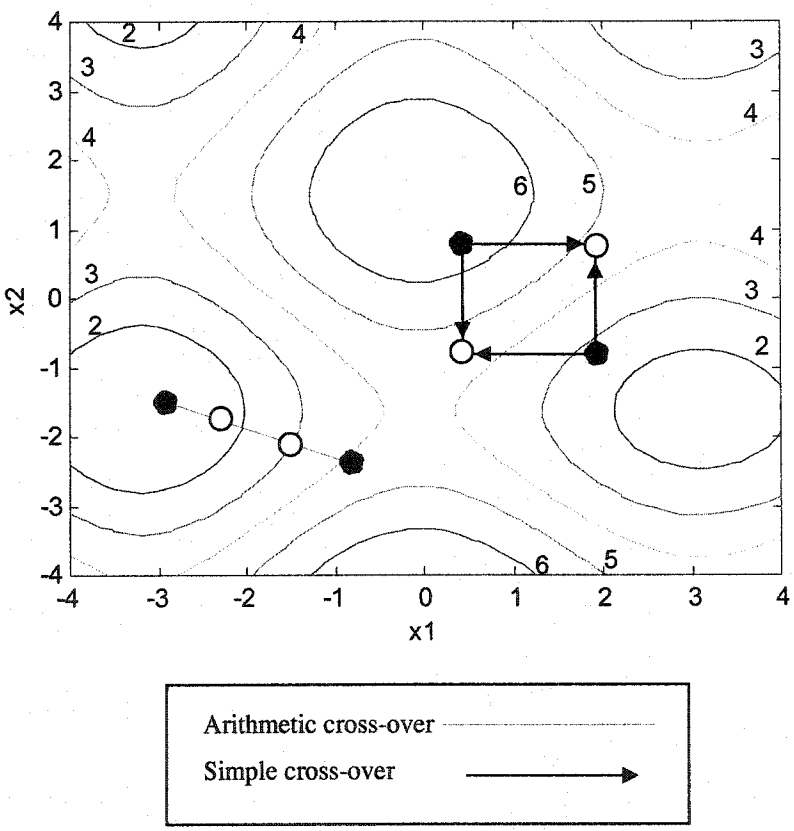


Figure (5.5) Different types of cross-over operators

5.7.1 Simple Cross-Over

Simple cross-over simulates the bit swapping found in the cross-over operator of binary coded genetic algorithms. Given a pair of parent chromosomes:

$$\underline{X}_1 = \{x_1^1, x_2^1, x_3^1\}$$

$$\underline{X}_2 = \{x_1^2, x_2^2, x_3^2\}$$

Choose a random location k, and produce the new chromosomes $\underline{Y}_1$ and $\underline{Y}_2$, by swapping the values in both chromosomes to the right of the location k.

$$\underline{Y}_1 = \{x_1^1, x_k^1, x_3^2\}$$

$$\underline{Y}_2 = \{x_1^2, x_k^2, x_3^1\}$$

This operator acts as an averaging search mechanism along the dimensions of the parent chromosomes.

5.7.2 Arithmetic Cross-Over

Given a pair of parent chromosomes:

$$\underline{X}_1 = \{x_1^1, x_2^1, x_3^1\}$$

$$\underline{X}_2 = \{x_1^2, x_2^2, x_3^2\}$$

Generate a random number α between [0, 1] and produce the new chromosomes $\underline{Y}_1$ and $\underline{Y}_2$, where

$$\underline{Y}_1 = \alpha\, \bar{x}_1 + (1-\alpha)\bar{x}_2$$

$$\underline{Y}_2 = (1-\alpha)\bar{x}_1 + \alpha\, \bar{x}_2$$

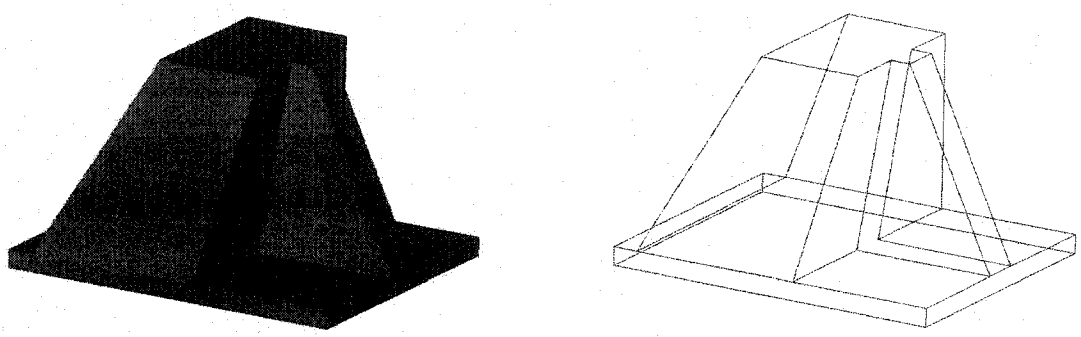
This operator produces new chromosomes on a straight line joining the parent chromosomes. It has some kind of an averaging effect between the values of the parent chromosomes. Such an operator is useful when a minimum is located between the parent chromosomes.

CHAPTER 6

RESULTS AND DISCUSSION

This chapter presents the results of applying the developed algorithm to a case study that was built using I-DEAS CAD/CAM software package.

6.1 Case Study

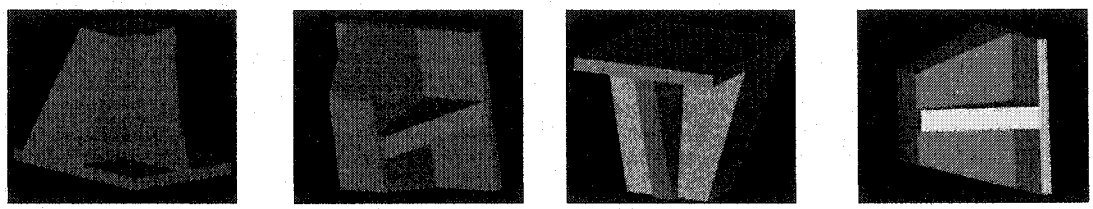


(a) (b)

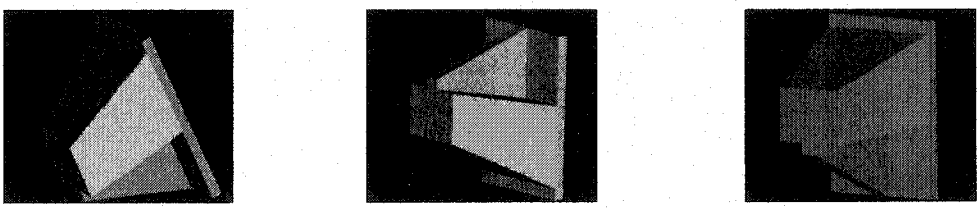
Figure (6.1) Geometric model of the case study in (a) Solid and (b) wire frame representations

The model of the part shown in Figure 6.1 was built on I-DEAS CAD/CAM software package. It was designed to include different face geometries with different angles of inclination, in order to keep it inconsistent. On the other hand, the edges were chosen to be straight and the model is symmetrical so as not to be very complicated. Figure 6.2 illustrates the seven different possible orientations for that part. Each orientation had different input data to the generated algorithm, since the inclined faces and their angles of inclination, etc. change with each orientation.

The different dimensions of the model including: face areas, angles of face inclinations, part volume, etc. were calculated as input data for testing the developed tool.



(a) (b) (c) (d)



(e) (f) (g)

Figure (6.2) Geometric model of the case study in different possible orientations

Before running the optimization algorithm the following optimization and GA settings had to be set:

Maximum allowable cusp height =0.25mm

Number of generations =50

Population size =50

Number of times to apply simple crossover =2

Number of times to apply arithmetic crossover =2

Number of times to apply uniform mutation =4

Number of times to apply non-uniform mutation =4

Number of times to apply whole non-uniform mutation =4

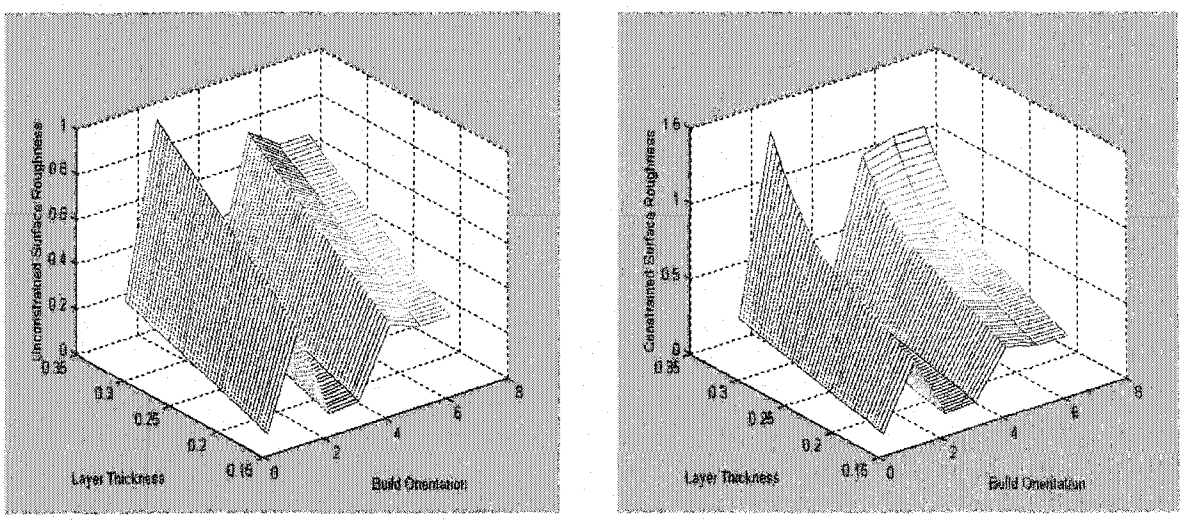
The next section will illustrate the results of the different parameters on our objectives after running the algorithm with the above optimization settings.

6.2 Results and Discussion

The results and their discussions are illustrated below showing the effects of the process parameters on surface roughness, support structure volume, build time, and dimensional accuracy objective functions respectively. The results are shown for two cases of the problem: (a) neglecting the effect of the maximum cusp height constraint (unconstrained) and (b) taking the constraint into consideration (constrained).

6.2.1 Surface Roughness

The graphs below, shown in Figure 6.3, demonstrate the effects of both layer thickness and build orientation on the surface roughness, which is indifferent to road width.



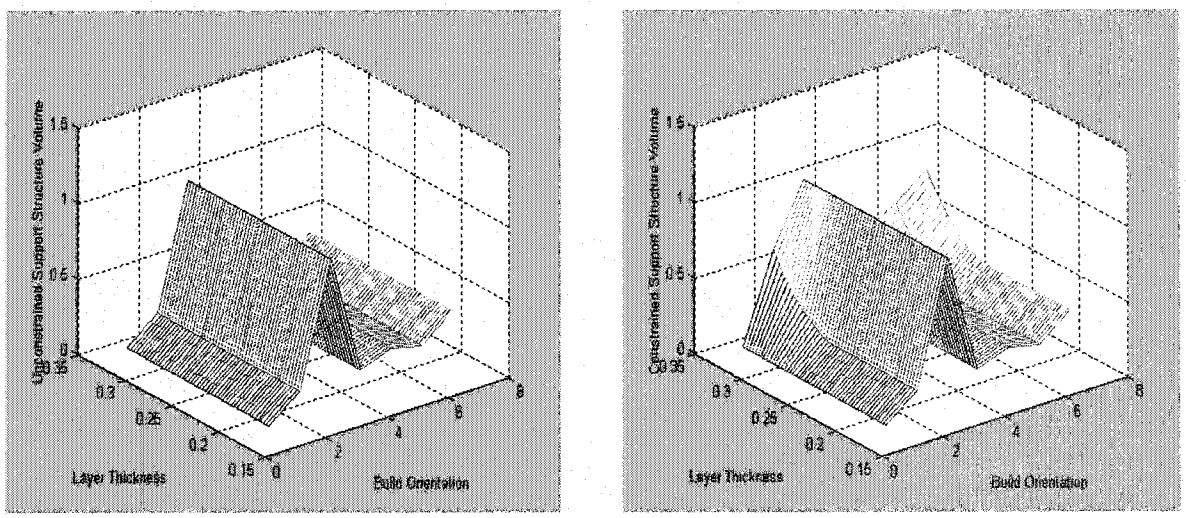
(a) (b)

Figure (6.3) Graphical illustration of the effect of layer thickness and build orientation on surface roughness (a) unconstrained and (b) constrained

From Figure 6.3a it is obvious that as the layer thickness increases, surface roughness increases, giving an undesirable finish. Orientation 2 is clearly the least favorable orientation with regards to surface roughness as it gives the highest value for roughness. The sudden increase in surface roughness shown in Figure 6.3b represents the effect of the added penalty function that was mentioned earlier in Chapter 4.

6.2.2 Support Structure Volume

As for the volume of the support structure, shown in Figure 6.4 which is only a function of build orientation as mentioned earlier, orientation 3 was the least favorable as it gives the greatest volume of support structure material to be used. Orientation 1 indicates the least volume of support structure and would therefore be regarded as the desirable orientation with respect to support structure volume.



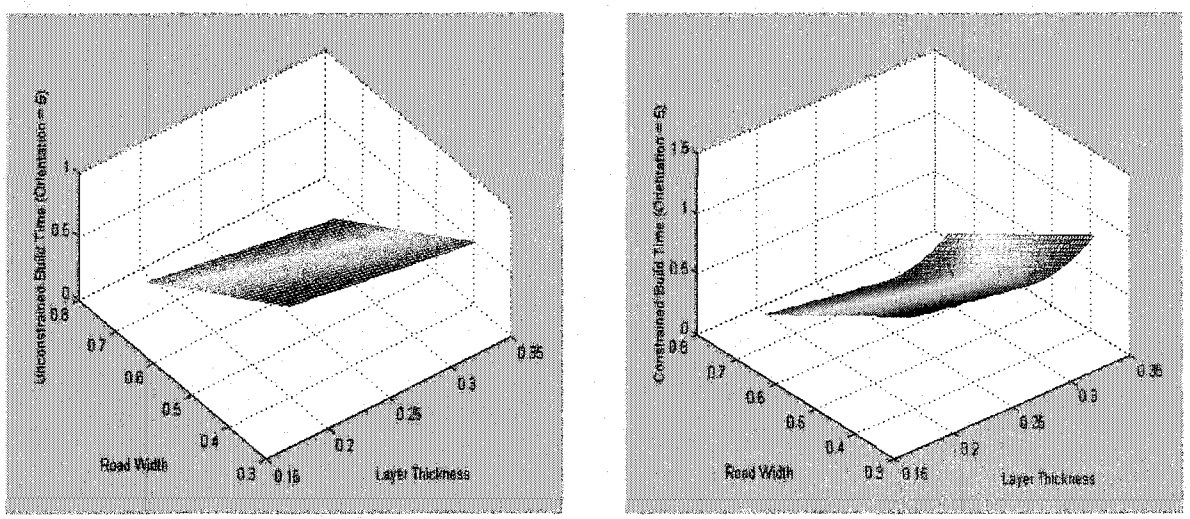
(a) (b)

Figure (6.4) Graphical illustration of the effect of layer thickness and build orientation on support structure volume (a) unconstrained and (b) constrained

Notice that the support structure volume in Figure 6.4a remains constant with changes in layer thickness values, while it suddenly increases in Figure 6.4b due to the penalty function.

6.2.3 Build Time

Regarding the build time objective, which is relative to both road width and layer thickness parameters, we could deduce from the charts in Figure 6.5 that the optimal result is achieved by maximizing both values of layer thickness and road width. To illustrate the results for the build time objective, which as mentioned before is not affected by orientation, orientation was randomly set to 5.



(a) (b)

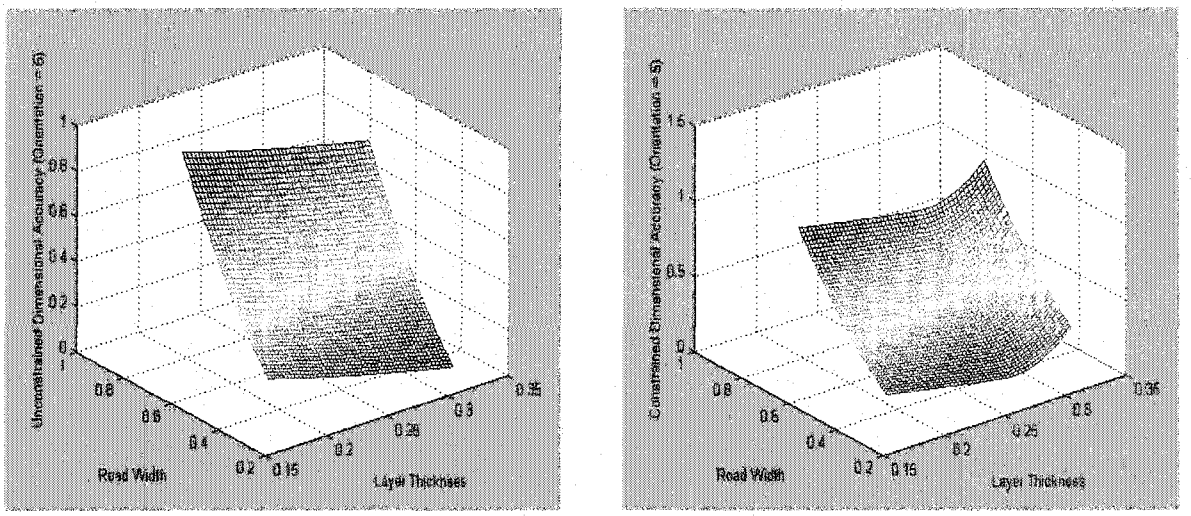
Figure (6.5) Graphical illustration of the effect of layer thickness and road width on build time (a) unconstrained and (b) constrained

The red zone in Figures 6.5a,b denotes the least desirable parameter values, while the deep blue zone indicates the optimal parameter values for the layer thickness and road width leading to the minimum build time.

6.2.4 Dimensional Accuracy

Similar to build time, dimensional accuracy is a function of both road width and layer thickness parameters, but instead seeks to decrease road width and increase layer thickness to reach optimal dimensional accuracy results. As illustrated in the charts in Figure 6.6, again randomly setting to orientation 5, the deep blue zone shows the

desirable dimensional accuracy results at minimum road width and maximum layer thickness values.



(a) (b)

Figure (6.6) Graphical illustration of the effect of layer thickness and road width on dimensional accuracy (a) unconstrained and (b) constrained

6.2.5 Utility Function

To reach the optimal results, taking into consideration the effect of all parameters simultaneously, all the different build objectives were gathered in a single utility function, as mention in Chapter 4. Since it is only possible to show the effect of two parameters at a time, Figure 6.7 illustrates the effect of layer thickness and build orientation on the utility function after fixing the value of the road width to 0.5 mm and Figure 6.8 shows the effect of road width and layer thickness after setting the orientation to alternative 4. These set values for the road width and the build orientation were anonymously chosen. Figure 6.9 illustrates the GA convergence curve.

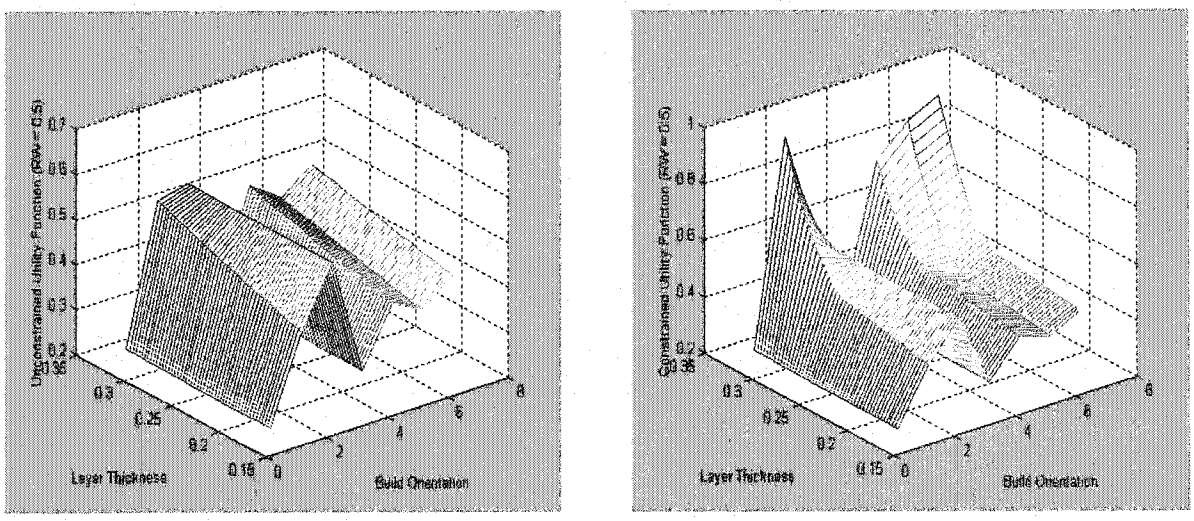
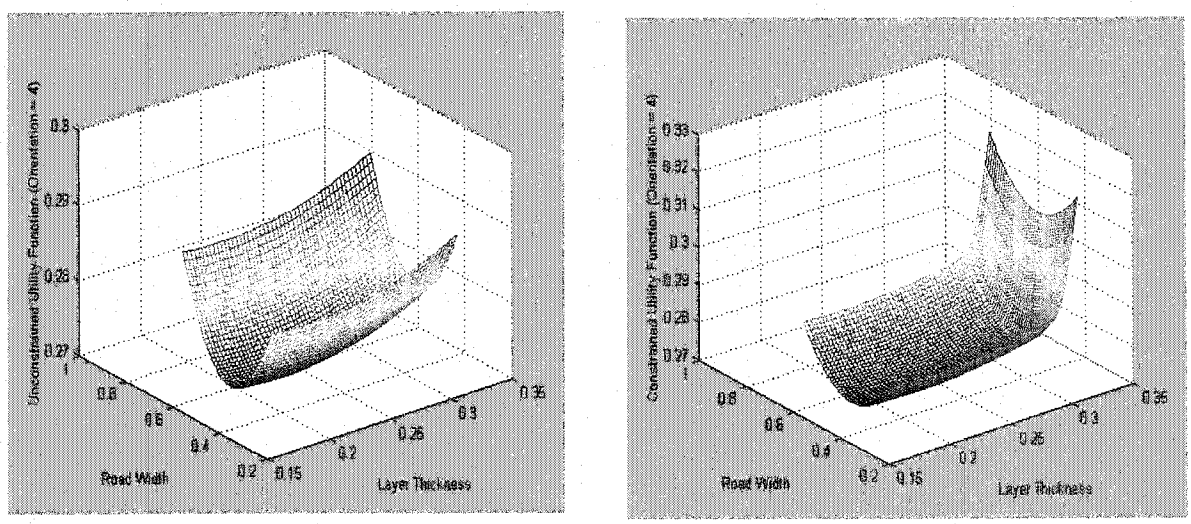
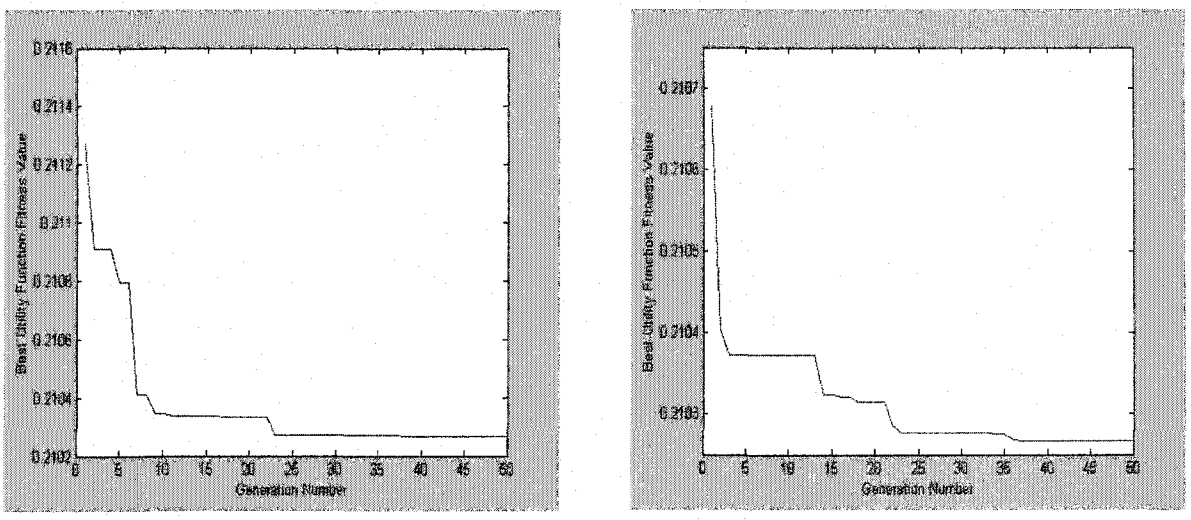


Figure (6.7) Graphical illustration of the effect of layer thickness and build orientation on the utility function (a) unconstrained and (b) constrained



(a) (b)

Figure (6.8) Graphical illustration of the effect of layer thickness and road width on the utility function (a) unconstrained and (b) constrained



(a) (b)

Figure (6.9) GA convergence curve (a) unconstrained and (b) constrained

The near optimal utility function value obtained by the developed optimization algorithm is 0.2102668 and Table 6.1 demonstrates its corresponding process parameter values.

Table 6.1: Near optimal process parameter values for continuous input values

Process Parameter	Near Optimal Value
Orientation	1
Layer Thickness	0.239mm
Road Width	0.503mm

According to the optimal parameters, the values of the process characteristics are demonstrated in Table 6.2.

Table 6.2: Optimal process characteristics results for continuous input values

Surface Roughness	Support Structure Volume	Build Time	Dimensional Accuracy
ACH_{min} =0.022 mm	V_{env} =1002903 mm^3	T_{min} =8.9721 h	AD_{min} =0.1487 mm
ACH_{max} =0.102 mm	V_{solid} =418173 mm^3	T_{max} =11.4481 h	AD_{max} =0.1565 mm
$ACH_{opt.}$ =0.030 mm	$VS_{opt.}$ =27192 mm^3	$T_{opt.}$ =10.3499 h	$AD_{opt.}$ =0.1519 mm
$ACH_{opt.norm.}$ =0.0956	$VS_{opt.norm.}$ =0.0465	$T_{opt.norm.}$ =0.5565	$AD_{opt.norm.}$ =0.4023
Maximum Cusp Height = 0.0994 mm			

6.2.6 Continuous vs. Discrete Parameters

As mentioned earlier, the values of both the layer thickness and road width are of continuous domains. According to other FDM machines, such as Stratasys Prodigy Plus, those values are of discrete domains (Stratasys inc., 2004). The values of the layer thickness are either fine (0.178mm), standard (0.245mm), or draft (0.330mm). The values of the road width are thin (0.333mm), standard (0.511mm), or wide (0.706mm). In order to accommodate for various types of machines, for the sake of completeness of the research work, the toolbox was changed to accommodate the input values of those parameters as discrete instead of continuous to be compared to the outcomes of the continuous domain optimization problem. Therefore, the general formulation of the utility function in hand would be as demonstrated in equation (6.1):

$$\text{Minimize UF} = 0.33 * ACH + 0.33 * VS + 0.17 * T + 0.17 * AD \qquad (6.1)$$

Subject to:

$\text{Max.}(C) \leq C_{all.}$

$Or = \{1,2,3,\ldots N_{or.}\}$

$L = \{0.178mm, 0.245mm, 0.330mm\}$

$Rw = \{0.333mm, 0.511mm, 0.706mm\}$

Using the same optimization and GA settings and running the algorithm with the new input values, the optimal utility function value obtained by the developed optimization algorithm is 0.2103107 while the near optimal results for the process parameters and the corresponding process characteristics are shown in Tables 6.3 and 6.4 respectively.

Table 6.3: Near optimal process parameter values for discrete input values

Process Parameter	**Near Optimal Value**
Orientation	1
Layer Thickness	0.245mm
Road Width	0.511mm

Table 6.4: Optimal process characteristics results for discrete input values

Surface Roughness	**Support Structure Volume**	**Build Time**	**Dimensional Accuracy**
ACH_{min} =0.022 mm	V_{env} =1002903 mm^3	T_{min} =8.9721 h	AD_{min} =0.1487 mm
ACH_{max} =0.102 mm	V_{solid} =418173 mm^3	T_{max} =11.4481 h	AD_{max} =0.1565 mm
$ACH_{opt.}$ =0.031 mm	$VS_{opt.}$ =27192 mm^3	$T_{opt.}$ =10.2824 h	$AD_{opt.}$ =0.1520 mm
$ACH_{opt.\ norm.}$ =0.1050	$VS_{opt.\ norm.}$ =0.0465	$T_{opt.\ norm.}$ =0.5292	$AD_{opt.\ norm.}$ =0.4138
Maximum Cusp Height = 0.1024 mm			

When comparing the results of using continuous and discrete input values, the difference in the results were not very obvious, since coincidentally the middle parameter values of the discrete domain were very close to the optimal values obtained by the continuous parameters optimization. Most of the process characteristics were very close if not the same. Since the constraints of the mixed utility function are more flexible, the outcomes using the mixed multi-objective problem gave better results for almost every build objective, although from the tables it was noted that the build time in the discrete input values was better than that of the continuous. It is possible to get better results for one specific build objective if optimized separately, but this thesis is concerned with the optimal results noted using the multi-objective problem.

CHAPTER 7

VISUALIZATION & VIRTUAL VALIDATION OF RESULTS

This chapter demonstrates the outcomes of using the virtual rapid prototyping software (VisCAM RP) for the purpose of visualizing and virtually validating the optimized results.

7.1 Visualization of Results

As mentioned earlier in chapter 6, a model was built using I-DEAS software to test the developed code, and the near optimal results of applying the code to the model were noted. To visualize and validate these results using the VRP software, several steps were taken (Appendix C explains these steps in details and gives a brief description of the capabilities of the VisCAM RP software). This section will concentrate on the visualization outcomes of the software.

Once the model was imported to the software, as shown in Figure 7.1, the designer has the ability to: (a) move and rotate the part to see it from any angle or position, (b) zoom in and out freely to see specific details, (c) select different orientations to build the part, (d) take measurements, (e) look at cross sections at any level, (f) view the model data as wire frame or solid, and more. All these options, shown in Figure 7.2, allow the designer to visualize the model even before generating the slices.

With these visualization capabilities, designers will save a lot of time wasted on machine setup, part fabrication, and physical measurements. Using the software gives the designer many advantages over traditional experimentation, as it allows the user to look at the part from angles and positions almost impossible for the naked eye to see,

especially when using the zoom or the cross section options and also taking accurate measurements of very fine details.

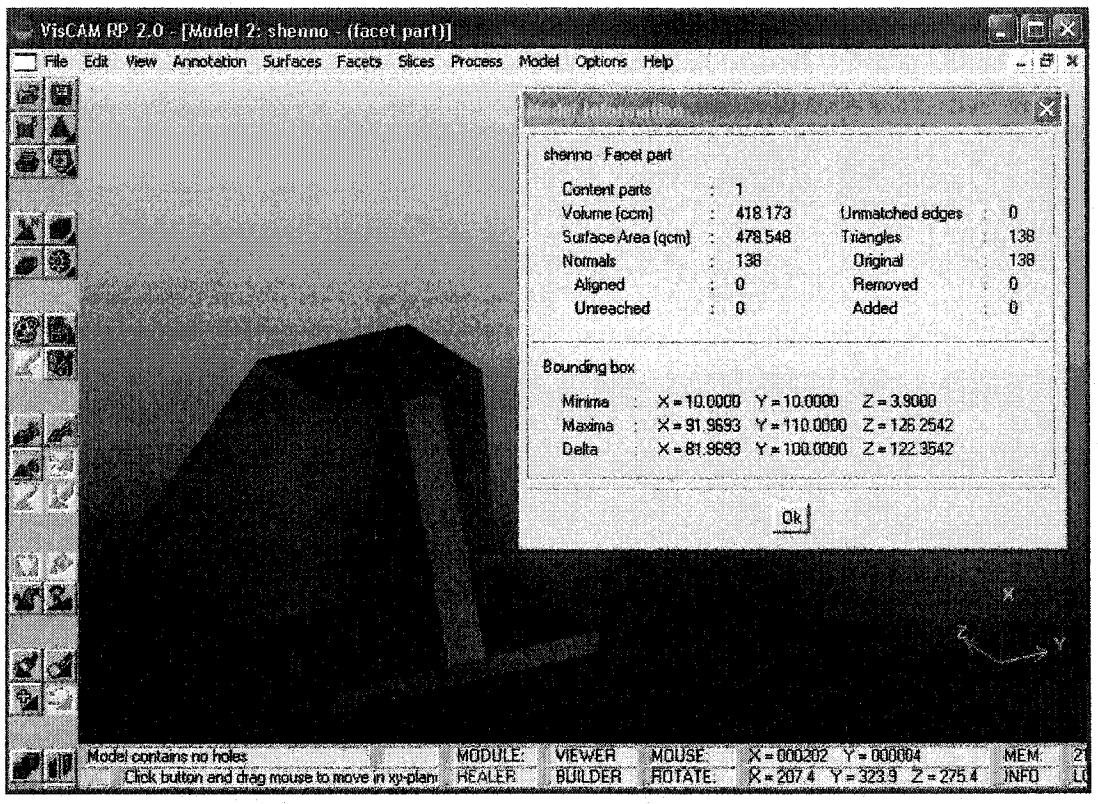


Figure (7.1) Case study imported to VisCAM RP

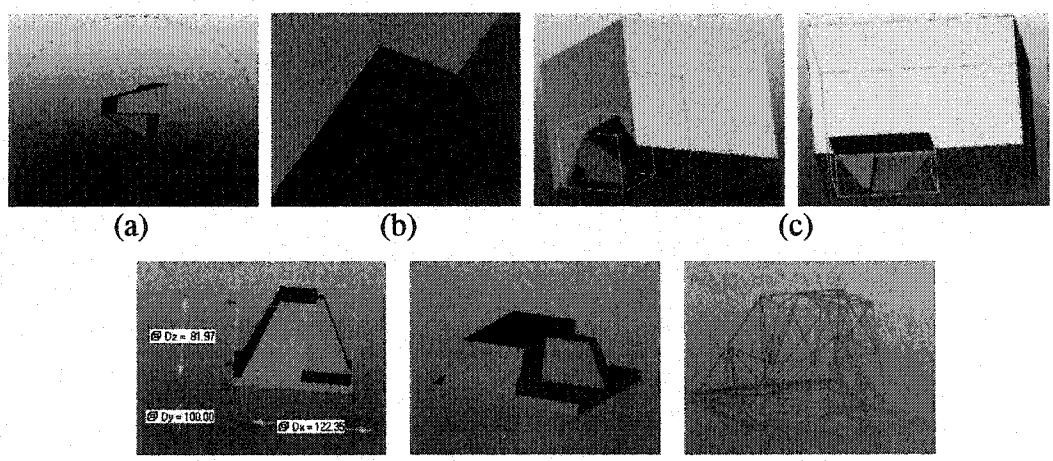


Figure (7.2) VisCAM RP visualization options

7.2 Preparing for Validation

After the desired RP machine has been defined for the building process, the orientation was set and then the slices were generated. In our optimization problem the optimal orientation was orientation 1, demonstrated in Figure 7.3.

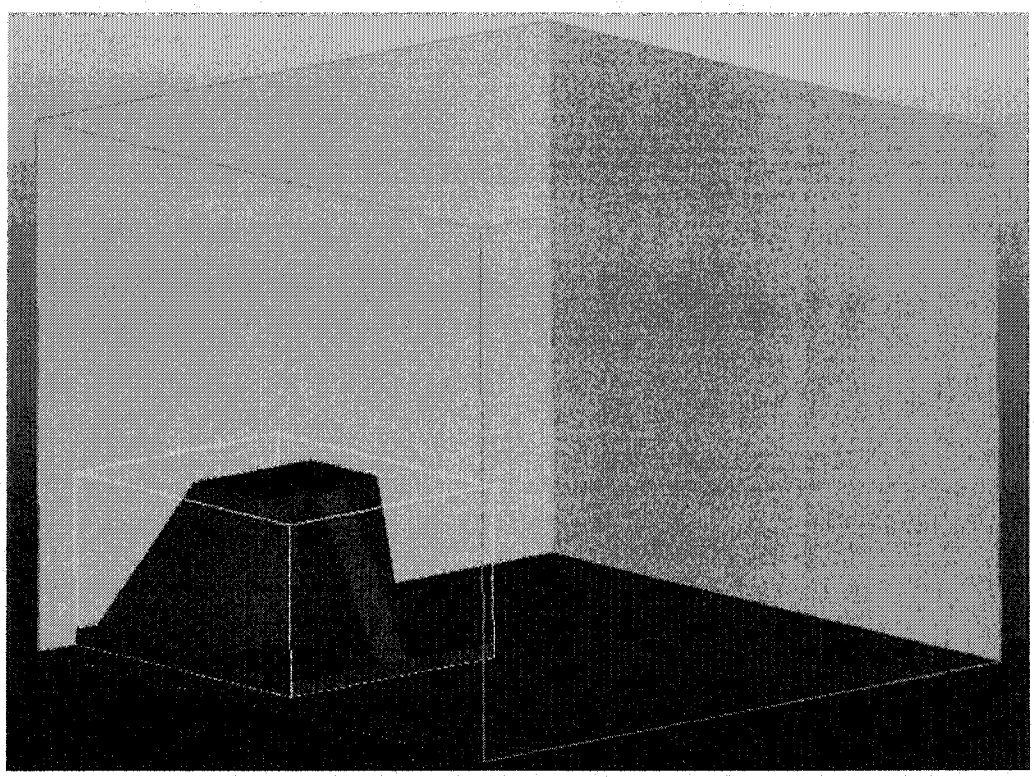


Figure (7.3) Orientation 1 in the workplace

Using the generate slice option (see Appendix C); the layer thickness was set to 0.239mm, which is the value obtained from the optimization algorithm. The software immediately slices the part into layers of 0.239mm each, as seen in Figure 7.4, which virtually demonstrates the actual appearance of the model as it would be fabricated in the physical process.

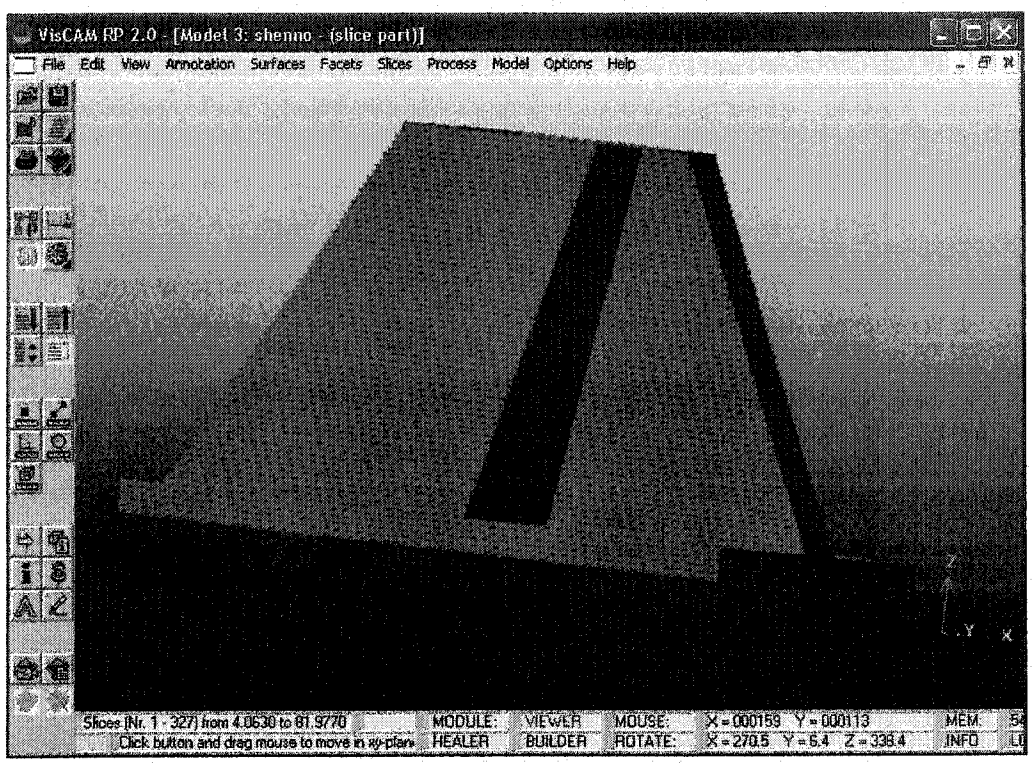


Figure (7.4) Stair-stepping effect on model after generating slices

Based on the near optimal results, the orientation and layer thickness were set. The next step was to generate the hatches or roads to the desired space or width. By selecting the generate hatch option (see Appendix C), the hatch style for the slice building process was defined to match the optimal result from the optimization problem. From the optimization results the optimal road width value was 0.503mm. After all the required parameters have been set, the validation phase was ready.

7.3 Validation of Results

The validation phase was intended to validate the optimal values of the selected objective functions: (a) Average Cusp Height, (b) Support Structure Volume, (c) Build Time, and (d) Absolute Average Deviation.

7.3.1 Average Cusp Height

The zooming capability of the software allowed us to take a close look at the layers, and measure the distance between the layer edges, as demonstrated in Figure 7.5. Having this distance, as well as the layer thickness, we were able to calculate the cusp heights for

each inclined surface and substantially calculate the average cusp height of the model at its optimal orientation. The average cusp height value calculated using the software capabilities was 0.025mm. The average cusp height value obtained from our optimization problem was 0.030mm.

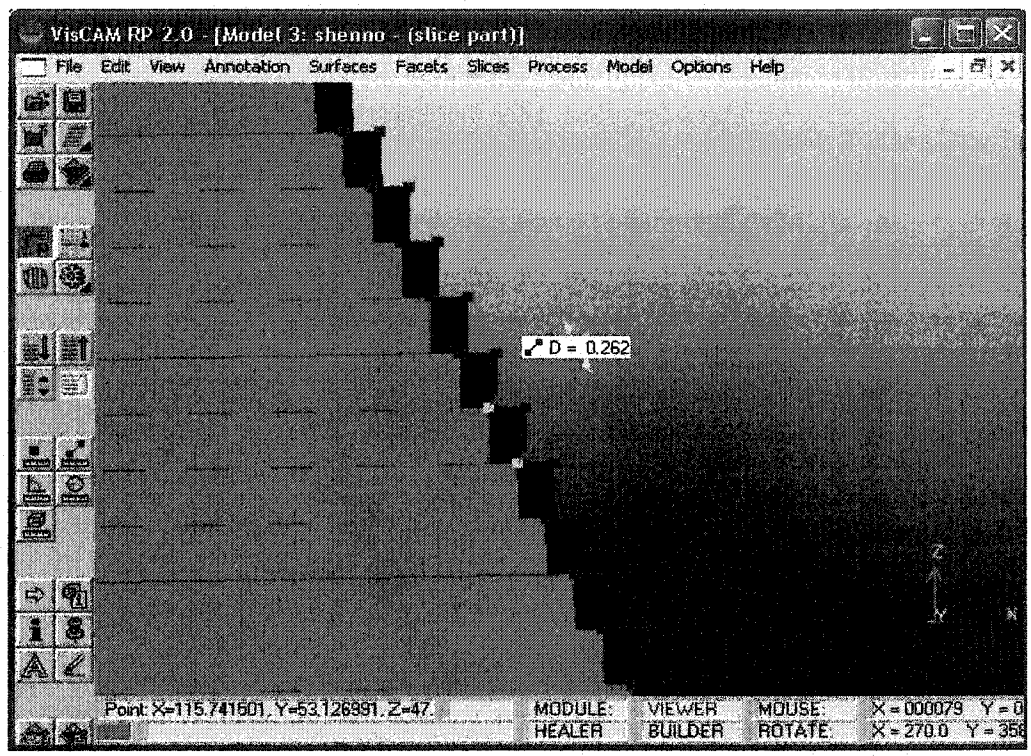


Figure (7.5) Distance measured between layer edges

When investigating the reason for the difference in results, we found out that the distance measured between most layer corners are not a perfect indication to calculate the cusp height. In case the inclined surface was adjacent to a surface perpendicular to the x-axis, then the distance between the layer corners would be indicative. Whereas, in case the inclined layer was adjacent to another inclined layer, then the distance between the layer corners would not be indicative due to the shifted effect illustrated in Figure 7.6.

Since the software does not allow the user to freely measure distances from any point on the model, the calculated average cusp height had a minor difference when compared to the optimal value obtained from the optimization problem. On the other hand, the

value was relatively close which still indicates the validity of using the virtual rapid prototyping tool for validating the average cusp height.

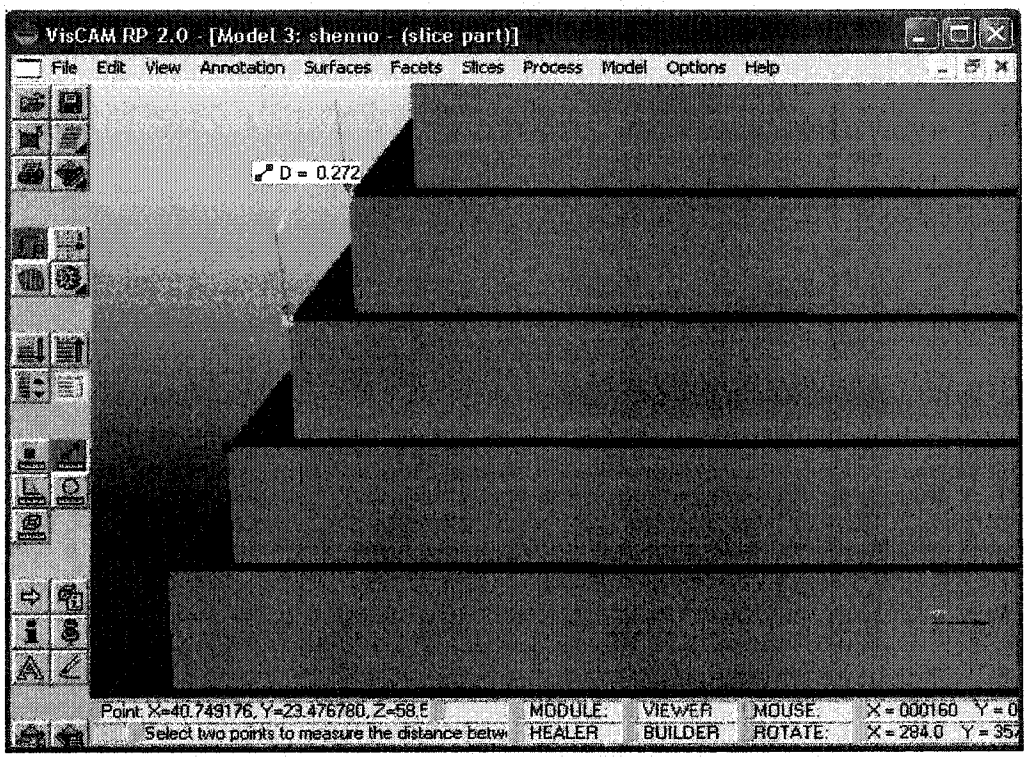


Figure (7.6) Shifted effect in measuring distance between layer corners

To confirm the validity of this result, the cusp height was calculated for an inclined surface adjacent to one that is perpendicular to the x-axis and compared to the value obtained using the software capabilities for the same surface. The value of the cusp height for Face 4, (see Appendix C), using the software capabilities was 0.0979mm. When calculating the cusp height value of the same surface using the data from the model built on I-DEAS, the value turned out to be 0.0984mm. Therefore both values are approximately 0.098mm, which demonstrates that the VRP software was capable of validating the cusp height value.

The small difference in the values is probably due to the slight deviation occurred after the model was sliced into layers, which is a true representation of the deviation that occurs in the practical physical rapid prototyping process. This deviation, although causes differences in results when compared to the original model designed on I-DEAS,

represents how the part will actually look like after it is fabricated on a rapid prototyping machine. This demonstrates the accuracy of using virtual rapid prototyping as an indicative tool for analyzing and testing RP parts

As for the other objectives, the software did not include an option for measuring the absolute average deviation therefore it was excluded in the validation phase. Regarding the build time, the model chosen in our optimization problem did not include all the details the software needed to calculate the total build time. Aside from just the layer thickness and the road width the parameters required were: contour speed, support speed, hatch speed, idle speed and recoating time. Those values could have been assumed, but then would not be an accurate means for validation. The support structure volume generated via the software conforms to the parameters defined and integrated in the machine database. In the optimization algorithm developed, the support structure volume was calculated by multiplying the projected area of the inclined surface by the average height of the inclined surface. Therefore, using the capabilities of the software to validate the outcomes for this objective would not be a valid indication for correct validation.

Thus, the only function we were able to validate using this particular tool, at this stage, was the surface roughness or average cusp height. The validation of the other three objectives is highly recommended after further investigation in future research work.

CHAPTER 8

CONCLUSIONS

8.1 Conclusions

The optimization of rapid prototyping process parameters has been tackled by various researchers, as reported in the published literature review. Different problems were examined along with different RP technologies. Some of the work considered the selection of a specific process parameter for optimizing more than one build objective were others dealt with several decision variables towards optimizing a single build objective, but far less research considered the multi-objective optimization of a combination of different objectives with respect to numerous variables.

The research work presented in this thesis addressed the optimization and visualization of rapid prototyping process parameters. A multi-objective model was generated and a tool was built for selecting near optimal values for the most crucial rapid prototyping process parameters. The outcomes of the optimization tool were validated, using VisCAM RP.

The following concluding remarks can be pointed out to the presented research:

1. The use of a multi-objective optimization method, such as weighted sum method, was essential for the problem at hand, which consisted of several build objectives that were functions of crucial process parameters.

2. The development of the mixed GA code ensured arrival of the utility function at near global optimum, which was characterized by multi-modal behavior due to the presence of multi-objectives.

3. Visualization using a VRP tool can assist RP designers with many capabilities that will subsequently save a lot of time wasted on machine setup, part fabrication, and physical measurements. Using the software gives the designer many advantages over traditional experimentation, as it allows the user to look at the part from different perspectives almost impossible for the naked eye to see.

4. Although virtual validation was only applied to one objective function, due to a limitation in the software used, but it still demonstrated the advantage of utilizing virtual rapid prototyping to validate the process characteristics obtained from the optimization problem.

Therefore, the thesis demonstrated how using a virtual rapid prototyping tool for the purpose of visualization and virtual validation of the process characteristics due to the optimally selected process parameters using a virtual rapid prototyping system can be considered a powerful tool that will significantly assist designers with an advantage over traditional RP experimentation, due to the several physical iterations performed until desired outcome is reached. The proposed software allowed us to look at a part from different perspectives not possible in the physical world, which is considered a great advantage and provides designers with a valuable tool in rapid prototyping analyses.

A number of issues, which might provide future research topics, can be drawn from the presented thesis. These include:

1. Using a more indicative model for build time, that includes all possible process parameters that affect the rapid prototyping process, like: contour speed, support speed, hatch speed, idle speed and recoating time, etc.

2. Developing an optimization algorithm for the use of adaptive slicing technology instead of uniform slicing.

3. Further investigation for analytical models for build time and support structure volume instead of using the adopted empirical models.

4. Combining both physical experimentation and virtual rapid prototyping as means of concurrent analyses.

5. Developing a virtual rapid prototyping tool, that is more flexible and enables more options for the virtual validation process. Capabilities that might provide more flexibility could include an option for measuring points anywhere on the model rather than specified nodes, calculating absolute average deviation, selecting the parameters to be included in calculating the build objectives, etc.

6. Involving different objectives in the virtual validation as further investigation in future work.

7. Using the comparison tables in the literature review, further investigation could include optimization and validation of other RP technologies such as, SLA and SLS, etc.

8. Further investigation of using different weights for the objective functions or other optimization settings for the genetic algorithm.

REFERENCES

Ahn, S.-H., Montero, M., Odell, D., Roundy, S. and Wright, P.K. (2002), "Anisotropic Material Properties of Fused Deposition Modeling ABS", Rapid Prototyping, Vol. 8, No. 4, pp. 248-257.

Ami, R.K. and Gupta, S.K. (1999), "Manufacturability Analysis for Solid Freeform Fabrication", Proceedings of DETC 1999: ASME Design Engineering Technical Conferences, Sept. 12-15, 1999, Las Vegas, Nevada, pp. 1-12.

Allen, J.K., Rosen, D.W. and Mistree, F., "An Approach to Designing Sustainable Enterprise Systems"

Cambell, R.I., Martorelli, M. and Lee, H.S. (2002), "Surface Roughness Visualisation for Rapid Prototyping Models", Computer-Aided Design, Vol. 34, pp. 717-125.

Cheng, W., Fuh, J.Y.H., Nee, A.Y.C., Wong, Y.S., Loh, H.T. and Miyazawa, T. (1995), "Multi-Objective Optimization of Part-Building Orientation in Steriolithography", Rapid Prototyping Journal, Vol. 1, No. 4, pp. 12-23.

Choi, S. H. and Samavedam, S. (2001), "Visualization of Rapid Prototyping", Rapid Prototyping Journal, Vol. 7, No. 2, pp. 99-114.

Choi, S. H. and Samavedam, S. (2002a), "Modeling and Optimization of Rapid Prototyping", Computers in Industry, Vol. 47, pp. 39-53.

Choi, S. H. and Kwok, K.T. (2002b), "A Tolerant Slicing Algorithm for Layered Manufacturing", Rapid Prototyping Journal, Vol. 8, No. 3, pp. 161-179.

Choi, S. H. and Chan, A. M. M. (2003), "A Layered-Based Virtual Prototyping System for Product Development", Computers in Industry, Vol. 51, pp. 237-256.

Chua, C. K., The S. H. and Gay, R.K.L. (1999), "Rapid Prototyping Versus Virtual Prototyping in Product Design and Manufacturing", International Journal of Advanced Manufacturing Technology, Vol. 15, pp. 597-603.

Degarmo, E.P., Black, T.T. and Kosher, R.A. (2003), Materials and Processes in Manufacturing, John Wiley & Sons, Inc, USA, 2003.

ElMaraghy, H.A., ElMaraghy, W.H. and Nassef, A.O. (2003), "Dimensional Adjustment for Assemblability of Rapid Prototyping Parts", ASME Design Engineering Technical Conference, Vol. 2A, pp. 187-195.

Fadel, G. and Kirschman, C. (1996), "Accuracy Issues in CAD to RP Translations", Rapid Prototyping Journal, Vol. 2, No. 2, pp. 4-17.

Gen, M. and Cheng, R. (1997), Genetic Algorithms and Engineering Design, John Wiley and Sons, New York, 1997.

Giannatsis, J., Dedoussis, V. and Laios, L. (2001), "A Study of the Build-Time Estimation Problem for Stereolithography Systems", Robotics and Computer Integrated Manufacturing, Vol. 17, pp. 295-304.

Gibson, I., Brown, D., Cobb, S. and Eastgate, R. (1993), "Virtual Reality and Rapid Prototyping", Virtual Reality in Engineering, pp. 51-63.

Han, W., Jafari, M.A. and Seyed, K. (2003), "Process Speeding Up Via Deposition Planning in Fused Deposition-Based Layered Manufacturing Processes", Rapid Prototyping Journal, Vol. 9, No. 4, pp. 212-218.

Han, W., Jafari, M.A., Danforth S.C. and Safari, A. (2002), "Tool Path-Based Deposition Planning in Fused Deposition Processes", Transactions of the ASME, Vol. 124, pp. 462-472.

Hope, R.L., Jacobs, P.A. and Roth, R.N. (1997a), "Rapid Prototyping with Sloping Surfaces", Rapid Prototyping Journal, Vol. 3, No. 1, pp. 12-19.

Hope, R.L., Roth, R.N., and Jacobs, P.A. (1997b), "Adaptive Slicing with Sloping Layer Surfaces", Rapid Prototyping Journal, Vol. 3, No. 3, pp. 89-98.

Hu, Z., Lee, K. and Hur, J. (2002), "Determination of Optimal Build Orientation for Hybrid Rapid-Prototyping", Journal of Materials Processing Technology, Vol. 131, pp. 378-383.

Hur, J. and Lee, K. (1998), "The Development of a CAD Environment to Determine the Preferred Build-Up Direction for Layered Manufacturing", International Journal of Advanced Manufacturing Technology, Vol. 14, pp. 247-254.

Hur, S.-M., Choi, K.-H., Lee, S.-H. and Chang, P.-K. (2001), "Determination of Fabricating Orientation and Packing in SLS Process", Journal of Materials Processing Technology, Vol. 112, pp. 236-243.

Jee, H. J. and Sachs, E. (2000), "A Visual Simulation Technique for 3D Printing", Advances in Engineering Software, Vol. 31, pp. 97-106.

Jeng, J.Y., Wong, Y.-S. and Li, Y.-H. (2000), "A new Practical Adaptive Slicing Method for the SLS RP System to Accelerate Fabrication Speed without Accuracy Sacrifice", Journal of the Chinese Society of Mechanical Engineers, Vol. 21, No. 4, pp. 351-358.

Jirathearanat, S, Vazquez, V., Rodriguez, C.A. and Altan, T. (2000), "Virtual Processing-Application of Rapid Prototyping for Visualization of Metal Forming Process", Journal of Materials Processing Technology, Vol. 98, pp. 116-124.

Kulkarni, P., Marsan, A. and Dutta, D. (2000), "A Review of Process Planning Techniques in Layered Manufacturing", Rapid Prototyping Journal, Vol. 6, No. 1, pp. 18-35.

Lan, P.-T., Chou, S.-Y., Chen, L.-L. and Gemmill, D. (1997), "Determining Fabrication Orientations for Rapid Prototyping with Steriolithography Apparatus", Computer-Aided Design, Vol. 29, No. 1, pp. 53-62.

Laperiere, L., ElBacha, T. and Desrochers, A. (2001), "Functional Assemblies Using Rapid Prototyping".

Lee, K.H. and Choi, K. (2000), "Generating Optimal Slice Data for Layered Manufacturing", The International Journal of Advanced Manufacturing Technology, Vol. 16, pp. 277-284.

Levy, G.N., Schindel, R. and Kruth, J.P. (2003), "Rapid Manufacturing and Rapid Tooling with Layer Manufacturing (LM) Technologies, State of the Art and Future Perspectives", Annals of the CIRP, keynote papers, Vol. 52, No. 2, pp. 1-21

Lin, F., Sun, W. and Yan, Y. (2001), "A Decomposition Accumulation Model for Layered Manufacturing Fabrication", Rapid Prototyping Journal, Vol. 7, No. 1, pp. 24-31.

Lin, F., Sun, W. and Yan, Y. (2001), "Optimization with Minimum Process Error for Layered Manufacturing Fabrication", Rapid Prototyping Journal, Vol. 7, No. 2, pp. 73-81.

Lu, L., Fuh, J. and Wong, Y.S. (2001), Laser-Induced Materials and Processed for Rapid Prototyping, Kluwer Academic Publishers, USA, 2001.

Luo, R.C., Chang, Y.C. and Tzou, J.H. (2001), "The Development of a New Adaptive Slicing Algorithm for Layered Manufacturing System", Proceedings of the 2001 IEEE, International Conference on Robotics and Automation, May 21-26, Seoul, Korea, pp. 1334-1339.

Lynn-Charney, C. and Rosen, D.W. (2000), "Usage of Accuracy Models in Stereolothography", Rapid Prototyping Journal, Vol. 6, No. 2, pp. 77-86.

Ma, W. and He, P. (1999), "An adaptive Slicing and Selective Hatching Strategy for Layered Manufacturing", Journal of Material Processing Technology, Vol. 89-90, pp. 191-197.

Marler, R. T. and Arora, J. S. (2004), "Survey of Multi-Objective Optimization Methods for Engineering", Review Article for Structural Multidisciplinary Optimization Journal, Vol. 26, pp. 369-395.

McClurkin, J. E. and Rosen, D.W. (1998), "Computer-Aided Build Style Decision Support for Stereolithography", Rapid Prototyping Journal, Vol. 4, No. 1, pp. 4-13.

McMains, S., Smith, J., Wang, J. and Sequin, C. (2000), "Layered Manufacturing of Thin-Walled Parts", Proceedings of DETC 00: ASME Design Engineering Technical Conferences Sept. 10-13, 2000, Baltimore, Maryland, pp. 1-9.

Michalewicz, Z. (1996), Genetic Algorithms +Data Structures =Evolution Programs, Springer-Verlag, Berlin, 1996.

Onuh, S.O. and Hon, K.K.B. (1998), "An Experimental Investigation into the Effect of Hatch Pattern in Stereolithography", Annals of the CIRP, Vol. 47, No. 1, pp. 157-160.

Pandey, P.M., Reddy, N.V. and Dhande, S.G. (2003), "Slicing Procedures in Layered Manufacturing: a Review", Rapid prototyping Journal, Vol. 9, No. 5, pp. 274-288.

Perez, C.J.L., Calvet, J.C. and Perez, M.A.S. (2001a), "Geometric Roughness Analysis in Solid Free-Form Manufacturing Process", Journal of Materials Processing Technology, Vol. 119, pp.52-57

Perez, C.J.L., Sebastian, M.A. (2001b), "Surface Roughness Analysis in Layered Forming Process", Journal of the International Societies for Precision Engineering and Nanotechnology", Vol. 25, pp. 1-12.

Reeves, P.E. and Cobb, R.C. (1997), "Reducing the Surface Deviation of Stereolithography Using In-Process Techniques", Rapid Prototyping Journal, Vol. 3, No. 1, pp. 20-31.

Qiu, D., Langrana, N.A., Danforth, S.C., Safari, A. and Jafari, M. (2001), "Intelligent Toolpath for Extrusion-Based LM Process", Rapid Prototyping Journal, Vol. 7, No. 1, pp. 18-23.

Qiu, D. and Langrana, N.A. (2002), "Void Eliminating Toolpath for extrusion based Mulit-Material Layered Manufacturing", Rapid Prototyping Journal, Vol. 8, No. 1, pp. 38-45.

Tata, K., Fadel, G., Bagchi, A. and Aziz, N. (1998), "Efficient Slicing for Layered Manufacturing", Rapid Prototyping Journal, Vol. 4, No. 4, pp. 151-167.

Tong, K., Lehtihet, E.A. and Joshi, S., "Parametric Error Modeling and Software Error Compensation for Rapid Prototyping", Rapid Prototyping Journal, Vol. 9, No. 5, pp. 301-313.

Tyberg, J. and Bohn, J.H.. (1998), "Local Adaptive Slicing", Rapid Prototyping Journal, Vol. 4, No. 3, pp. 118-127.

Vasudevarao, B., Natarajan, D.P. and Henderson, M., "Sensitivity of RP Surface Finish to Process Parameter Variation"

Williams, J.D. and Deckard C.R. (1998), "Advances in Modeling the effects of Selected Parameters on the SLS Process", Rapid Prototyping Journal, Vol. 4, No. 2, pp. 90-100.

Williams, C.B., Panchal, J.H., and Rosen, D.W. (2003), "A General Decision-Making Method for the Rapid Manufacturing of Customized Parts", Proceedings of DETC 03: ASME Design Engineering Technical Conferences Sept. 2-6, 2000, Chicago, Illinois, pp. 1-9.

Wohlers, T. (1996), "Rapid Guide to Rapid Prototyping", publication developed by Wohlers Associates and made available courtesy of Stratasys Inc.

Wu, T. (2001), "Integration and Visualization of Rapid Prototyping and Reverse Engineering", M.Sc. Thesis, University of Windsor.

Xu, F., Loh, H.T. and Wong, Y.S. (1999), "Considerations and Selection of Optimal Orientation for Different Rapid Prototyping Systems", Rapid Prototyping Journal, Vol. 5, No. 2, pp. 54-60.

Xu, F., Wong, Y.S., Loh, H.T. and Miyazawa, T. (1997), "Optimal Orientation with Variable Slicing in Stereolithography", Rapid Prototyping Journal, Vol. 3, No. 3, pp. 76-88.

Yan, X. and Gu, P. (1996), "A Review of Rapid Prototyping Technologies and Systems", Computer Aided Design, Vol. 28, No.9, pp. 63-4.

Yang, Y., Loh, H.T., Fuh, J.Y.H. and Wang, Y.G. (2002), "Equidistant Path Generation for impriving Scanning Efficiency in Layered Manufacturing", Rapid Prototyping Journal, Vol. 8, No. 1, pp. 30-37.

Zhang, L.-C., Han, M. and Huang, S.-H. (2002), "An Effective Error-Tolerance Slicing Algorithm for STL Files", The International Journal of Advanced Manufacturing Technology, Vol. 20, pp. 363-367.

Zhou, J.G., Herscovici, D. and Chen, C.C. (2000), "Parametric Process Optimization to Improve the Accuracy of Rapid Prototyped Stereolithography Parts", International Journal of Machine Tools and Manufacturing, Vol. 40, pp. 363-379.

Zhong, X. (2003), "A Data Exchange System in E-Manufacturing", M.Sc. Thesis, University of Windsor.

Zhou, M.Y., Xi, J.T. and Yan, J.Q. (2004), "Adaptive Direct Slicing with Non-Uniform Cusp Heights for Rapid Prototyping", International Journal of Advanced Manufacturing Technology, Vol. 23, pp. 20-27.

Zieman, C.W. and Crawn, P.M. (2001), "Computer Aided Decision Support for Fused Deposition Modeling", Rapid Prototyping Journal, Vol. 7, No. 3, pp. 138-147.

Zorriassatine, F., Wykes, C., Parkin, R. and Gindy, N. (2003), "A Survey of Virtual Prototyping Techniques for Mechanical Product Development", Institute of Mechanical Engineers, Vol. 217, Part B.

APPENDIX A

BINARY GENETIC ALGORITHMS

A.1 Coding

Given a function $F(x_1, x_2, ..., x_n)$ to be optimized where each variable x_i ranges between two extremes L_i and U_i, $i \in \{1,2,.....,n\}$

If x_i is to be discretized into a set of discrete values using binary coding, then an array of binary digits of length l is defined, as shown in Figure A1

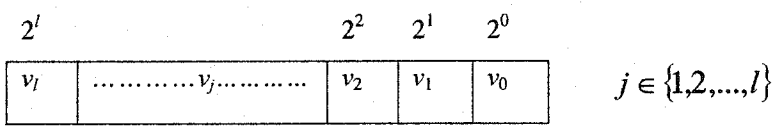


Figure (A1) Binary String for Discretized Variable

The array shown in Figure A1 is filled with values v_j where v_j =0 or 1

The real value of the array is equal to;

$$R_i = L_i + \sum_{j=0}^{l} \delta 2^j$$

where, δ is the discretization increment (the difference between each successive pair of discrete values).

If all elements of the array are equal to zero then $R_i = L_i$ and if all elements of the array are equal to one, then R_i should be equal to U_i

i.e.; $L_i + \sum_{j=0}^{l} \delta 2^j = U_i$

The above equation can be used to estimate the length 'l' of the array:

$$\sum_{j=0}^{l} 2^j = \frac{(U_i - L_i)}{\delta}$$

$$2^{l+1} - 1 = \frac{(U_i - L_i)}{\delta}$$

$$l = \log_2[\frac{U_i - L_i}{\delta} + 1] - 1$$

Since each variable has its own array, the set of arrays for all variables form another array known in the genetic algorithms literature as *chromosome* (in some cases known also as agent, individual or string). Each location in the chromosome array is known as a *gene*, and the value it assumes (0 or 1) is known as *allele*.

A.1.1 Initialization of a Population

The main data structure on which a GA operates is the population of chromosomes. Each chromosome corresponds to a solution point in the space of the independent variables. The integer N is known as the population size

x_1	x_2	.....	x_n	
				Chromosome 1 ($\underline{X}_1$)
				Chromosome 2 ($\underline{X}_2$)
. .. . .		.....		
				Chromosome N ($\underline{X}_N$)

Figure (A2) A Population of Chromosomes

Once the population matrix is constructed as shown in Figure A2, the population is filled at random with zeros and ones (i.e. each gene assumes a value zero or one drawn randomly). The population undergoes several changes through the iterative application of genetic operators (described in the forthcoming sections) until it settles at a near-global

optimum solution. The following section describes the general genetic algorithm and the subsequent sections detail the used operators.

A.1.2 The General GA Procedure

Given below is a brief pseudo-code of the general GA (Michalewicz, 1996)

Procedure: General GA

Step 0: -Initialize the initial population

-Set generations counter $G=1$

Step 1: - (Generations loop)

1.1 : For each chromosome $\underline{X}_k$, $k \in \{1,....,N\}$ evaluate $F_k(\underline{X}_k)$, where F_k is the objective function value, of the k^{th} chromosome.

1.2 Convert the objective function value into fitness value f_k such that optimization is converted into fitness maximization problem.

1.3 Apply the selection operator, and copy (probabilistically) the high fitness chromosomes to a temporary population. (Survival of the fittest)

1.4 Select (probabilistically) pairs of chromosomes to apply the cross-over operator.

1.5 Select (probabilistically) chromosomes to apply the mutation operator.

1.6 Replace the population by the temporary population.

Step 2: (End the generations loop)

If $G < G_{max}$ (a pre-specified number of generations)

let $G = G + 1$, and Goto step 1

Else Deliver the chromosome with the highest fitness as the problem's solution.

End Procedure.

The above algorithm shows that the operation of genetic algorithm consists of a loop of steps applied to a population of search points (chromosomes). This contrasts the

action of the traditional gradient based and direct search methods which depend on applying successive moves to a single search point. Step 1.3 is responsible for selecting the fittest chromosomes for the new population, while steps 1.4 and 1.5 are responsible for generating new solution points (chromosomes) from the selected ones.

In the following section the main genetic operators in the above algorithm are described. The main genetic operators are:

(1) Selection

(2) Cross-over

(3) Mutation.

A.2 The Genetic Operators

A.2.1 Selection

This operator is responsible for the repetition of the high fitness chromosomes. The fitness of each chromosome is a measure of its importance relative to the objective function. An example of a fitness function is shown below.

Given a chromosome $\underline{X}_k$, $k \in \{1,....,N\}$

$f_k(\underline{X}_k) = F(\underline{X}_k)$ in the case of maximization

$f_k(\underline{X}_k) = \max(F(\underline{X})) - F(\underline{X}_k)$ in the case of minimization

As shown above, the minimization problem is turned into a maximization one.

The fitness value is then used to obtain a probability value p_k associated with each chromosome.

$$p_k = \frac{f_k}{\sum_{l=1}^{N} f_l} \tag{5.4}$$

The cumulative probability P_k is evaluated after sorting the population in an ascending order according to the fitness value. Hence P_k is scaled from zero to one.

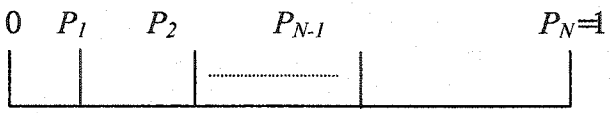


Figure (A3) Representation of Cumulative Probability

When the cumulative probability is scaled down, a random number α is generated from a uniform distribution between zero and one, and if α falls between P_{k-1} and P_k, then chromosome $\underline{X}_k$ is copied to the temporary population. This step is repeated N times. The procedure of evaluating the cumulative probability and generating the new temporary population is known as the *roulette wheel selection*.

A.2.2 Cross-Over

Cross-over is an operator used for the generation of new chromosomes (solutions) by emulating the same operator in genetics. The algorithm for the cross-over operator is as follows:

Procedure: Cross-Over for Binary GAs

1. Select two chromosomes randomly $\{\underline{X}_1, \underline{X}_2\}$ for the application of cross-over operator.
2. Generate a random number α , such that $\alpha \in [0,1]$,

If $\alpha <$ cross-over probability (p_c) [typical values for p_c range between 0.5-0.9]

Apply cross-over

-For each variable x_i, find a random location Q, such that $Q \in \{2,3,4,....,l_i\}$, then swap all bits after location Q. (Figure A4).

-Copy the two selected chromosomes to the new population.

Else

-Do not apply cross-over and copy the original chromosomes into the new population

End Procedure

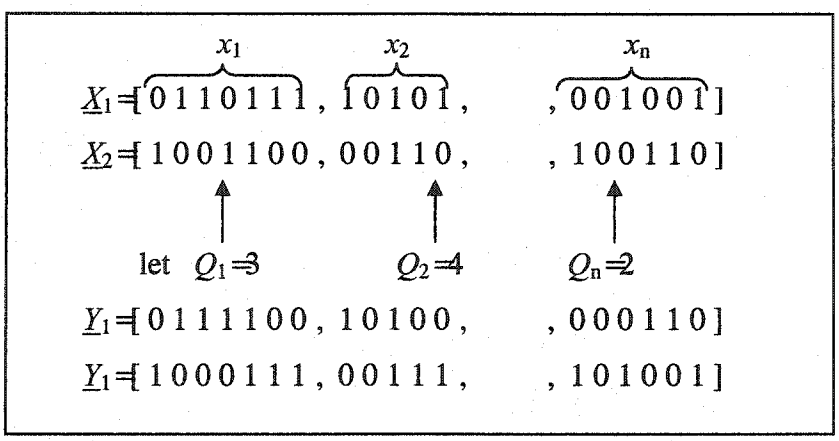


Figure (A4) Example of Cross-Over

A.2.3 Mutation Operator

Mutation is another operator used for generating new solutions. In binary coded genetic algorithms the mutation operator works by finding a random location and flipping the value of the string in that location. The algorithm for the mutation operator is as follows:

Procedure: Mutation for Binary GAs

1. Select a chromosome $\underline{X}$ to apply the mutation
2. Generate a random number β, where $\beta \in [0,1]$,

If $\underline{X}$ < mutation probability (p_m) [Typical values for p_m range between 0.01 - 0.03]

-Given a chromosome of length l_i for each variable x_i, generate a random location Q, where $Q \in \{1,2,....,l_i\}$.

-Reverse the value of the Q^{th} bit from zero to one (if its value is equal to zero) or from one to zero (if its value is equal to one). An example of mutation is shown in Figure A5.

-Copy the chromosome to the new population.

Else

-Do not apply mutation and just copy the original chromosome to the new population.

End If

End Procedure

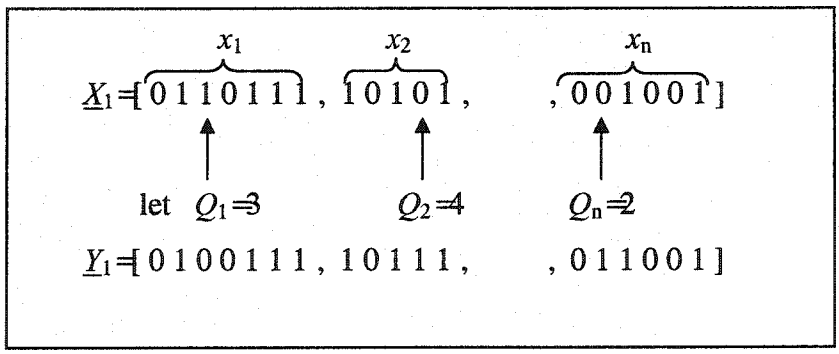


Figure (A5) Example of Mutation

A.3 The Overall Action of Simple Genetic Algorithms

The overall action of a genetic algorithm produces successive populations of chromosomes (candidate solutions). The transition between each pair of consecutive populations is known as a generation. Within a generation a temporary population is needed between the selection and the cross-over and mutation operators as illustrated in Figure A6.

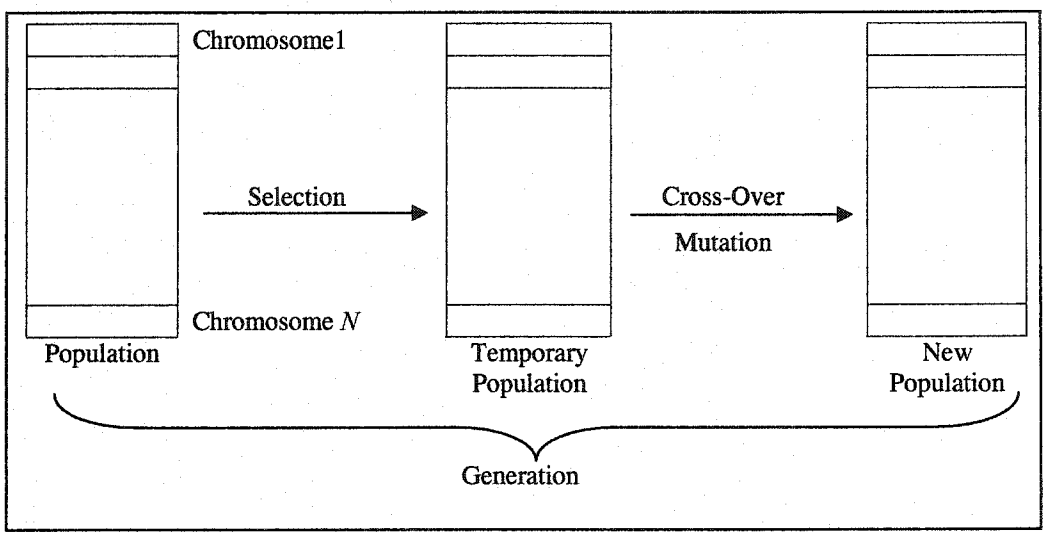


Figure (A6) A Generation

The generation shown in Figure A6 is repeated for a pre-specified number of times (maximum number of generations).

APPENDIX B

CASE STUDY DIMENSIONS

B.1 Model Dimensions

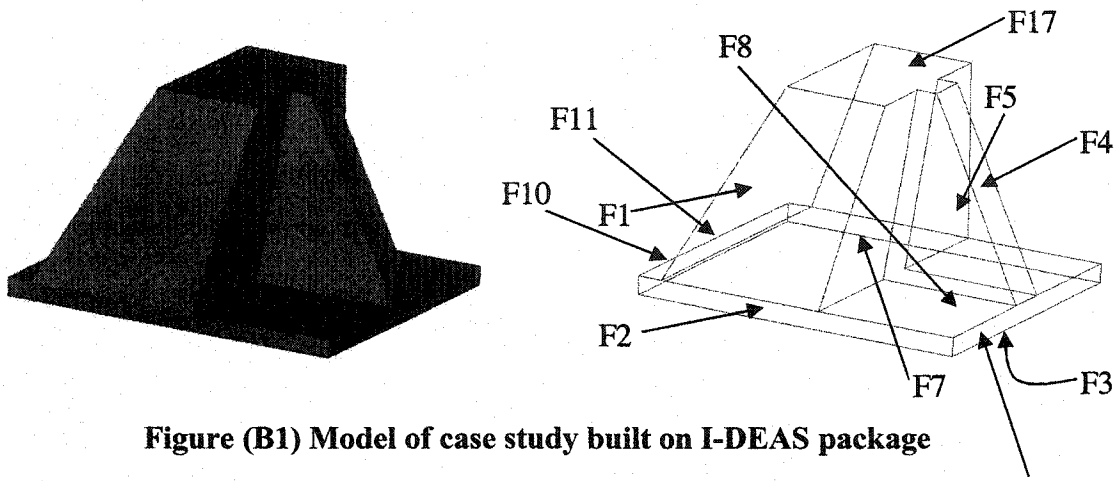


Figure (B1) Model of case study built on I-DEAS package

Total Volume of Part =418,173 mm^3

Solid Surface Area =47,854.8 mm^2

No. of Faces =17 faces

Areas of Faces:

F1 =3,851.191 mm^2
F2 =880.620 mm^2
F3 =12,235.420 mm^2
F4 =1,148.825 mm^2
F5 =2,234.171 mm^2
F6 =2,234.171 mm^2
F7 =2,455.923 mm^2
F8 =2,252.357 mm^2
F9 =6,498.981 mm^2
F10 =879.671 mm^2
F11 =719.731 mm^2
F12 =719.731 mm^2
F13 =880.621 mm^2
F14 =2,455.923 mm^2
F15 =2,252.357 mm^2
F16 =3,851.191 mm^2
F17 =2,303.955 mm^2

APPENDIX C

VISCAM RP USER MANUAL

C.1 VisCAM RP Capabilities

VisCAM RP is an extensive software solution for the complete preparation of CAD/CAM data for Rapid Prototyping processes like Stereo lithography (SL), Laser Sintering (SLS), Fused Deposition Modeling (FDM) or 3D Color Printing (3DP). VisCAM RP supports you in all processing steps from the verification and repair of the CAD/CAM data over the assembly of the build envelope up to the generation of build-ready RP slice files including hatches and supports. Moreover, VisCAM RP is not limited to process only facet data like STL, but offers also the direct processing of CAD surface and/or RP slice data in all common formats.

VisCAM RP is based on a modular and flexible component system for different formats and separate processing steps. All modules can be composed in almost all combinations pursuant to the own needs and infrastructures of the RP user. The customization of VisCAM RP guarantees an optimal integration and support within the existing CAD/CAM process chain of the RP user.

VisCAM Solid Viewer is a freeware 3D-Viewer for the fast visualization and communication of 3D models. VisCAM Solid Viewer imports 3D models from STL (ASCII, Binary, Colored), VRML, PLY, ZCP, DXF (3D-FACE), 3D Studio (3DS) and VisCAM RP (VFX). To support the fast verification of the imported model geometry, unconnected edges, flipped facet normal and individual solids can be shown directly on the model.

Additional model information like dimensions, surface area or the model volume can be retrieved at any time. An extensive set of measuring and annotation functions support the analysis of model details and enrich the model with additional visual information. The

annotated 3D model can be exported as a compressed file within the VisCAM RP format (VFX) and exchanged with all other VisCAM RP products. VisCAM Solid Viewer is the base module of the VisCAM RP software series.

VisCAM RP is based on a modular customization approach and can be used on graded levels. You can combine all modules into an extensive all-in one system or you can setup several coordinated stand-alone systems for individual subtasks.

The flexibility and productivity of VisCAM RP gives you the guarantee for an optimal integration of the system into your existing CAD/CAM process chain.

C.1.1 View and Communicate

VisCAM Solid Viewer is the base module for the fast visualization and verification of 3D models. Visual annotations and measures can easily be added to the model and can be exchanged within the compressed VFX file format. VisCAM Solid Viewer is available as a freeware product and can be used as standard tool for efficient data exchange and communication with all your customers and colleagues.

C.1.2 Place and Calculate

VisCAM Solid Builder is an efficient tool to assemble build jobs and to estimate time and costs for the build job. An integrated database with more than 130 pre-defined RP machines provides you with adjustable settings to arrange and project your build jobs. VisCAM Solid Builder can be used as a separate and cost-effective solution for early order planning and quoting within your sales and distribution department.

C.1.3 Repair and Adjust

VisCAM Solid Healer is a comprehensive tool to repair, edit and manipulate facet files. All detected model errors can be highlighted and automatically corrected. Additionally you can perform interactive fixing or editing and manipulate the model geometries with CAD functions like Booleans or cutting. Triangle reduction or smoothing can be applied to optimize the part quality and file size. The additional module

VisCAM Solid Painter is able to add and process colors on facet files for 3D color printing applications.

The **VisCAM Surface** sub-system contains optional modules to process surface models. Imported IGES or VDA-FS can be repaired, edited and converted to accurate facet files.

C.1.4 Slice and Control

The system can be extended with several modules to generate slices **(Solid Slicer, Surface Slicer)** and to process slice files **(Slice Viewer, Slice Healer)**, Slice Builder).

Detailed time and cost calculations as well as advanced visualization and analysis tools are available to give you the ability to fully control the generated slice files before they are exported to your RP machine.

Optional modules are available to calculate efficient hatch styles for stereo lithography, laser sintering or fused deposition modeling systems **(VisCAM Slice Hatcher)** and to calculate support structures for stereo lithography and laser sintering systems **(VisCAM Slice Supporter)**.

VisCAM RP is the ideal software package for the complete preparation of your CAD/CAM data for any Rapid Prototyping application. The system assists you in all processing steps from the verification and repair of the CAD/CAM data to the generation of build-ready RP slice files including hatches and support structures. Moreover VisCAM RP is not limited to process only facet data like STL, but also offers you the direct processing of CAD surface data and RP slice data at your choice. The flexible system concept together with the full control of your model data at any time makes VisCAM RP the ideal software solution for your RP data preparation.

C.1.5 VisCAM Solid (Solid processing)

VisCAM Solid offers you the preparation of facet data from STL, 3DS, VRML, DXF, PLY, ZCP and VFX files for your RP application. The fast real time viewer

enables you to visualize and verify the imported facet data as well as included color information. Model faults such as unmatched edges, holes and flipped normal can be indicated and corrected easily. Aimed model manipulations can be carried out with the detection and treatment of individual solids, surfaces and facets. Afterwards, accurate slice files can be generated fast and easy.

C.1.6 VisCAM Slice (Slice processing)

VisCAM Slice offers you the preparation of slice data from CLI, SSL, SLC, F&S and STD files for your RP application. The slice data can be visualized and verified completely in 3D or 2D. You can optimize your slice data with the automatic error correction, changeable contour accuracies and a variable layer thickness calculation. Extensive hatch styles and a fast support generation are available to generate build-ready slice files which can be interfaced to different RP machines in their native machine formats.

C.1.7 VisCAM Surface (Surface processing)

VisCAM Surface offers you the direct preparation of CAD data from IGES and VDAFS files for your RP application. Imported surfaces can be visualized and interactively manipulated in real time. Comprehensive repair functions assist you in the fast generation of a closed volume model. RP slice files can be generated with definable accuracy directly from the surface data without the so far usual intermediate step over the STL format. However, the direct generation of precise STL files from the surface data is of course also available.

C.2 Steps to Visualize and Validate Results

The model is first converted in I-DEAS to STL file format, which as mentioned earlier is the standard format for most common rapid prototyping technologies, and is ready to be imported into VisCAM RP. Once the model is imported to the software, as shown in Figure C1, the designer has the ability to move and rotate the part to see it from any angle or position, zoom in and out freely to see specific details, select different orientations to build the part, take measurements, look at cross sections at any level, view the model data as wire frame or solid, and more. All these options, shown in Figure C2, allow the designer to visualize the model even before generating the slices.

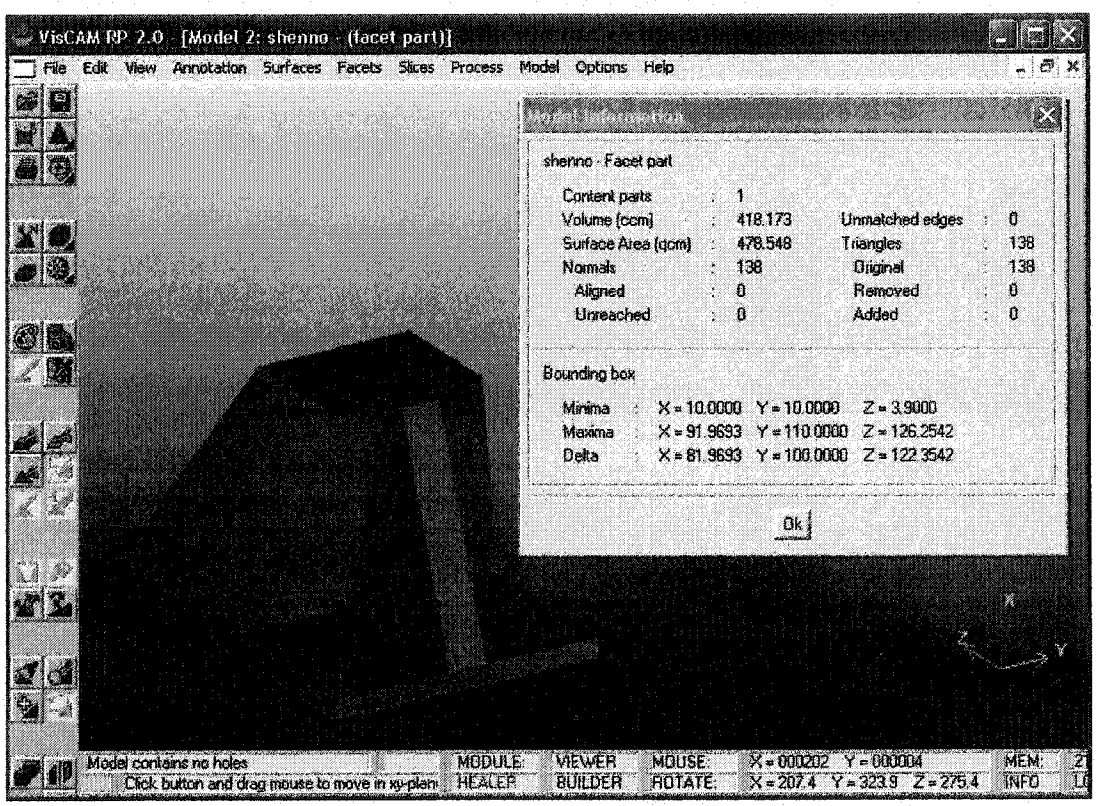


Figure (C1) Case study imported to VisCAM RP

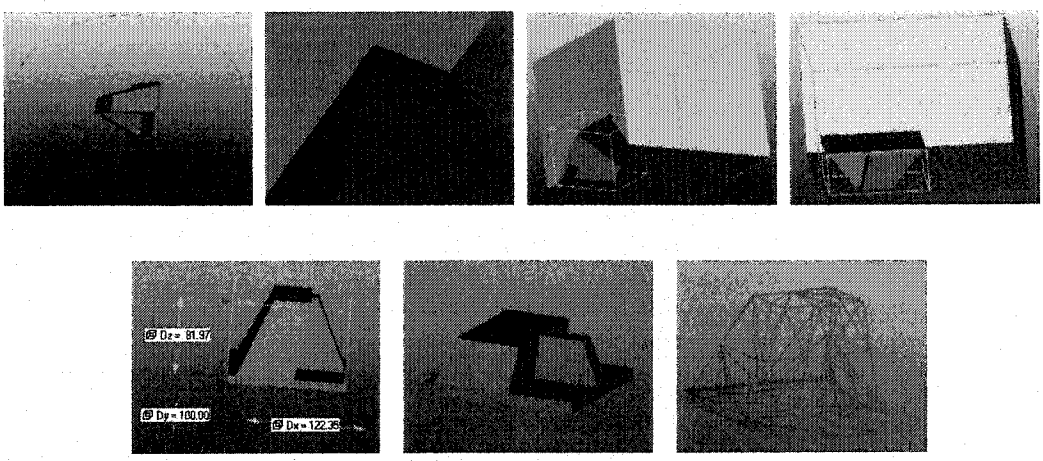


Figure (C2) VisCAM RP visualization options

After the model is imported to the software the designer can then define the machine used to build the part. The software contains a machine database, which is divided into (a) predefined machines and (b) user defined machines. The predefined machines are integrated with over 130 RP machines, which are well known to the system. The parameters are present and cannot be modified. As for the user defined machines, the user can fill it with new machines and private parameters. Figure C3 shows the define machine menu.

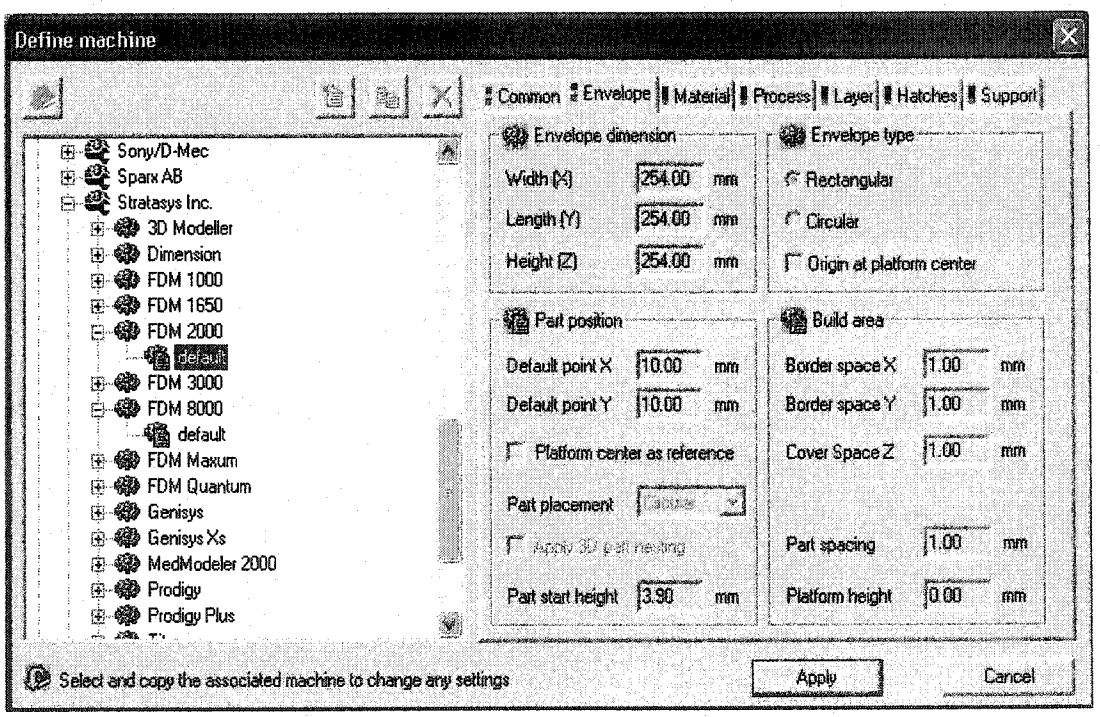


Figure (C3) Define machine menu

The parameters that were used in this thesis were based on the FDM2000 machine Ziemian and Crawn [2001].

C.3 Preparing for Validation

After the machine has been defined and the part is now ready to be virtually built, the user will start by first generating the slices. In order to generate the slices the user defines the build orientation by selecting the bottom plane. In our optimization problem the optimal orientation was orientation 1, demonstrated in Figure C4.

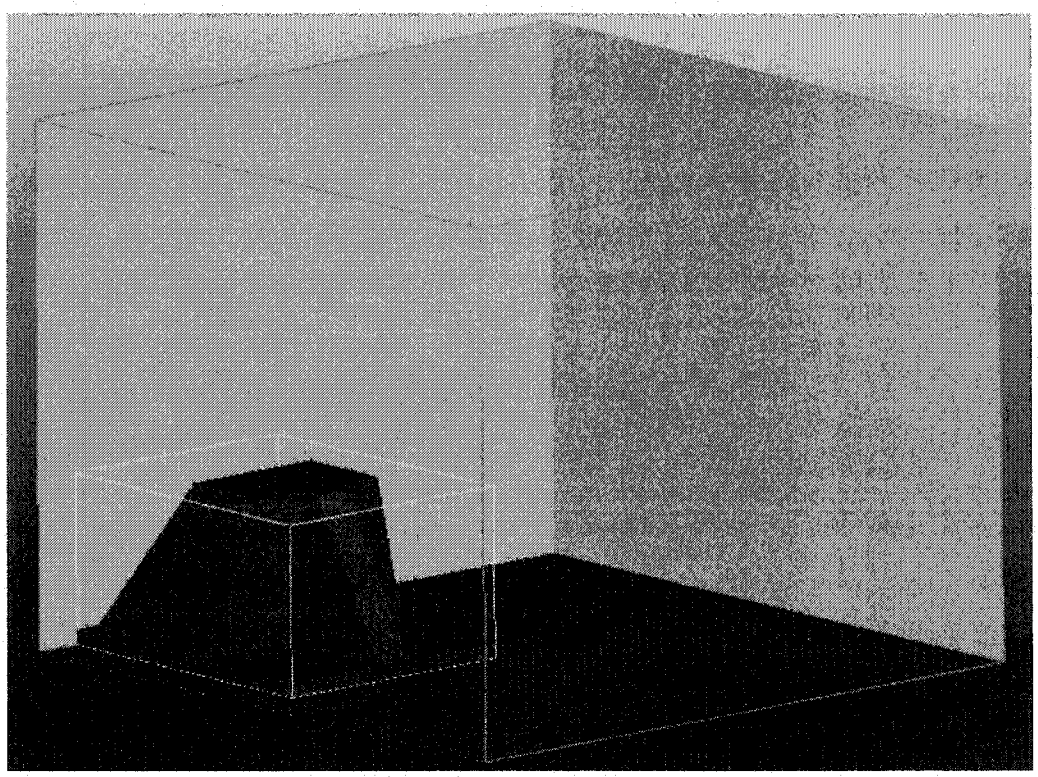


Figure (C4) Orientation 1 in the workplace

Using the generate slice option; the layer thickness was set to 0.239mm, which is the value obtained from the optimization algorithm. Figure C5 illustrates the window where the layer thickness parameter is set in the generate slice option. The software immediately slices the part into layers of 0.239mm each, as seen in Figure C6.

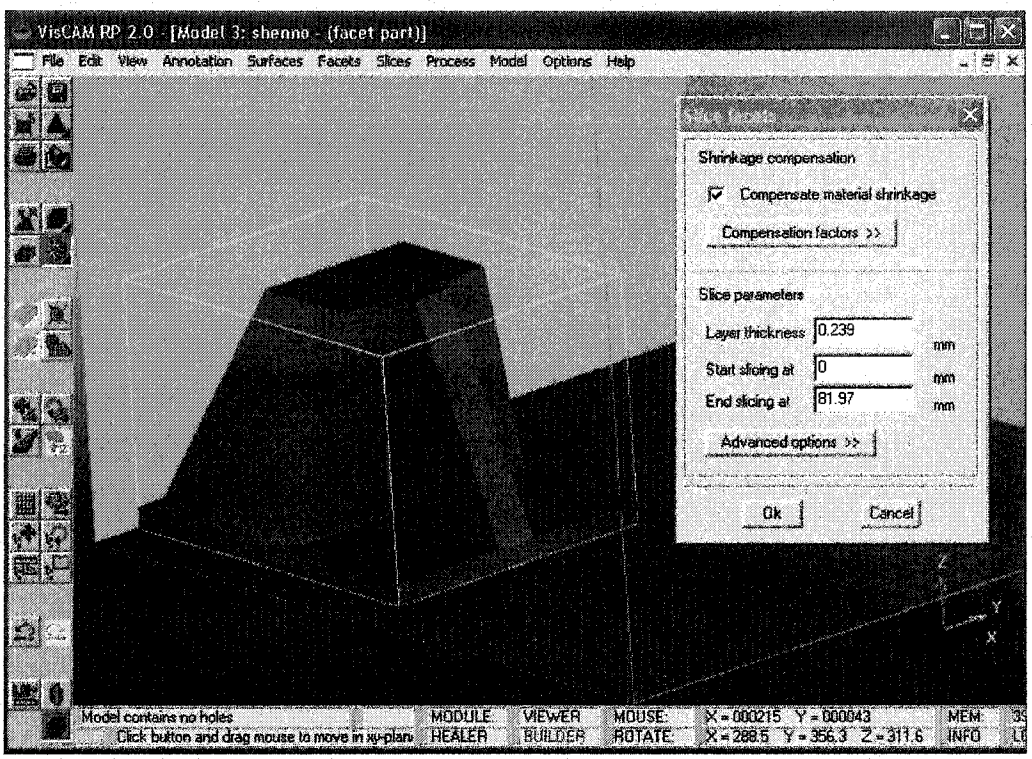


Figure (C5) Layer thickness entry in generate slice option

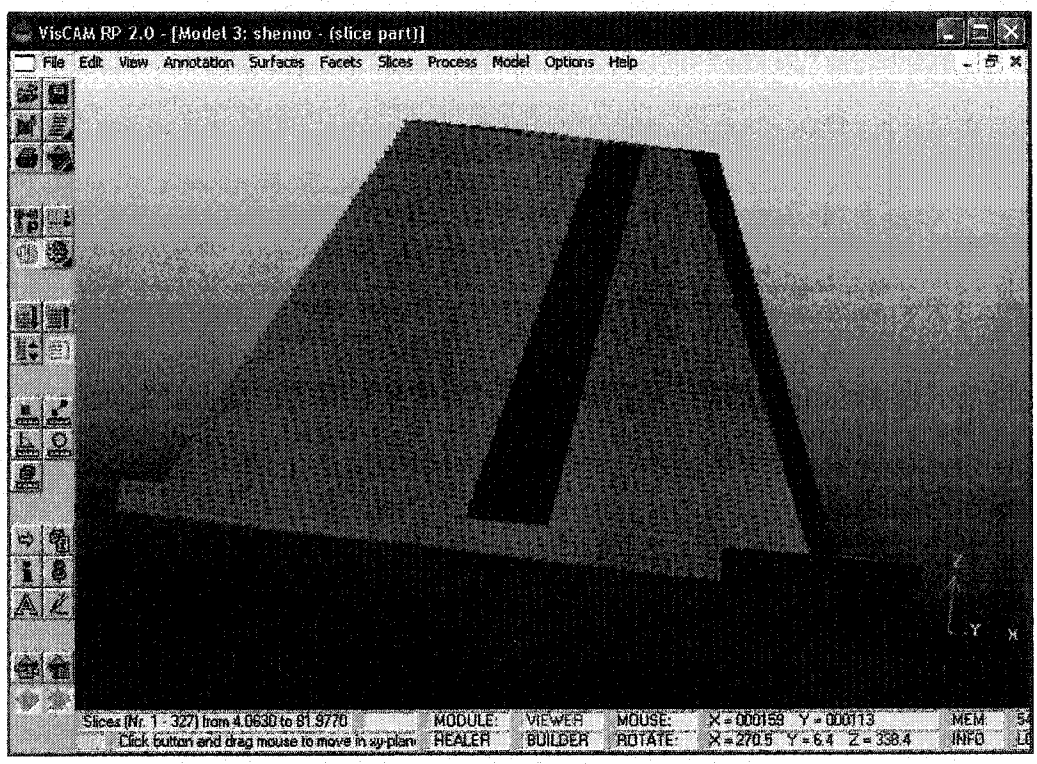


Figure (C6) Stair-stepping effect on model after generating slices

According to our near optimal results, the orientation and layer thickness are set. The next step is to generate the hatches or roads to the desired space or width. From our optimization results the optimal road width value is 0.503mm. By selecting the generate hatch option, the menu shown in Figure C7 appears, allowing the user to define the hatch style for the slice building process. To define the hatch style, the user can either accept the default values, or choose to define own values. In our case the value was selected to match our optimal result from the optimization problem, as shown in the menu illustrated in Figure C8.

After all the required parameters have been set, the user can now start the validation phase.

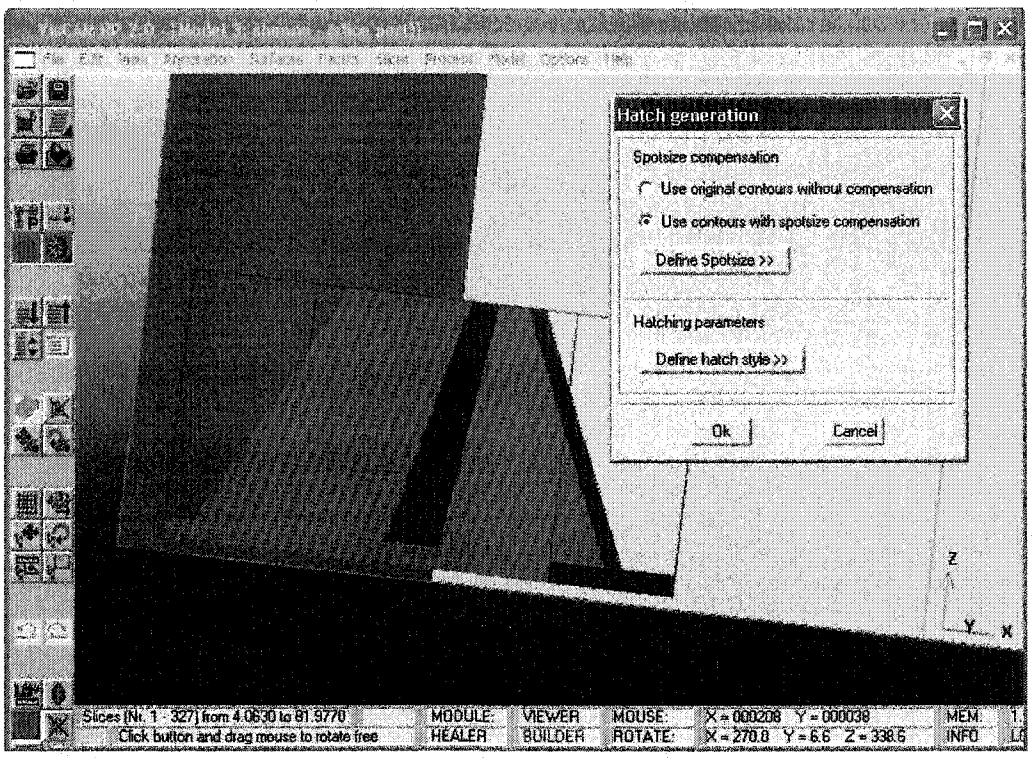


Figure (C7) Generate hatch option

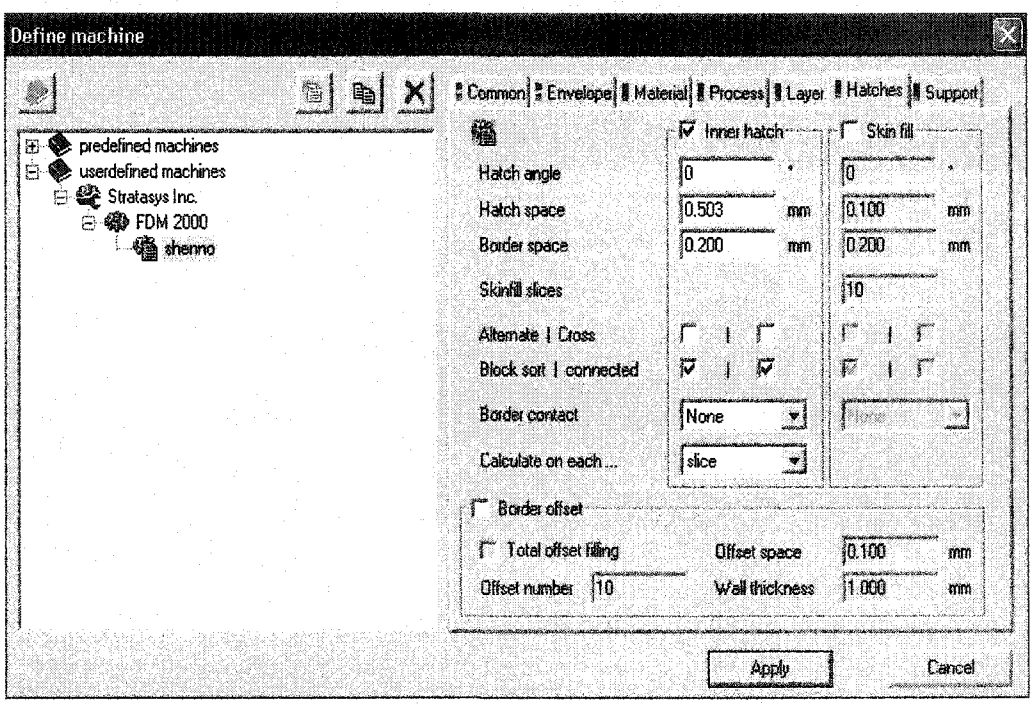


Figure (C8) Defining hatch space menu

APPENDIX D

GA CODE

D.1 Main File to Execute for Mixed Code

```
%          This is the Main File to execute
%          ================================

clear;

% Name of File of the Objective Function:
 name ='objfunction';

% Varialbles Data (Maximum, Minimum, Type(0 ->Descret, 1 ->real)):
 MinA =[1,0.178,0.333];
 MaxA =[7,0.330,0.706];
 TypA =[0,1,1];

 Bound=[MaxA;MinA;TypA];

% Optimization Parameters ( Number of Variables, Minimization, Population Size,
Number of Generation):
 PARAMS =[3,0,50,50];
 PARAMS =[3,0,50,50,4,0,4,2,2,4,0];
% Solution:
[FSol,Sol,BestFitness,Generations] =GAMIX(name,Bound,PARAMS);

FSol
Sol
```

```
objfunctionval(Sol);

figure(2);
plot(Generations,BestFitness)
xlabel('Generation Number');
ylabel('Best Utility Function Fitness Value');
```

D.2 Main File to Execute for Discrete Code

```
%          This is the Main File to execute
%          ================================

clear;

% Name of File of the Objective Function:
 name ='objfunctiond';

% Varialbles Data (Maximum, Minimum, Type(0 ->Descret, 1 ->real)):
 LTVal=[0.178,0.245,0.33];
 RWVal=[0.333,0.511,0.706];
 MinA =[1,1,1];
 MaxA =[7,3,3];
 TypA =[0,0,0];

 Bound=[MaxA;MinA;TypA];

% Optimization Parameters ( Number of Variables, Minimization, Population Size,
Number of Generation):
 PARAMS =[3,0,50,50];
 PARAMS =[3,0,50,50,4,0,4,2,2,4,0];
% Solution:
[FSol,Sol,BestFitness,Generations] =GAMIX(name,Bound,PARAMS);
```

```
FSol
Solt(1) =Sol(1);
Solt(2) =LTVal(Sol(2));
Solt(3) =RWVal(Sol(3));
Solt

objfunctionvald(Sol);

figure(2);
plot(Generations,BestFitness)
xlabel('Generation Number');
ylabel('Best Utility Function Fitness Value');
```

D.3 Build Objective Function (Mixed)

```
function Val =objfunction(VARS)

Ornt =VARS(1);
Lt   =VARS(2);
RW =VARS(3);

LtMin =0.178;
LtMax =0.33;
RWMin =0.333;
RWMax =0.706;

FaceAreas =[3851.191,880.6207,12235.42,1148.825,2234.171,2234.171,2455.923,22
52.357,6498.981,...

879.6707,719.7306,719.7306,880.6207,2455.923,2252.357,3851.191,2200.635,103.32];
```

```
OrntParam = [...%Ornt    Height      BFace#  #ofIncFac   IncFac# FaceAngle
IncFac# FaceAngle  #ofOHF  OHF#   OHFAng     #ofOHFH OHFH   OHFHFreq
OHFH   OHFHFreq
                1,   81.96934,  3,   6,      1,   74.4536,  16,   74.4536,...
                                          4,   65.67,   7,   81.48422,...
                                          14,  81.48422,  9,   65.3026,...
                                          0,   0,      0,   0,...
                                          0,   0,      0,   0,...
                                          0,   0,                  2,   7,   81.48422,
2,   0,   2,      74.77,  2,...
                                                                   14,  81.48422,  2,
0,   2,      74.77,  2,...
                                                                   0,   0,      0,
0,   0,      0,   0,...
                                                                   0,   0,      0,
0,   0,      0,   0,...
                                                                   0,   0,      0,
0,   0,      0,   0,...
                                                                   0,   0,      0,
0,   0,      0,   0,...
                                                                   0,   0,      0,
0,   0,      0,   0;...
                2,   97.46448,  1,   11,     2,   17.67538,  13,   17.67538,...
                                          3,   72.32462,  5,   29.75176,...
                                          6,   29.75176,  8,   72.3246,...
                                          10,  72.3246,   15,  72.3246,...
                                          17,  72.3246,   18,  72.3246,...
                                          16,    34.26568,                 7,   5,
29.75176,  2,   21.15,  2,      41.4,  2,...
                                                                   8,   72.3246,   2,
0,   2,      41.4,  2,...
```

2, 17.67538, 2,
0, 2, 1.95, 2,...
15, 72.3246, 2,
0, 2, 45, 2,...
17, 72.3246, 2,
0, 2, 55.77, 2,...
18, 72.3246, 2,
21.15, 2, 34.62, 2,...
10, 72.3246, 2,
0, 2, 96.26, 2;...
3, 81.96934, 17, 6, 1, 74.4536, 16, 74.4536,...
4, 65.67, 7, 81.48422,...
14, 81.48422, 9, 65.3026,...
0, 0, 0, 0,...
0, 0, 0, 0,...
0, 0, 7, 8, 0, 1,
74.77, 4, 0, 0,...
15, 0, 1,
74.77, 4, 0, 0,...
9, 65.3026, 2,
0, 2, 74.77, 2,...
10, 0, 1,
74.77, 4, 0, 0,...
4, 65.67, 2,
0, 2, 74.77, 2,...
1, 74.4536, 2,
0, 2, 74.77, 2,...
16, 74.4536, 2,
0, 2, 74.77, 2;...
4, 100, 13, 2, 1, 35.96, 16, 35.96,...
0, 0, 0, 0,...

0, 0, 0, 0,...
0, 0, 0, 0,...
0, 0, 0, 0,...
0, 0, 2, 16, 35.96, 2,
0, 2, 21.03, 2,...
6, 0, 1,
43, 4, 0, 0,...
0, 0, 0,
0, 0, 0, 0,...
0, 0, 0,
0, 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, 114.45, 4, 10, 7, 35.96, 14, 35.96,...
8, 65.6716, 15, 65.6716,...
9, 50.55771, 11, 24.33,...
12, 24.33, 17, 65.67,...
18, 65.67, 10, 65.67,...
0, 0, 7, 8, 65.6716,
2, 0, 2, 47.73, 2,...
15, 65.6716, 2,
0, 2, 47.73, 2,...
12, 24.33, 2,
0, 2, 2.97, 2,...
7, 35.96, 2,
6.7, 2, 47.73, 2,...

```
                                                    14,   35.96,   2,
6.7,  2,      47.73, 2,...
                                                    17,   65.67,   2,
6.7,  2,      41.34, 2,...
                                                    18,   65.67,   2,
0,    2,      6.7,   2;...
          6,   122.35,  11,  4,     7,   17.68,   14,   17.68,...
                                    9,   28.33,   4,    24.33,...
                                    0,   0,       0,    0,...
                                    0,   0,       0,    0,...
                                    0,   0,       0,    0,...
                                    0,   0,                 1,   9,   28.33,   2,
8.8,  2,      43.18, 2,...
                                                    0,    0,        0,
0,    0,      0,     0,...
                                                    0,    0,        0,
0,    0,      0,     0,...
                                                    0,    0,        0,
0,    0,      0,     0,...
                                                    0,    0,        0,
0,    0,      0,     0,...
                                                    0,    0,        0,
0,    0,      0,     0,...
                                                    0,    0,        0,
0,    0,      0,     0;...
          7,   122.35,  12,  4,     7,   17.68,   14,   17.68,...
                                    9,   28.33,   4,    24.33,...
                                    0,   0,       0,    0,...
                                    0,   0,       0,    0,...
                                    0,   0,       0,    0,...
```

```
                                        0,    0,                3,    4,    24.33,    2,
0,    2,       43.18, 2,...
                                                                7,      17.68,     2,
41.18, 2,       52.38, 2,...
                                                                14,     17.68,     2,
41.18, 2,       52.38, 2,...
                                                                0,      0,         0,
0,    0,        0,    0,...
                                                                0,      0,         0,
0,    0,        0,    0,...
                                                                0,      0,         0,
0,    0,        0,    0,...
                                                                0,      0,         0,
0,    0,        0,    0];
```

```
V =418173;
VEnvelope =1002902.95;
A =47854.8;
%tw =3.6;
%tf =0.1;

[NumOrient,x] =size(OrntParam);
OrntMin =1;
OrntMinArea =A;
OrntMax =1;
OrntMaxArea =0;
for j =1:NumOrient
   NumIncFaces =OrntParam(j,4);
   OrntAreaTemp =0;
   for i =1:NumIncFaces
```

```
    FaceAngle =OrntParam(j,4+i*2);
    FaceAngleRad =FaceAngle * pi/180;
    FaceArea =FaceAreas(OrntParam(j,3+i*2));
    OrntAreaTemp =OrntAreaTemp +FaceArea * cos(FaceAngleRad);
  end
  if OrntAreaTemp <OrntMinArea
    OrntMin =j;
    OrntMinArea =OrntAreaTemp;
  end
  if OrntAreaTemp >OrntMaxArea
    OrntMax =j;
    OrntMaxArea =OrntAreaTemp;
  end
end

NumIncFaces =OrntParam(OrntMin,4);
SurfRoughTemp =0;
for i =1:NumIncFaces
  FaceAngle =OrntParam(OrntMin,4+i*2);
  FaceAngleRad =FaceAngle * pi/180;
  CuspHeight =LtMin*cos(FaceAngleRad);
  FaceArea =FaceAreas(OrntParam(OrntMin,3+i*2));
  SurfRoughTemp =SurfRoughTemp +CuspHeight * FaceArea;
end
SurfRoughnessMin =SurfRoughTemp/A;

NumIncFaces =OrntParam(OrntMax,4);
SurfRoughTemp =0;
for i =1:NumIncFaces
  FaceAngle =OrntParam(OrntMax,4+i*2);
  FaceAngleRad =FaceAngle * pi/180;
```

```
    CuspHeight =LtMax*cos(FaceAngleRad);
    FaceArea =FaceAreas(OrntParam(OrntMax,3+i*2));
    SurfRoughTemp =SurfRoughTemp +CuspHeight * FaceArea;
  end
  SurfRoughnessMax =SurfRoughTemp/A;

  NumIncFaces =OrntParam(Ornt,4);
  MaxCuspHeight =0;
  SurfRoughTemp =0;
  for i =1:NumIncFaces
    FaceAngle =OrntParam(Ornt,4+i*2);
    FaceAngleRad =FaceAngle * pi/180;
    CuspHeight =Lt*cos(FaceAngleRad);
    if CuspHeight >MaxCuspHeight
      MaxCuspHeight =CuspHeight;
    end
    FaceArea =FaceAreas(OrntParam(Ornt,3+i*2));
    SurfRoughTemp =SurfRoughTemp +CuspHeight * FaceArea;
  end
  SurfRoughness =SurfRoughTemp/A;
  NormSurfRoughness =(SurfRoughness - SurfRoughnessMin)/(SurfRoughnessMax -
SurfRoughnessMin);

  MaxNumIncFac =max(OrntParam(:,4));
  a =MaxNumIncFac;
  NumOvHangFac =OrntParam(Ornt,5+a*2);

  OvHangVol =0;
  for i =1:NumOvHangFac
    FacNum =OrntParam(Ornt,a*2+i*7-1);
    FaceAngle =OrntParam(Ornt,a*2+i*7);
```

```
    FaceAngleRad =FaceAngle * pi/180;
    FacProjArea =FaceAreas(FacNum) * cos(FaceAngleRad);
    NumOHFHeights =OrntParam(Ornt,a*2+i*7+1);
    CumHeights =0;
    TotNumVer =0;
    for j =1:NumOHFHeights
      Height =OrntParam(Ornt,a*2+i*7+j*2);
      NumVer =OrntParam(Ornt,a*2+i*7+1+j*2);
      CumHeights =CumHeights +Height * NumVer;
      TotNumVer =TotNumVer +NumVer;
    end
    AvHeight =CumHeights/TotNumVer;
    OvHangVol =OvHangVol +FacProjArea*AvHeight;
  end
  NormOvHangVol =OvHangVol/(VEnvelope - V);

  % MaxOrientHeight =0;
  % MinOrientHeight =V;
  % for i =1:NumOrient
  %   OrientHeight =OrntParam(i,2);
  %   if OrientHeight >MaxOrientHeight
  %     MaxOrientHeight =OrientHeight;
  %   end
  %   if OrientHeight <MinOrientHeight
  %     MinOrientHeight =OrientHeight;
  %   end
  % end
  %
  % OrientHeight =OrntParam(Ornt,2);
  % fabtime =(6320 - 2005 * Lt - 2299 * RW +454 * Lt * Lt)*V/OrientHeight;
```

```
% fabtimeMin = (6320 - 2005 * LtMax - 2299 * RWMax + 454 * LtMax *
LtMax)*V/MaxOrientHeight;
% fabtimeMax = (6320 - 2005 * LtMin - 2299 * RWMin + 454 * LtMin *
LtMin)*V/MinOrientHeight;
% Normfabtime =(fabtime - fabtimeMin)/(fabtimeMax - fabtimeMin);

fabtime =6320 - 2005 * Lt - 2299 * RW +454 * Lt * Lt;
fabtimeMin =6320 - 2005 * LtMax - 2299 * RWMax +454 * LtMax * LtMax;
fabtimeMax =6320 - 2005 * LtMin - 2299 * RWMin +454 * LtMin * LtMin;
Normfabtime =(fabtime - fabtimeMin)/(fabtimeMax - fabtimeMin);

DimAccuracy =0.005961 - 0.000714 * Lt +0.000558 * Lt * Lt +0.000625 * RW *
RW;
DimAccuracyMin =0.005961 - 0.000714 * LtMax +0.000558 * LtMax * LtMax +
0.000625 * RWMin * RWMin;
DimAccuracyMax =0.005961 - 0.000714 * LtMin +0.000558 * LtMin * LtMin +
0.000625 * RWMax * RWMax;
NormDimAccuracy = (DimAccuracy - DimAccuracyMin)/(DimAccuracyMax -
DimAccuracyMin);



Wght =[0.33,0.33,0.17,0.17];
%Wght =[1,0,0,0];
%Wght =[0,1,0,0];
%Wght =[0,0,1,0];
%Wght =[0,0,0,1];

%ValTemp = Wght(1) * SurfRoughness + Wght(2) * OvHangVol + Wght(3) *
fabtime +Wght(4) * DimAccuracy;
```

```
ValTemp = Wght(1) * NormSurfRoughness + Wght(2) * NormOvHangVol +
Wght(3) * Normfabtime +Wght(4) * NormDimAccuracy;

MaxAllCuspHeight =0.25;

Val =ValTemp;

if MaxCuspHeight >MaxAllCuspHeight
  Val =ValTemp +(10*( MaxCuspHeight - MaxAllCuspHeight))^2;
else
  Val =ValTemp;
End
```

D.4 Build Objective Function (Discrete)

```
function Val =objfunctiond(VARS)

LTVal=[0.178,0.245,0.33];
RWVal=[0.333,0.511,0.706];

Ornt =VARS(1);
Lt =LTVal(VARS(2));
RW =RWVal(VARS(3));

LtMin =0.178;
LtMax =0.33;
RWMin =0.333;
RWMax =0.706;

FaceAreas=[3851.191,880.6207,12235.42,1148.825,2234.171,2234.171,2455.923,22
52.357,6498.981,...
```

879.6707,719.7306,719.7306,880.6207,2455.923,2252.357,3851.191,2200.635,103.32];

OrntParam = [...%Ornt Height BFace# #ofIncFac IncFac# FaceAngle IncFac# FaceAngle #ofOHF OHF# OHFAng #ofOHFH OHFH OHFHFreq OHFH OHFHFreq

1, 81.96934, 3, 6, 1, 74.4536, 16, 74.4536,...
4, 65.67, 7, 81.48422,...
14, 81.48422, 9, 65.3026,...
0, 0, 0, 0,...
0, 0, 0, 0,...
0, 0, 2, 7, 81.48422, 2, 0, 2, 74.77, 2,...
14, 81.48422, 2, 0, 2, 74.77, 2,...
0, 0, 0, 0, 0, 0, 0, 0,...
0, 0, 0, 0, 0, 0, 0, 0,...
0, 0, 0, 0, 0, 0, 0, 0,...
0, 0, 0, 0, 0, 0, 0, 0;...
2, 97.46448, 1, 11, 2, 17.67538, 13, 17.67538,...
3, 72.32462, 5, 29.75176,...
6, 29.75176, 8, 72.3246,...
10, 72.3246, 15, 72.3246,...
17, 72.3246, 18, 72.3246,...

16, 34.26568, 7, 5,
29.75176, 2, 21.15, 2, 41.4, 2,...
8, 72.3246, 2,
0, 2, 41.4, 2,...
2, 17.67538, 2,
0, 2, 1.95, 2,...
15, 72.3246, 2,
0, 2, 45, 2,...
17, 72.3246, 2,
0, 2, 55.77, 2,...
18, 72.3246, 2,
21.15, 2, 34.62, 2,...
10, 72.3246, 2,
0, 2, 96.26, 2;...
3, 81.96934, 17, 6, 1, 74.4536, 16, 74.4536,...
4, 65.67, 7, 81.48422,...
14, 81.48422, 9, 65.3026,...
0, 0, 0, 0,...
0, 0, 0, 0,...
0, 0, 7, 8, 0, 1,
74.77, 4, 0, 0,...
15, 0, 1,
74.77, 4, 0, 0,...
9, 65.3026, 2,
0, 2, 74.77, 2,...
10, 0, 1,
74.77, 4, 0, 0,...
4, 65.67, 2,
0, 2, 74.77, 2,...
1, 74.4536, 2,
0, 2, 74.77, 2,...

16, 74.4536, 2,
0, 2, 74.77, 2;...
4, 100, 13, 2, 1, 35.96, 16, 35.96,...
0, 0, 0, 0,...
0, 0, 0, 0,...
0, 0, 0, 0,...
0, 0, 0, 0,...
0, 0, 2, 16, 35.96, 2,
0, 2, 21.03, 2,...
6, 0, 1,
43, 4, 0, 0,...
0, 0, 0,
0, 0, 0, 0,...
0, 0, 0,
0, 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, 114.45, 4, 10, 7, 35.96, 14, 35.96,...
8, 65.6716, 15, 65.6716,...
9, 50.55771, 11, 24.33,...
12, 24.33, 17, 65.67,...
18, 65.67, 10, 65.67,...
0, 0, 7, 8, 65.6716,
2, 0, 2, 47.73, 2,...
15, 65.6716, 2,
0, 2, 47.73, 2,...

12, 24.33, 2,
0, 2, 2.97, 2,...
7, 35.96, 2,
6.7, 2, 47.73, 2,...
14, 35.96, 2,
6.7, 2, 47.73, 2,...
17, 65.67, 2,
6.7, 2, 41.34, 2,...
18, 65.67, 2,
0, 2, 6.7, 2;...
6, 122.35, 11, 4, 7, 17.68, 14, 17.68,...
9, 28.33, 4, 24.33,...
0, 0, 0, 0,...
0, 0, 0, 0,...
0, 0, 0, 0,...
0, 0, 1, 9, 28.33, 2,
8.8, 2, 43.18, 2,...
0, 0, 0,
0, 0, 0, 0,...
0, 0, 0,
0, 0, 0, 0,...
0, 0, 0,
0, 0, 0, 0,...
0, 0, 0,
0, 0, 0, 0,...
0, 0, 0,
0, 0, 0, 0;...
7, 122.35, 12, 4, 7, 17.68, 14, 17.68,...
9, 28.33, 4, 24.33,...

```
                                    0,   0,     0,   0,...
                                    0,   0,     0,   0,...
                                    0,   0,     0,   0,...
                                    0,   0,                  3,   4,   24.33,   2,
0,   2,      43.18, 2,...
                                                             7,     17.68,     2,
41.18, 2,      52.38, 2,...
                                                             14,    17.68,     2,
41.18, 2,      52.38, 2,...
                                                             0,     0,         0,
0,   0,      0,   0,...
                                                             0,     0,         0,
0,   0,      0,   0,...
                                                             0,     0,         0,
0,   0,      0,   0,...
                                                             0,     0,         0,
0,   0,      0,   0];
```

```
V =418173;
VEnvelope =1002902.95;
A =47854.8;
%tw =3.6;
%tf =0.1;

[NumOrient,x] =size(OrntParam);
OrntMin =1;
OrntMinArea =A;
OrntMax =1;
OrntMaxArea =0;
for j =1:NumOrient
```

```
    NumIncFaces =OrntParam(j,4);
    OrntAreaTemp =0;
    for i =1:NumIncFaces
      FaceAngle =OrntParam(j,4+i*2);
      FaceAngleRad =FaceAngle * pi/180;
      FaceArea =FaceAreas(OrntParam(j,3+i*2));
      OrntAreaTemp =OrntAreaTemp +FaceArea * cos(FaceAngleRad);
    end
    if OrntAreaTemp <OrntMinArea
      OrntMin =j;
      OrntMinArea =OrntAreaTemp;
    end
    if OrntAreaTemp >OrntMaxArea
      OrntMax =j;
      OrntMaxArea =OrntAreaTemp;
    end
  end

  NumIncFaces =OrntParam(OrntMin,4);
  SurfRoughTemp =0;
  for i =1:NumIncFaces
    FaceAngle =OrntParam(OrntMin,4+i*2);
    FaceAngleRad =FaceAngle * pi/180;
    CuspHeight =LtMin*cos(FaceAngleRad);
    FaceArea =FaceAreas(OrntParam(OrntMin,3+i*2));
    SurfRoughTemp =SurfRoughTemp +CuspHeight * FaceArea;
  end
  SurfRoughnessMin =SurfRoughTemp/A;

  NumIncFaces =OrntParam(OrntMax,4);
  SurfRoughTemp =0;
```

```
for i =1:NumIncFaces
  FaceAngle =OrntParam(OrntMax,4+i*2);
  FaceAngleRad =FaceAngle * pi/180;
  CuspHeight =LtMax*cos(FaceAngleRad);
  FaceArea =FaceAreas(OrntParam(OrntMax,3+i*2));
  SurfRoughTemp =SurfRoughTemp +CuspHeight * FaceArea;
end
SurfRoughnessMax =SurfRoughTemp/A;

NumIncFaces =OrntParam(Ornt,4);
MaxCuspHeight =0;
SurfRoughTemp =0;
for i =1:NumIncFaces
  FaceAngle =OrntParam(Ornt,4+i*2);
  FaceAngleRad =FaceAngle * pi/180;
  CuspHeight =Lt*cos(FaceAngleRad);
  if CuspHeight >MaxCuspHeight
    MaxCuspHeight =CuspHeight;
  end
  FaceArea =FaceAreas(OrntParam(Ornt,3+i*2));
  SurfRoughTemp =SurfRoughTemp +CuspHeight * FaceArea;
end
SurfRoughness =SurfRoughTemp/A;
NormSurfRoughness =(SurfRoughness - SurfRoughnessMin)/(SurfRoughnessMax -
SurfRoughnessMin);

MaxNumIncFac =max(OrntParam(:,4));
a =MaxNumIncFac;
NumOvHangFac =OrntParam(Ornt,5+a*2);

OvHangVol =0;
```

```
for i =1:NumOvHangFac
   FacNum =OrntParam(Ornt,a*2+i*7-1);
   FaceAngle =OrntParam(Ornt,a*2+i*7);
   FaceAngleRad =FaceAngle * pi/180;
   FacProjArea =FaceAreas(FacNum) * cos(FaceAngleRad);
   NumOHFHeights =OrntParam(Ornt,a*2+i*7+1);
   CumHeights =0;
   TotNumVer =0;
   for j =1:NumOHFHeights
      Height =OrntParam(Ornt,a*2+i*7+j*2);
      NumVer =OrntParam(Ornt,a*2+i*7+1+j*2);
      CumHeights =CumHeights +Height * NumVer;
      TotNumVer =TotNumVer +NumVer;
   end
   AvHeight =CumHeights/TotNumVer;
   OvHangVol =OvHangVol +FacProjArea*AvHeight;
end
NormOvHangVol =OvHangVol/(VEnvelope - V);

% MaxOrientHeight =0;
% MinOrientHeight =V;
% for i =1:NumOrient
%    OrientHeight =OrntParam(i,2);
%    if OrientHeight >MaxOrientHeight
%       MaxOrientHeight =OrientHeight;
%    end
%    if OrientHeight <MinOrientHeight
%       MinOrientHeight =OrientHeight;
%    end
% end
%
```

```
% OrientHeight =OrntParam(Ornt,2);
% fabtime =(6320 - 2005 * Lt - 2299 * RW +454 * Lt * Lt)*V/OrientHeight;
% fabtimeMin = (6320 - 2005 * LtMax - 2299 * RWMax + 454 * LtMax * LtMax)*V/MaxOrientHeight;
% fabtimeMax = (6320 - 2005 * LtMin - 2299 * RWMin + 454 * LtMin * LtMin)*V/MinOrientHeight;
% Normfabtime =(fabtime - fabtimeMin)/(fabtimeMax - fabtimeMin);

fabtime =6320 - 2005 * Lt - 2299 * RW +454 * Lt * Lt;
fabtimeMin =6320 - 2005 * LtMax - 2299 * RWMax +454 * LtMax * LtMax;
fabtimeMax =6320 - 2005 * LtMin - 2299 * RWMin +454 * LtMin * LtMin;
Normfabtime =(fabtime - fabtimeMin)/(fabtimeMax - fabtimeMin);

DimAccuracy =0.005961 - 0.000714 * Lt +0.000558 * Lt * Lt +0.000625 * RW * RW;
DimAccuracyMin =0.005961 - 0.000714 * LtMax +0.000558 * LtMax * LtMax + 0.000625 * RWMin * RWMin;
DimAccuracyMax =0.005961 - 0.000714 * LtMin +0.000558 * LtMin * LtMin + 0.000625 * RWMax * RWMax;
NormDimAccuracy = (DimAccuracy - DimAccuracyMin)/(DimAccuracyMax - DimAccuracyMin);

Wght =[0.33,0.33,0.17,0.17];
%Wght =[1,0,0,0];
%Wght =[0,1,0,0];
%Wght =[0,0,1,0];
%Wght =[0,0,0,1];
```

```
%ValTemp = Wght(1) * SurfRoughness + Wght(2) * OvHangVol + Wght(3) *
fabtime +Wght(4) * DimAccuracy;
ValTemp = Wght(1) * NormSurfRoughness + Wght(2) * NormOvHangVol +
Wght(3) * Normfabtime +Wght(4) * NormDimAccuracy;

MaxAllCuspHeight =0.25;

Val =ValTemp;

if MaxCuspHeight >MaxAllCuspHeight
  Val =ValTemp +(10*( MaxCuspHeight - MaxAllCuspHeight))^2;
else
  Val =ValTemp;
End
```

VITA AUCTORIS

Name:	Ahmed M. El Shenawy
Place of Birth:	Abu Dhabi, United Arab Emirates
Year of Birth:	1979
Education:	Arab Academy for Science & Technology, Alexandria, Egypt 1996 – 2001 B.Sc. University of Windsor, Windsor, Ontario, Canada 2002 – 2004 M.A.Sc.